Universidade Federal Rural de Pernambuco - UFRPE

Unidade Acadêmica de Garanhuns

Curso de Bacharelado em Ciência da Computação

Raul Pedro de Vasconcelos Sousa

Análise dos Componentes Principais Supervisionada: Uma abordagem não-paramétrica

Garanhuns

2019

Raul Pedro de Vasconcelos Sousa

Análise dos Componentes Principais Supervisionada: Uma abordagem não-paramétrica

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação da Unidade Acadêmica de Garanhuns da Universidade Federal Rural de Pernambuco.

Universidade Federal Rural de Pernambuco - UFRPE

Unidade Acadêmica de Garanhuns

Curso de Bacharelado em Ciência da Computação

Orientador: Tiago Buarque Assunção de Carvalho

Garanhuns

2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Ariano Suassuna, Garanhuns - PE, Brasil

S725a Sousa, Raul Pedro de Vasconcelos

Análise dos componentes principais supervisionada: uma abordagem não-paramétrica / Raul Pedro de Vasconcelos Sousa. - 2019.
58 f. : il.

Orientador: Tiago Buarque Assunção de Carvalho.
Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal Rural de Pernambuco, Departamento de Ciência da Computação, Garanhuns, BR-PE, 2019.
Inclui referências

1. Análise bayesiana 2. Mineração de dados 3. Estatística - Processamento de dados I.Carvalho, Tiago Buarque Assunção de, orient. II. Título

CDD 004.932

Raul Pedro de Vasconcelos Sousa

Análise dos Componentes Principais Supervisionada: Uma abordagem não-paramétrica

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação da Unidade Acadêmica de Garanhuns da Universidade Federal Rural de Pernambuco.

Trabalho aprovado. Garanhuns, 04 de Fevereiro de 2019:

Tiago Buarque Assunção de Carvalho -
Orientador
UAG
UFRPE

Luis Filipe Alves Pereira - Examinador
UAG
UFRPE

Maria Aparecida Amorim Sibaldo de Carvalho - Examinador
UAG
UFRPE

Garanhuns
2019

Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.

Agradecimentos

Agradeço principalmente a minha família pelo amparo e apoio durante essa jornada acadêmica sem vocês tudo teria sido muito mais difícil. Agradeço também ao corpo docente e aos amigos que fiz, levarei sempre comigo o que aprendi com vocês.

"É verdade: amamos a vida não porque estamos acostumados à vida,
mas porque estamos acostumados a amar.
Há sempre alguma loucura no amor,
mas há também sempre alguma razão na loucura.
(Friedrich Nietzsche)

Resumo

Problemas de classificação tem se tornado cada vez mais comuns, sendo utilizados desde da detecção de e-mails *spams* até classificação de tumores em malignos e benignos. Nestes problemas a quantidade de características desempenha um papel fundamental tanto na qualidade quanto no desempenho dos classificadores, nos quais, dados que possuem alta dimensionalidade tendem apresentar taxa de acerto inferior e maior tempo de processamento. Assim técnicas de extração de características são excelentes opções para contornar essa situação, gerando novas características e selecionando as melhores para a classificação. O *Principal Component Analysis* (PCA) é uma das técnicas de extração de características mais utilizadas obtendo, em termos gerais, ótimos resultados, contudo, por ser uma técnica não supervisada que utiliza a variância como critério de seleção, há situações em que o método não consegue extrair as melhores características. Então desenvolvemos uma versão supervisionada do PCA utilizando classificação Bayesiana em conjunto com técnica de estimação de densidade de *Kernel* (janela de Parzen) para avaliar e selecionar as características, ao invés de utilizar a variância como na tradicional implementação do PCA. Propondo assim uma seleção que utiliza o erro Bayesiano como critério base da avaliação. Esse método surgiu como uma extensão do *Minimum Classification Error* PCA (MCPCA) que utiliza o erro Bayesiano como métrica também, contudo, apresentado uma série de restrições, como ser limitado a problemas de apenas 2 classes. Comparamos o método proposto com o PCA, MCPCA e com o *Supervised PCA* (SPCA), outra abordagem supervisionada do PCA, comparando a taxa de acerto por quantidade de características em 4 classificadores para 16 bases de dado. O método proposto apresentou maior taxa de acerto em 72% dos casos, enquanto o PCA, MCPCA e SPCA conseguiram 31%, 36%, 12% respectivamente. No cenário de apenas uma característica o resultado obtido foi de 89%, 14%, 37%, e 25% dos casos para o proposto, PCA, MCPCA e SPCA respectivamente.

Palavras-chave: Análise dos Componentes Principais. Classificação Bayesiana. Janela de Parzen.

Abstract

Problems of classification of data become more commonly used. Classification task has a broader range of applications, ranging from detection of spam emails to classification of malignant and benign tumors. In these problems, the quantity of characteristics plays a fundamental role both in the quality and performance of the classifiers. Data having a high dimensionality tends to have lower accuracy and longer processing time. Feature extraction techniques are excellent solutions to this situation, generating a new set of features and selecting the best ones for classification. Principal Component Analysis (PCA) is one of the most common feature extraction techniques. In general, PCA presents excellent results, but because it is an unsupervised technique there are situations where the method can not extract discriminant features. We developed a supervised version of the PCA using Bayesian classification with the kernel density estimation (KDE) to select features. This method has emerged as an extension of the Minimum Classification Error PCA (MCPCA). MCPCA also uses the Bayesian error as a metric however it presents a series of constraints. Comparing the exposed method with PCA, MCPCA and Supervised PCA (SPCA), another supervised approach to PCA, comparing the accuracy by characteristics in four classifiers to sixteen databases. The proposed method presented the greater accuracy in 72% of the cases. For PCA, MCPCA, and SPCA this number is 31%, 36%, 12%, respectively. When using a single extracted feature, the maximum accuracy if achieved is 89%, 14%, 37%, and 25% of the cases for proposed method, PCA, MCPCA, and SPCA, respectively.

Keywords: Principal Component Analysis. Bayes Classification. Kernel Density Estimation.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

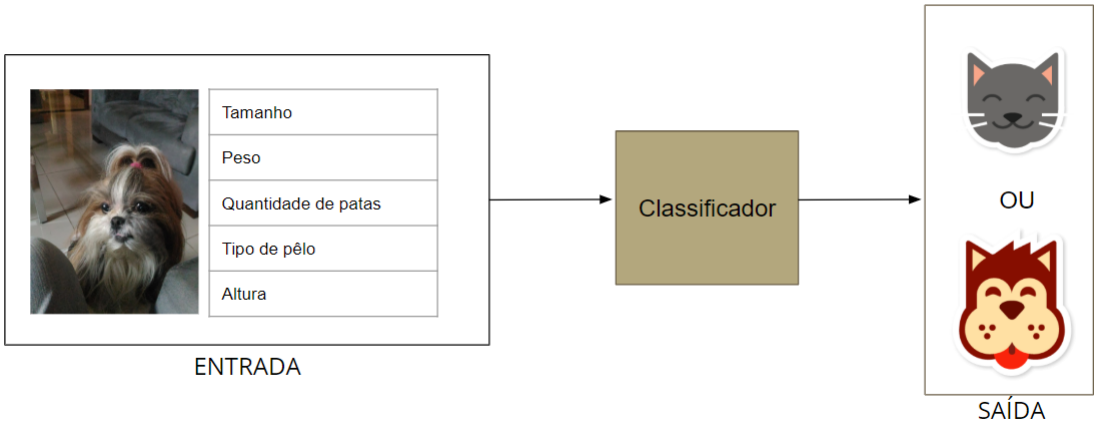


Figura 1 – Exemplo de classificação de cães e gatos. Fonte: Autor

Problemas de classificação em *Machine Learning*, consistem em catalogar um conjunto de dados em grupos predefinidos chamados de classes. Os dados de um problema de classificação consistem em um coleção de instancias, onde cada uma é definida por uma tupla (x,y), onde x é um conjunto de características que descrevem a instancia e y o rótulo da classe que a instância pertence. As características podem ser que qualquer tipo, enquanto as classe deve ser categórica (TAN et al., 2018). A Figura 1 ilustra o processo, nele temos a classificação de cães e gatos, onde temos como entrada uma imagem de um cachorro ou um gato, possuindo como características: tamanho, peso, quantidade de patas etc. A partir delas o classificador determinará se é um cão ou um gato, as classes do problema.

Quando o número de características é grande podemos afirmar que os dados apresentam alta dimensionalidade (DIAS, 2013), o que impacta significativamente o processo de classificação gerando processos de classificação mais longos, demandando mais recursos computacionais para executa-los e armazena-los. Além disso, pode levar a perda de especificidade dos dados, uma vez que, as características se tornam esparsas e apresentando distancias praticamente equidistantes, a chamada: maldição da dimensionalidade (MACEDO, 2012 apud DY, 2007). Portanto, técnicas de redução de dimensionalidade são bem-vindas e desejadas.

O PCA é uma técnica de extração de características não supervisionada, ou seja, não utiliza as classes dos dados em seu funcionamento. O PCA funciona projetando novas características que são independentes, utilizando a variância como métrica para avaliar a qualidade da característica. Por se tratar de uma técnica não supervisionada o PCA não

considera nenhuma informação das classes dos dados observados e por vezes a variância não é um critério discriminante para a classificação, gerando classificações menos eficientes. Pensando nisso Carvalho, Tsang e Cavalcanti (2017) propôs o *Minimum Classification Error PCA* (MCPCA), uma implementação supervisionada do PCA que utilizava um *score* baseado no erro Bayesiano para avaliar as características. O método apresentou bons resultados, contudo, tem algumas limitações, como de só funcionar para problemas com apenas duas classes, restringindo consideravelmente a quantidade de problemas que o MCPCA pode ser aplicado

Este trabalho consiste em desenvolver uma versão supervisionada do PCA que utiliza uma métrica semelhante, mas sem possuir as restrições existentes, utilizando alguns recursos da estatística como: o teorema de Bayes e a estimativa de densidade de *Kernel*, conhecido também como janela de Parzen. Ao final realizamos uma comparação entre o método proposto, o PCA, o MCPCA e o *Supervised PCA* (SPCA), outra implementação supervisionada do PCA proposta por Barshan et al. (2011).

1.1 Justificativa

Com o aumento na quantidade de dados disponíveis e a evolução dos classificadores os problemas de classificação estão cada vez mais comuns seja em reconhecimento facial, diagnósticos de doenças, perfil de cliente etc, contudo, a grande massa de dado pode apresentar diversos problemas para classificação seja devido ao tempo de execução para processar grandes quantidades de dados ou devido à maldição da dimensionalidade, que afirma que existe um limite máximo de características onde após ele o desempenho do classificador irá decair. O PCA é o algoritmo mais utilizado para redução de dimensionalidade geralmente obtendo ótimos resultados, contudo, o PCA utiliza a variância como métrica de seleção das características, porém esta forma de escolha muitas vezes prejudica a tarefa de classificação. Portanto, técnicas alternativas para redução de dimensionalidade que consigam resultados superiores nessas situações são extremamente bem-vindas e desejadas.

1.2 Objetivos

1.2.1 Geral

Este trabalho tem como objetivo propor uma versão supervisionada do PCA,tomando como base a implementação do mesmo realizada por Carvalho, Tsang e Cavalcanti (2017), o MCPCA, sanando as limitações apresentadas pelo mesmo, utilizando alguns conceitos da estatística tais como o Teorema de Bayes e modelos não paramétricos, visando obter resultados semelhantes ao MCPCA em problemas de 2 classes e superiores ao PCA e SPCA nos demais.

1.2.2 Específicos

1. Desenvolver uma versão supervisionada do PCA sem restrições apresentados pelo MCPCA;
2. Desenvolver um método utilizando o erro Bayes como critério de seleção;
3. Conseguir obter taxas de acerto superiores ao PCA em diversas bases;
4. Comparar o método proposto com outra implementação supervisionada do PCA, o SPCA;

1.3 Organização do Texto

A organização deste trabalho segue a seguinte forma: O Capítulo 2 apresenta a fundamentação teórica dos conceitos utilizados no desenvolvimento do método e da pesquisa. Neste capítulo é descrito em mais detalhes o processo de classificação e o papel da extração de características nela, assim como o funcionamento da janela de parzen e da classificação Bayesiana. O Capítulo 3 apresenta o funcionamento do método proposto descrevendo como os conceitos apresentados no capitulo anterior são aplicados por ele. Já o Capítulo 4 detalha como foram realizados os experimentos, o Capítulo 5 que exibe os resultados obtidos e a Capítulo 6 mostra a conclusão do trabalho realizado e quais são os possíveis trabalhos futuros para esta pesquisa.

2 Fundamentação teórica

Neste capítulo serão apresentados os principais conceitos utilizados no desenvolvimento desta pesquisa. Iniciamos apresentando o que é classificação e as razões para utilizar extração de características nesses problemas. Em seguida apresentamos o *Principal Component Analysis* (PCA) uma técnica de extração de características que serve como base para as demais abordagens apresentadas neste trabalho. Na seção 2.3 apresentamos o *Minimum Classification Error PCA* (MCPCA) que é à abordagem supervisionada do PCA utilizada como base para desenvolvimento do método proposto. Na seção 2.4 *Supervised PCA (SPCA)*, outra abordagem supervisionada do PCA que utiliza um processo de extração de características diferente do PCA. Na seção 2.5 explicamos o funcionamento do teste de hipótese de sinal. Nas demais seções apresentamos os conceitos utilizados para desenvolver o método proposto: a janela de Parzen e classificação Bayesiana junto com o erro de Bayes.

2.1 Classificação

2.1.1 O que é classificação?

Em aprendizagem de máquina podemos dividir os problemas em supervisionados: quando as instâncias são dadas em conjunto de um rótulo que corresponde a saída correta (KOTSIANTIS, 2007) e não-supervisionados quando não possuem. Problemas de classificação tratam de ensinar máquinas a catalogar dados em um conjunto de rótulos pré-estabelecidos chamados de classes, mediante a algum critério especifico utilizando as características dos mesmo. Características ou atributos são as menores propriedades individuais que descrevem cada instância dos dados (BISHOP, 2006). Estes problemas são presentes nas mais diversas áreas do conhecimento como por exemplo: classificação de e-mails em *spams* a partir de seus *headers*, tumores em malignos e benignos baseados nos resultados de exames de ressonância magnética e classificação de galaxias baseadas em seus formatos (TAN; STEINBACH; KUMAR, 2005). Em geral, se divide os dados a serem classificados em dois tipos: treino e teste, onde o "treino"corresponde aos dados que realizam calibragem do modelo e o "teste"a avaliação, verificando a capacidade de generalização do classificador. Quando a classificação apresenta apenas 2 classes é chamada de binária e multi classe quando contem mais de 2 classes.

2.1.2 Alta dimensionalidade

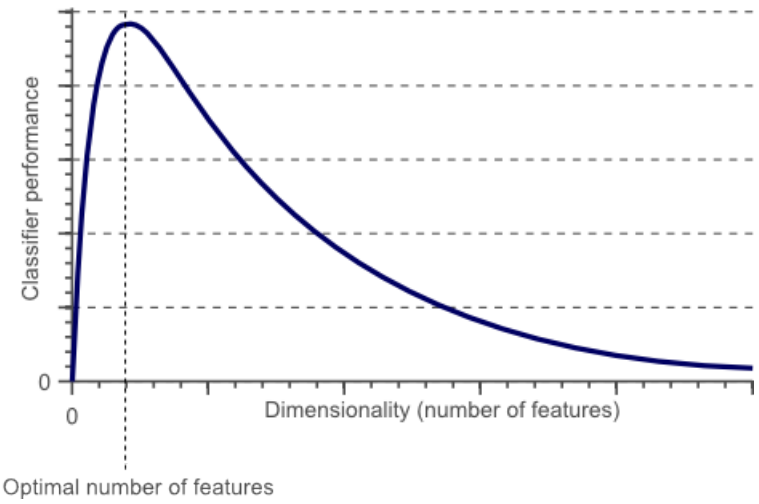


Figura 2 – Maldição da dimensionalidade em um classificador. Fonte: Spruyt (2014a)

Quando as instâncias de dados apresentam um grande número de características dizemos que os mesmo apresentam alta dimensionalidade. Na classificação a alta dimensionalidade costuma trazer problemas tais como:

1. Características correlacionadas ou redundantes: esta situação faz com os dados apresentem características desnecessárias, em outras palavras, significa que dados apresentam uma dimensionalidade maior que necessária (DIAS, 2013). Essas situações acabam por tornar os custo de classificação maiores, seja devido ao aumento do tempo execução dos classificadores ou seja pela maior necessidade de armazenado que os dados demandam.

2. A maldição da dimensionalidade: foi proposta por Bellman (1957) e afirma que em dados com um grande número de características ocorre uma perda da particularidade dos dados, uma vez que, a distância Euclidiana do ponto mais distante e do mais próximo são praticamente as mesmas. Sendo assim podemos afirmar que um grande número de características prejudica não só a qualidade da classificação como também que existe uma quantidade ideal de características, em que, a partir deste valor haverá perda no desempenho da classificação como ilustrado na Figura 2.

Portanto reduzir a dimensionalidade dos dados, removendo ou transformando características que atrapalham o processo de classificação é algo desejável e importante, gerando um ganho tanto de performance computacional quanto nas taxas de acerto dos classificadores.

2.1.3 Extração de características

É um processo de redução de dimensionalidade pelo qual um conjunto inicial de dados brutos é reduzido a grupos mais gerenciáveis para processamento. Uma característica desses grandes conjuntos de dados é um grande número de variáveis que exigem muitos recursos de computação para processar combinando variáveis reduzindo efetivamente a quantidade de dados que devem ser processados e, ao mesmo tempo, descrevendo de forma precisa e completa o conjunto de dados original (DEEPAI, 2018).

2.2 Principal Component Analysis - PCA

2.2.1 A Técnica

A Análise de Componentes Principais (PCA) é uma técnica de extração de características, que realiza uma combinação linear dos dados, projetando novos dados que apontam para as direções da variância máxima no espaço (MAćKIEWICZ, 1993). Essas direções são autovetores gerados a partir da matriz de covariância dos dados observados. Geralmente, apenas alguns autovetores são selecionados, aqueles que possuem os maiores autovalores. O autovalor é equivalente à variância de uma nova variável, que é obtida projetando os dados em um autovetor. As novas variáveis não só têm variância máxima, mas eles também não são correlacionadas (BISHOP, 2006). É uma técnica não supervisionada e muito utilizada atualmente para a extração de características.

2.2.2 Extração de características

O processo de extração de características com PCA funciona da seguinte maneira:

1. Receber a matriz de dados D_{nxf} onde n são os elementos e f são as características (*features*).
2. Encontrar a matriz de covariância, COV, dos dados. $COV = (D - M) * (D - M)^T$, onde M_{nxf} é uma matriz em que cada elemento de todas linhas são a média de cada coluna de D.
3. Extrair os autovalores e autovetores da matriz de covariância.
4. Ordenar os autovetores em função dos autovalores de forma que os que possuem maior valor são priorizados.
5. Projetar a matriz D'_{nxk} de novos dados.$D' = D * A$ onde A_{fxk} A são os autovetores de D e k quantidade de características se deseja manter, onde $k < f$.

2.2.3 Problemas

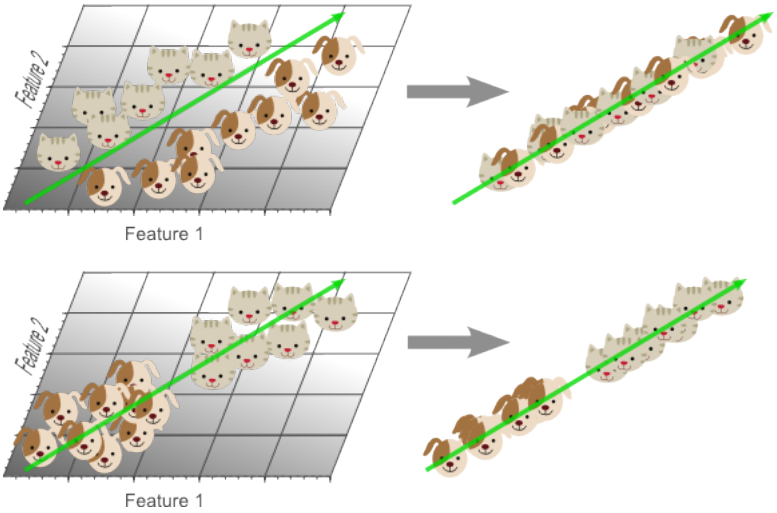


Figura 3 – Duas situação ao extrair características com PCA. Fonte: Spruyt (2014b)

O PCA por se tratar de uma técnica não-supervisionada, ou seja, não utiliza nenhuma informação advinda das classes, assumindo que características com maior variância são melhores para a classificação. Contudo, nem sempre isso é verdade, pois, por vezes o autovetor que possui a informação mais discriminante tem menor variância, levando a uma perda significativa na qualidade da classificação (SPRUYT, 2014b) como demostrado na Figura 3 onde na imagem acima temos uma situação onde PCA seleciona uma característica não-discriminante levando a uma péssima separação dos dados, em contra-partida da imagem abaixo onde é feita uma seleção correta que separa os dados idealmente.

2.3 Minimum Classification Error PCA

2.3.1 A Técnica

O *Minimum Classification Error PCA* (MCPCA) é uma implementação supervisionada do PCA que seleciona as características visando minimizar o erro Bayesiano. O método segue basicamente as mesmas etapas do PCA para realizar a extração de características diferindo somente em utilizar um *score*, definido por:

$$score_i = \begin{cases} \frac{(w_{1i}-w_{2i})^2}{\lambda_i} & \text{, se } \lambda_i \neq 0 \\ 0 & \text{, caso contrário,} \end{cases} \tag{2.1}$$

onde w são as médias por classe e i o índice da característica avaliada, ao invés da variância para selecionar os melhores autovetores, onde as características com menor score são consideradas melhores para a classificação. A Equação 2.1 é originada da estimação do erro de Bayes através da distancia de *Mahalanobis*.

2.3.2 Limitações

Por estimar o erro de Bayes utilizando a distancia de *Mahalanobis* o método possui as seguintes limitações:

1. Limitado a apenas problemas binários (2 classes).
2. As classes devem possuir probabilidade iguais.
3. Os dados devem seguir a distribuição normal.

2.4 Supervised PCA

É uma implementação supervisionada do PCA proposta por (BARSHAN et al., 2011) que consiste em na projeção de um sub-espaço que maximiza o critério de independência de Hilbert-Schmidt. Neste método os autovetores são extraídos de uma matriz $Q = DHLHD^T$, onde :

1. D é matriz de dados;
2. H é uma matriz dada por: $H = I - n^{-1}O$, onde O é uma matriz quadrada composta apenas de 1;
3. L é uma matriz dado por: $L = Y^T Y$, onde $Y_{ji} = \begin{cases} 1, \text{ se } D_i \text{ pertence a classe de índice j} \\ 0, \text{ caso não pertença} \end{cases}$

Assim como no PCA os autovetores são selecionados maximizando a variância e a projeção dos novos dados é realizada de forma análoga ao PCA.

2.5 Teste do Sinal

O teste do sinal é um teste de hipóteses simples e não-paramétrico. O teste é utilizado quando se deseja saber se existe diferença de comportamento entre duas amostra pareadas, sendo a hipótese nula, H_0, a não existência de diferenças entre as amostras. O teste pode ser descrito em 5 passos:

1. Determinar a hipótese alternativa H_1 e o grau de significância α. Em geral utiliza-se $alpha = 0.05$.
2. Remover do teste os pares que possuem valor igual.
3. Calcular a diferença entre os pares restantes e determinar a direção de seu sinal.

$$sinal_i = \begin{cases} 1, \text{ se } P_{1i} - P_{2i} > 0 \\ 0, \text{ caso não seja.} \end{cases}$$

4. Obter o $p - valor$ a partir dos dados do problema:

$$p - valor = X \sim Binomial(n; 0.5). \tag{2.2}$$

   onde n é quantidade de pares restantes do problemas e X variável aleatória, cuja a condição vai depender de H_1.

5. Rejeitar H_0 caso $p - valor > \alpha$.

2.6 Janela de Parzen

Janela de Parzen ou Estimador de Densidade por *Kernel* (KDE) é um método estatístico não paramétrico para estimar a função de densidade de probabilidade (FDP) em variáveis continuas. Para isso o método utiliza uma função *Kernel*, $K(.)$, não-negativa e parâmetro de suavização h, que consiste em uma janela de tamanho fixo onde os elementos mais próximos do valor estimado tem maior peso de que os que se encontram mais longe. É descrita matematicamente da seguinte forma:

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{2.3}$$

Embora a escolha do *Kernel* afeta a estimativa da densidade, a literatura sugere que esse efeito é bastante pequeno, com resultados empíricos muito semelhantes para diferentes escolhas de *Kernel* (TERRELL; SCOTT, 1992). Já a escolha de h tem impactos significativos na qualidade dos resultados, como exemplificado na figura Figura 4.

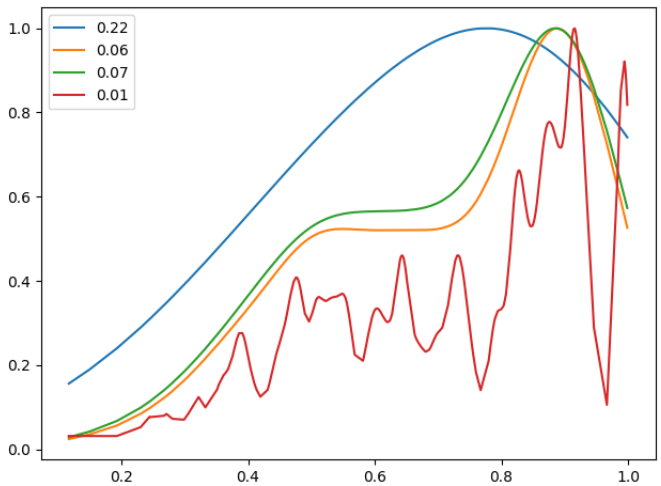


Figura 4 – Comparação entre funções de densidade probabilidade estimadas com parâmetros de suavização diferentes para a mesma base. Fonte: Autor

2.6.1 Escolha do parâmetro de suavização

Como dito anteriormente a escolha do parâmetro de suavização é extremamente importante assim como desafiador. A escolha de um parâmetro correto impacta significativamente a compreensão e estimação da probabilidade dos dados. A literatura apresenta diversas formas para obtê-la, neste trabalho utilizamos minimização do erro médio quadrado integrado (MISE), descrita da seguinte forma:

$$MISE(f(x)) = \frac{1}{4}h^4k_2^2 \int f''(x)^2dx + \frac{1}{nh}\int K(t)^2dt. \tag{2.4}$$

Sendo h a largura do *Kernel*, k_2^2 uma constante resultante pertencente ao segundo termo no processo de expansão da Série de Taylor, f"(x) a derivada segunda da densidade real, n o tamanho da amostra dos dados e K a função *Kernel* utilizada (VARGAS, 2015). Contudo, devido a natureza não paramétrica dos dados não se conhece a função geradora dos dados, então supõem-se que os dados pertencem à distribuição normal (*Kernel* Gaussiano) e manipula-se a Equação 2.4 obtendo a seguinte equação:

$$h = 1.06 * \delta * n^{-\frac{1}{5}}. \tag{2.5}$$

Onde δ é o desvio padrão, contudo, a Equação 2.5 funciona bem em casos onde os dados seguam uma distribuição gaussiana, mas não em casos onde isso não ocorre. Então utilizando como base uma função geradora bimodal, substitui-se o desvio padrão pelo intervalo interquartis (IQR), gerando duas opções de equações de suavização: Equação 2.6 e Equação 2.7 (SILVERMAN, 1986).

$$h1 = 0.79 * IQR * n^{-\frac{1}{5}}. \tag{2.6}$$

$$h2 = 0.9 * a * n^{-\frac{1}{5}}, onde \quad a = min\left(\delta, IQR/1.34\right). \tag{2.7}$$

2.6.2 Função de Kernel Gaussiana

Existem diversas funções de *Kernel*, a função escolhida para este trabalho foi a função Gaussiana, Equação 2.8.

$$K(u) = \frac{1}{2\pi}exp\left(-\frac{1}{2}u^2\right). \tag{2.8}$$

O *Kernel* Gaussiano corresponde a distribuição normal sendo esta a função mais comumente utilizada. Aplicando Equação 2.8 na Equação 2.3 obtemos a janela de Parzen no seguinte formato:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h\sqrt{2\pi}}exp\left[-0.5\left(\frac{x-x_i}{h}\right)^2\right]. \tag{2.9}$$

Temos então a janela de Parzen com *Kernel* Gaussiano.

2.7 Classificação Bayesiana

A classificação Bayesiana, como o nome sugere, utiliza o teorema de Bayes para atribuir uma classe a um elemento. O teorema Bayes é uma forma simples de calcular as probabilidades condicionais, descrita pela seguinte equação:

$$P(c|x) = \frac{P\left(x|c\right) P\left(c\right)}{P\left(x\right)}, \tag{2.10}$$

onde $P(x|c)$ é a verossimilhança (*likelihood*), $P(c)$ a probabilidade da classe (priori) e $P(x)$ a probabilidade do elemento, dada por:

$$P(x) = \sum_{i=1}^{l} P\left(x|c_i\right) P\left(c_i\right). \tag{2.11}$$

em que l é a quantidade de classes. $P(c|x)$ é a probabilidade a posteriori, em outras, palavras é a probabilidade da classe para o elemento avaliado x. A classificação Bayesiana se da de forma simples, selecionando a como correta a classe com maior probabilidade (posteriori) garantindo o erro de Bayes, a menor probabilidade erro que um classificador pode obter (KEINOSUKE, 1990). É possível obter $P(x|c)$ a partir de uma função de densidade de *Kernel*.

2.7.1 Erro de Bayes

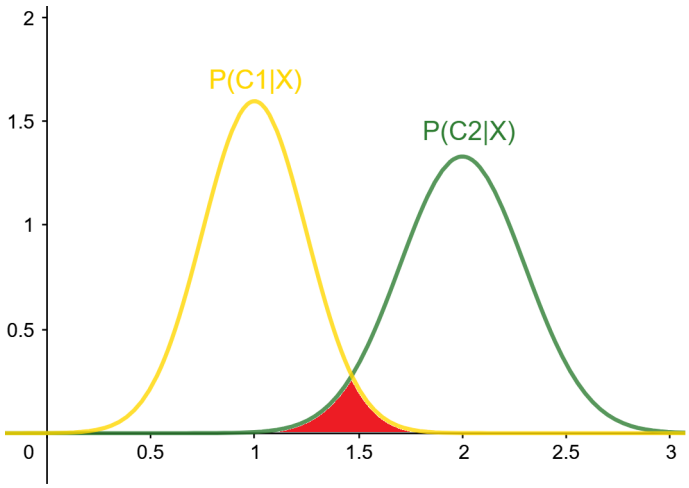


Figura 5 – Erro Bayesiano entre duas distribuições normais destacado em vermelho. Fonte: Autor

A probabilidade de erro é um parâmetro essencial para o reconhecimento de padrões. Devido ao teorema de Bayes o erro de Bayesiano possui a menor probabilidade de erro para a distribuição dada (KEINOSUKE, 1990). Podemos entender o erro de Bayes como a área onde ocorre a sobreposição das probabilidades das classes, como exibido na Figura 5, ou seja, é área onde existem elementos que podem pertencem a ambas as classes, gerando erros

na classificação. Para problemas multi-classe o erro pode ser obtido a partir do somatório das regiões de maior probabilidade de escolha do classificador Bayesiano, representado matematicamente através da seguinte equação:

$$E_{bayes} = 1 - \sum_{i=1}^{k} \int p\left(c_i\right) p\left(x|c_i\right) dx. \tag{2.12}$$

Equação 2.12 onde $p(c_i)$ é a priori da classe e $p(x|c_i)$ a probabilidade do elemento para a classe avaliada.

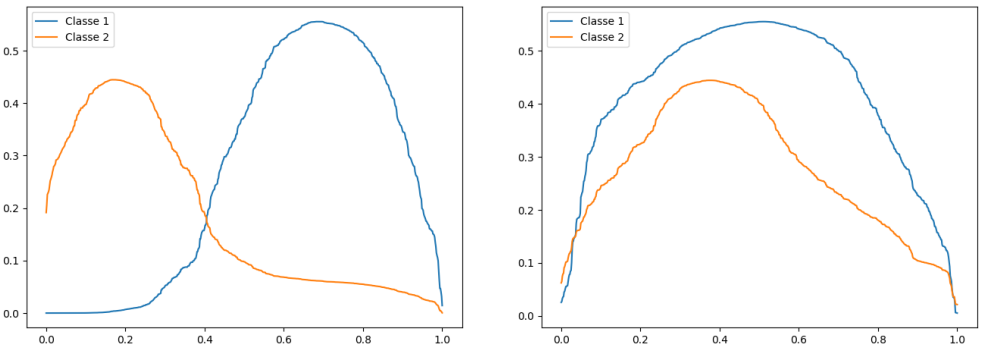
2.8 Considerações Finais do Capítulo

Neste capítulo vimos todos os conceitos que envolvem essa pesquisa, no Capítulo 3 veremos razão da utilizar o erro de Bayes como base para selecionar as características além de como o método proposto utiliza a classificação Bayesiana junto com a janela de Parzen para realizar avaliação das características. No Capítulo 4 apresentaremos uma discussão acerca da escolha do parâmetro de suavização h e como ele influencia nos resultados. Já no Capítulo 5 utilizamos o teste do sinal para validar alguns dos resultados obtidos.

3 Método Proposto

Neste capítulo será detalhada o funcionamento do algoritmo proposto apresentando seu funcionamento e como ele utiliza a classificação Bayesiana em conjunto da janela de Parzen para avaliar qualidade das características no processo de classificação assim como a combinação dessas técnicas permite contornar as limitações do *Minimum Classification Error PCA* (MCPCA) apresentadas na seção 2.3.

3.1 Seleção de características



(a) Característica com menor erro Bayesiano

(b) Característica com maior erro Bayesiano

Figura 6 – Comparação entre a função de densidade de duas características de uma mesma base. Fonte: Autor

O método utiliza classificação Bayesiana para selecionar as melhores características para a classificação, visando atingir o erro Bayesiano Equação 2.12. Uma vez que o erro Bayesiano nos oferece a menor taxa de erro possível a um classificador (KEINOSUKE, 1990) utiliza-lo como critério de seleção permite escolher características que apresentam melhor separação entre as classes. Por exemplo na Figura 6 temos duas características de uma mesma base, podemos notar que a Figura 6a é uma melhor opção para classificação, pois, apresenta uma separação entre classes bem mais evidente do que a Figura 6b, onde ocorre uma sobreposição entre os gráficos bem mais intensa, ou seja, possui um erro Bayesiano maior, em outras palavras a Figura 6a é mais discriminante do que a Figura 6b.

3.2 Descrição

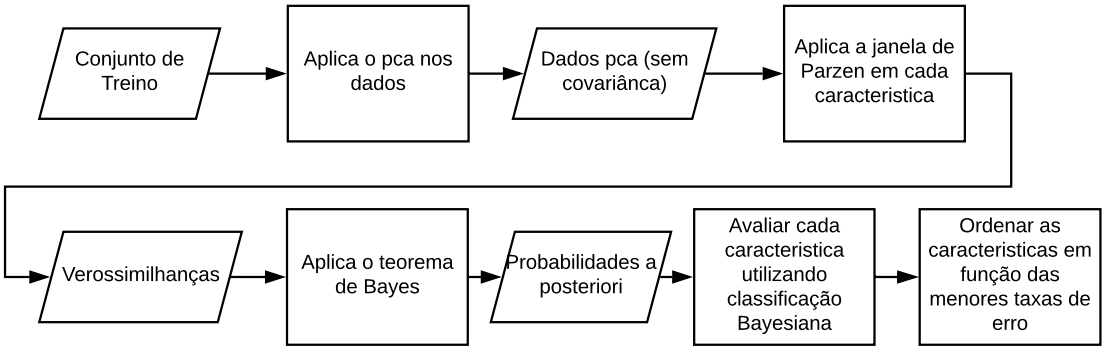


Figura 7 – Diagrama do processo de treinamento do método. Fonte: Autor

O método consiste em uma abordagem supervisionada do PCA utilizando classificação Bayesiana para selecionar as melhores características em problemas de classificação, ao invés da variância como é tradicionalmente realizado pelo PCA. Diferentemente do *Minimum Classification Error PCA* (MCPCA) o método proposto não utiliza a distancia de Mahalanobis para estimar o erro Bayesiano, consequentemente, as limitações de somente funcionar para bases binárias com distribuição normal e prioris entre classes iguais não se aplicam ao método proposto.

A Figura 7 é um diagrama de como é realizado o treinamento do método. Primeiramente aplica-se PCA nos dados avaliados, projetando novos dados independentes, ou seja, que não apresentam covariância entre si. Esse processo é importante, pois garante que as características possam ser avaliadas isoladamente. Então são calculadas as probabilidades posteriores das classes para cada características utilizando teorema de Bayes, Equação 2.10, contudo, é necessário obter a verossimilhança (*likelihood*), que nada mais é que a probabilidade do elemento dado a classe, a $P(x|c)$ citado na seção 2.7. Para isso primeiramente é necessário separar os dados por classe e em seguida utilizasse a janela de Parzen, Equação 2.9, para estimar esse valor.

O processo de calculo da verossimilhança é feito para cada componente da características, onde excluímos o elemento atualmente avaliado e aplicamos a janela de parzen dele para os demais, semelhante a um *leave-one-out*. É interessante comentar que a utilização da janela de Pazen não só permite obter a verossimilhança como também sana a necessidade dos dados pertencerem à distribuição normal, como é exigida pelo MCPCA, uma vez que janela estima o comportamento dos dados.

Após obter o vetor de verossimilhanças aplica-se o teorema de Bayes em cada ponto do vetor de características obtendo suas probabilidades a posteriori para cada classe, em seguida aplica-se a classificação Bayesiana para cada ponto, extraindo a taxa de erro que é

dada pela Equação 3.1

$$error_f = \sum_{i=1}^{n} \begin{cases} 1, \text{ se } \arg\max_{k=1,\ldots,l}\{P(c_k|x_i)\} \neq L_{real} \\ 0, \text{ caso contrário.} \end{cases} \tag{3.1}$$

onde f é o índice da característica, arg max a classe de maior probabilidade e L_{real} a classe que o elemento realmente pertence. Em caso de empate nas taxas de erro é utilizado a probabilidade de erro média, Equação 3.2, como critério de desempate, onde max é o valor de maior probabilidade.

$$M_f = \frac{1}{n}\sum_{i=1}^{n} 1 - \max_{k=1,\ldots,l}\{P(c_k|x_i)\}. \tag{3.2}$$

Uma vez treinado o processo de projeção de novas características é feita de forma análoga ao PCA.

3.3 Comparação com MCPCA

A distancia de Mahalanobis utilizada pelo MCPCA para estimar o erro Bayesiano consegue apresentar maior precisão ao erro real existente nas características, uma vez que, método proposto utiliza a janela de Parzen para estimar a função de densidade de probabilidade (FDP) das características que é bastante sensível ao parâmetro de suavização escolhido, o que já indica que o comportamento estimado e o real dos dados apresentaram algumas diferenças, consequentemente impactando a área do erro Bayesiano. Contudo essa abordagem ainda consegue obter uma aproximação ao erro Bayesiano suficiente para realizar uma boa avaliação da qualidade das características, como veremos no Capítulo 5, além de contornar as limitações do MCPCA, pois, a janela de Parzen faz com que os dados não precisem ser gaussianos. A classificação Bayesiana pode ser aplicada a n classes levando em consideração também a priori das mesmas, não exigindo que elas sejam iguais.

3.4 Algoritmo

Apresentaremos a seguir um algoritmo que descreve o treino do método proposto, recebendo como parâmetros: uma matriz contendo os dados em que se deseja realizar a extração de características, um vetor contendo as classes dos dados, um com as probabilidades

a priori das classes e a dimensão dos dados.

Data: $dados, classes, prioris, dimensao$
Result: Autovetores ordenados em função da taxa de erro

```
1  dadosPCA ← aplicarPCA(dados);
2  dadosPorClasses ← separarPorClasse(dadosPCA, classes);
3  for i ← 0; i < dimensao; i++ do
4      caracAvaliada ← dados.T[i];
5      h ← calcularH(caracAvaliada);
6      for each indice,dadosClasse ∈ dadosPorClasses do
7          verossimilhancas[indice] ← janelaParzen(caracAvalia, dadosClasse, h);
8          probabilidades[indice] ← teoBayes(prioris, verossimilhancas[indice]);
9      end
10     erros[i] ← calcularErro(probabilidades, classes);
11 end
12 ordenarAutovetores(erros, autoVetores)
```

Algoritmo 1: Pseudocódigo do funcionamento do método

Na linha 1 é aplicado o PCA ao conjunto de dados que compõe o treino tornando os dados independentes, na linha 2 separamos o conjunto por suas respectivas classes gerando l conjuntos novos, em que l é a quantidade de classes existente na base de dados. Na linha 3 percorremos os dados em função de suas características para se obter a verossimilhança e calcular a taxa de erro para cada uma delas. Na linha 4 é feita a separação dos elementos da características i a ser avaliada, gerando um vetor que contem apenas elementos que compõem essa característica, na linha 5 é calculado o parâmetro de suavização e seu valor é atribuído a variável h. Nas linhas 6 a 9 é realizado o calculo da verossimilhança e taxa de erro utilizamos um laço (linha 6) para aplicar esse processo para cada classe existente na base obtendo assim as probabilidade Bayesianas (linha 8) para cada uma delas. Na linha 10 é feito o calcula dos erros seguindo as critérios das equações: 3.1 e 3.2. Encerramos o algoritmo ordenando os os autovetores em função da menor taxa de erro (linha 12).

3.5 Considerações Finais do Capítulo

Neste capítulo detalhamos o funcionamento do método proposto descrevendo como ele utiliza a classificação Bayesiana e a janela de Parzen, apresentados no Capítulo 2. Mostrou-se as diferenças em relação ao PCA e como essa abordagem permite contornar as limitações do MCPCA. No capítulo seguinte veremos como foram realizados experimentos e ilustraremos também como a escolha do parâmetro de suavização da janela de Parzen impacta a classificação.

4 Metodologia de pesquisa

Neste capítulo serão descritos como os experimentos foram realizados, apresentando os processos, assim como as bases dados e classificadores utilizados. Também teremos um discussão sobre a escolha do parâmetro de suavização h e como ele impacta na taxa de acerto do classificador.

4.1 Experimento

A taxa de acerto por quantidade de características mantida foi utilizada como métrica para avaliação. Para cada ponto é calculada a média de 100 *holdout*, onde 50% dados destinados a treino.A taxa de acerto foi obtida utilizando os seguintes classificadores:1NN (Nearest Neighbor) com distância euclidiana, *Naive Bayes* com *Kernel* gaussiano, Arvore de decisão com índice de Gini's e Analise de discriminante linear (LDA). Calculamos um intervalo de confiança de 95% supondo que as médias das taxas de acerto seguem a distribuição *t-Student* no intervalo de $[a - E, a + E]$, onde a é média da taxa de acerto e $E = \frac{1{,}984\delta}{\sqrt{100}}$ e δ é o desvio padrão da taxa de acerto. O experimento foi desenvolvido utilizando linguagem de programação *python* com as bibliotecas:*numpy*,*sklearn* e *numba*.

4.2 Bases utilizadas

Aqui são listadas as bases de dados utilizadas nos experimentos.

- Banknote: A *Banknote authentication* tem 1372 elementos, 4 características e 2 classes.
- Pima: A *Pima Indians Diabetes* tem 768 elementos, 8 características e 2 classes.
- Survival: A *Haberman's Survival* tem 306 elementos, 3 características e 2 classes.
- Monk: A *MONK's Problems* tem 432 elementos, 6 características e 2 classes.
- Immunotherapy: A *Immunotherapy* tem 90 elementos, 7 características e 2 classes.
- Hillvalley: A *HillValley* tem 606 elementos, 100 características e 2 classes.
- Bank: A *Bank Marketing* tem 4,521 elementos, 44 características e 2 classes. Foi realizada um processo de binarização para converter as características categóricas em numéricas gerando 16 novas características.

- Titanic: *Titanic cleaned* tem 891 elementos, 10 características e 2 classes. Foi realizada um processo de binarização em na característica "Embarked"gerando 3 novas características de valor 0 ou 1. Também foram removidas as características:"Cabin", "name", "PassengerId", "Ticket"e "Title".
- Obs network: A *Burst Header Packet (BHP) flooding attack on Optical Burst Switching (OBS) Network* tem 1075 elementos, 22 características e 4 classes.
- Lettter: A *Letter Recognition* tem 20000 elementos, 15 características e 26 classes.
- Dermatology: A *Dermatology* tem 366 elementos, 34 características e 6 classes.
- Leaf: A *Leaf* tem 340 elementos, 15 características e 36 classes.
- Wine: A *Wine* tem 178 elementos, 13 características e 3 classes.
- Wine quality: A *Wine Quality* tem 6497 elementos, 11 características e 10 classes.
- Mice: *Mice Protein Expression* tem 552 elementos, 80 características e 8 classes. Removemos as linhas que continham atributos faltando e convertemos os atributos categóricos para 0 e 1.
- UserKnowledge: A *User Knowledge Modeling* tem 80 elementos, 5 características e 4 classes.

4.3 Classificadores

Aqui são citados os classificadores utilizados na pesquisa assim como uma breve explicação de seu funcionamento.

- 1NN: É a implementação do algoritmo KNN com k = 1. Utiliza a distancia euclidiana para realizar a classificação onde a classe é selecionada em função da classe do elemento mais próximo ao observado.
- Naive Bayes: utiliza classificação Bayesiana,como na Equação 2.10, para determinar a classe do elemento. Utilizamos a versão gaussiana do método que supõem que os dados seguem uma distribuição normal utilizando o desvio padrão dos dados como um dos parâmetros para estimar as probabilidades.
- LDA: *linear discriminant analysis*, cada classe é modelada como Gaussiana (com uma matriz de covariância e um vetor de média). os dados são classificadas na classe do vetor de média mais próximo de acordo com a distância de *Mahalanobis*. (TORKKOLA, 2001)

- Arvore de decisão (*Decision Tree*): classifica os dados, através de uma série de perguntas sobre os recursos associados as características. Cada questão está contida em um nó e cada nó interno aponta para um nó filho para cada resposta possível à sua pergunta. As perguntas formam, assim, uma hierarquia, codificada como uma árvore (KINGSFORD; SALZBERG, 2008). Em nossa pesquisa utilizamos o índice de Gini com no mínimo 10 nós por folha.

4.4 Escolha do Parâmetros de Suavização

Como dito anteriormente no Capítulo 2 a escolha do parâmetro de suavização impacta significativamente na estimação de densidade de *(*Kernel), consequentemente, vai impactar na taxa de acerto dos classificadores citados previamente. Utilizamos tanto a Equação 2.6 quanto a Equação 2.7 e foi possível observar que a Equação 2.7 obteve resultados gerais melhores de obter taxas de acerto mais altas em praticamente todas as bases. Na Figura 8 exemplifica bem como a escolha do parâmetro impacta os resultados além de refletir o que ocorreu na maioria das bases, em que a Equação 2.7 obteve resultados superiores. Já a Figura 9 apresenta os resultados para a base *Bank* representando um dos poucos casos em que a Equação 2.6 se sobressaiu. Resultados empíricos demonstram quem em 84% das vezes o valor do parâmetro de suavização é $h \leq \frac{\delta}{2}$, onde δ é desvio padrão (WANDERLEY, 2013). Acreditamos que isso reflete no sucesso da Equação 2.7, pois, esta tende a gerar resultados menores, justificando também a razão da Equação 2.6 ter se saído melhor na base *Bank* que demonstrou se adaptar melhor em janelas mais amplas. A seguir apresentaremos alguns gráficos que exemplificam bem esse fenômeno, onde o eixo X é a quantidade de características e Y a taxa de acerto.

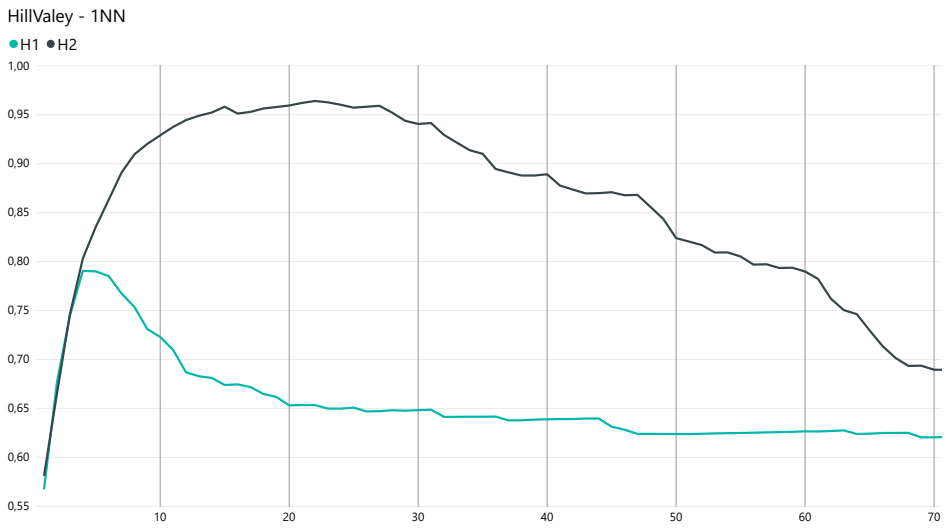


Figura 8 – Base HillValley - comparação entre parâmetros de suavização. Taxa de acerto por quantidade de características. Fonte: Autor

Na Figura 8 podemos notar como a escolha do parâmetro pode impactar na taxa de certo, onde na base *Hill-Valey* ocorreu uma diferença enorme entre os parâmetros a partir de 4 características onde o método utilizando a Equação 2.7 conseguiu obter 96,4% de acerto enquanto a Equação 2.6 conseguiu atingir no máximo 70%.

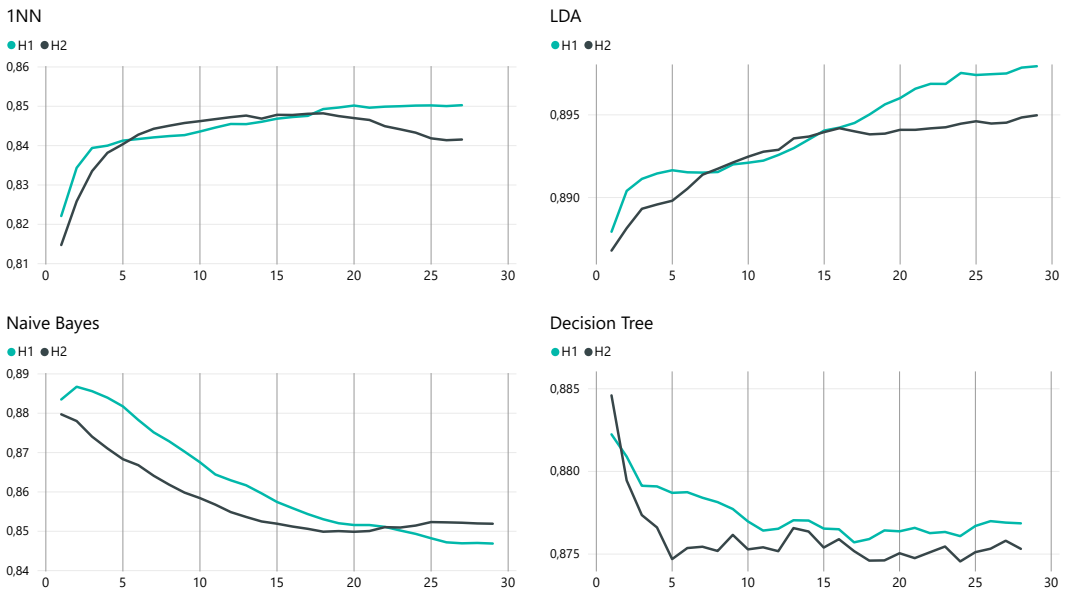


Figura 9 – Base Bank - comparação entre parâmetros de suavização. Taxa de acerto por quantidade de características. Fonte: Autor

Na Figura 9 temos a base *Bank*, podemos observar que o a Equação 2.6 obteve resultados melhores em 3 dos 4 classificadores ficando só abaixo na Arvores de decisão com uma diferença ínfima de apenas 0,24% em 1 características.

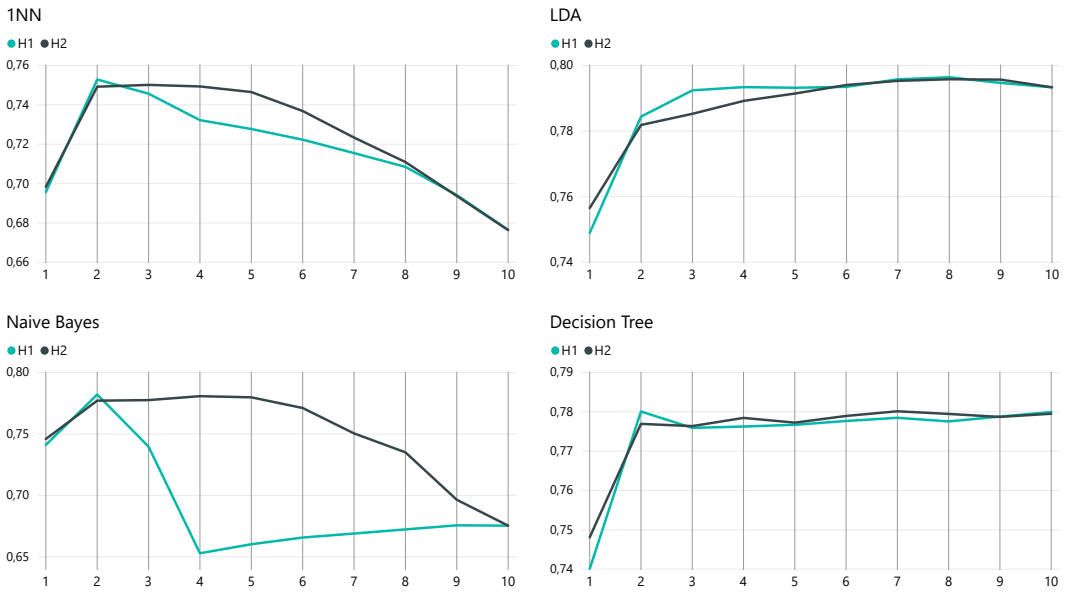


Figura 10 – Base Titanic - comparação entre parâmetros de suavização. Taxa de acerto por quantidade de características. Fonte: Autor

A Figura 10 apresenta os resultados da comparação para a base *Titanic*, assim como na base *Bank* a Equação 2.6 obteve resultados melhores, apresentando taxa de acerto maiores com menos características em todos os 4 classificadores. É interessante notar que apesar da visível queda que ocorre no *Naive Bayes* a partir de 3 característica com a equação Equação 2.6, ela obtém uma taxa de acerto máxima com 2, o que entendemos como sendo um resultado melhor.

4.5 Considerações finais do capítulo

Neste capitulo discorremos como o método funciona e a forma em que os experimentos foram realizados para obter os resultados que serão apresentados no capítulo seguinte. Também foi feita uma pequena comparação entre parâmetros de suavização onde o a Equação 2.7 apresentou resultados melhores, sendo esta utilizada como parâmetro padrão dos resultados, com exceção da bases *Bank* e *Titanic* que utilizaram a Equação 2.6, possivelmente por essas bases apresentarem comportamento que exige uma janela maior para obtenção de um ajuste mais preciso.

5 Resultados

Neste capítulo serão apresentados e discutidos os resultados obtidos pelo método proposto, comparando-o com outras implementações supervisionadas do PCA existentes na literatura e com próprio PCA. Os resultados são apresentados utilizando gráficos lineares, onde o eixo X é a quantidade de características presentes e o eixo Y a taxa de acerto.

5.1 Comparação com outros PCAs supervisionados

Nesta seção será realizada uma comparação entre método proposto com PCA tradicional e outras versões do mesmo, que são: *Minimun Classification Error PCA* (MCPCA) e o *Supervised PCA* (SPCA). A seção foi divida em duas partes, onde na primeira são os resultados para bases com apenas 2 classes e a segunda para bases com mais de 2 classes, essa divisão foi feita devido ao MCPCA ser limitado para apenas base de dados com 2 classes.

5.1.1 Bases com 2 classes

5.1.1.1 BankNote

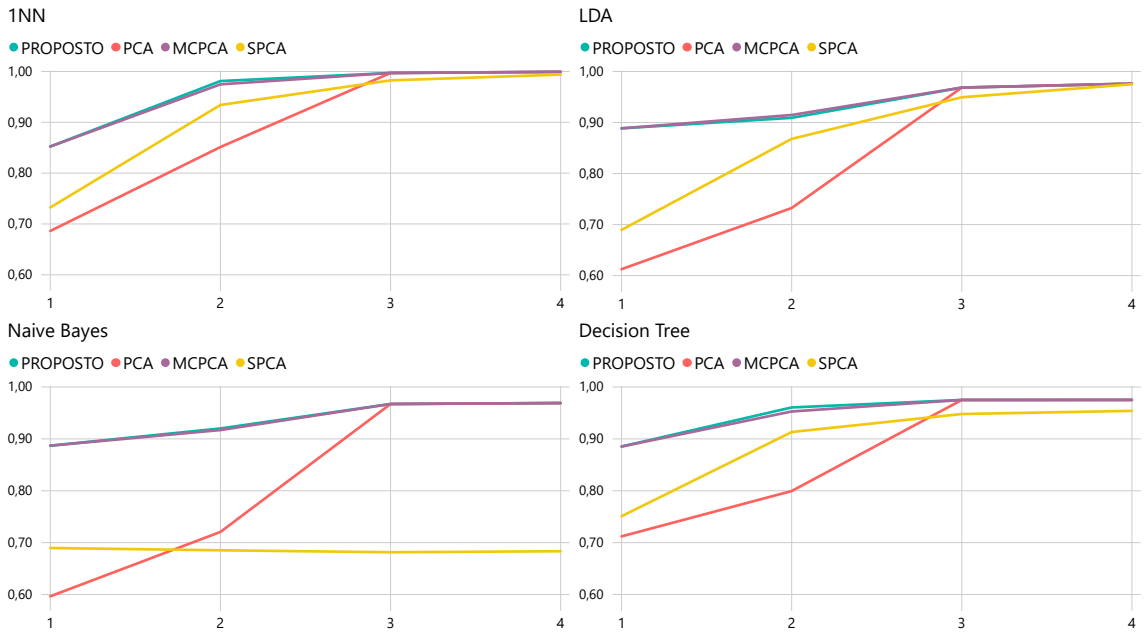


Figura 11 – Base BankNote - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

A Figura 11 apresenta os resultados para base *BankNote*, nela o método proposto e o MCPCA foram equivalentes obtendo os melhores resultados no intervalo de até 2 características obtendo taxas de acerto maiores que o PCA e SPCA. A maior taxa de acerto ocorreu com 4 características, 99,9% sendo atingida por todos os métodos no classificador 1NN. Já maior diferença surge no *Naive Bayes* entre o método proposto e o MCPCA, ambos com 88,7% para com o PCA que obteve 59,6% no intervalo de 1 característica, diferença de 29,1%.

5.1.1.2 Pima

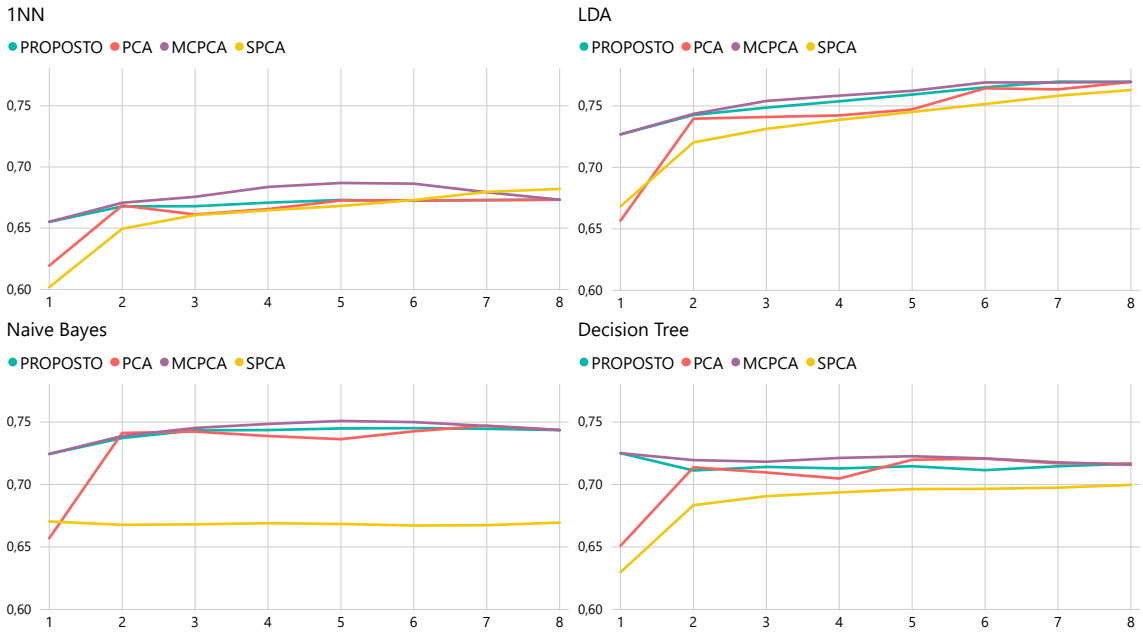


Figura 12 – Base Pima - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base Pima,(Figura 12) o MCPCA e o método proposto obteve os melhores resultados, onde para 1 característica conseguiram diferenças significativas em relação aos demais como podemos observar na Figura 12. No intervalo entre 3 e 6 características houve uma leva vantagem para o MCPCA, contudo, somente para o classificador 1NN essa diferença não cai dentro do intervalo de confiança. A maior taxa de acerto, de aproximadamente 77%, foi obtida no LDA por todos os métodos com exceção do SPCA, que obteve 76,5%, em 8 características. Já maior diferença aparece na árvore de decisão em 1 característica, onde método proposto e o MCPCA obtiveram 72,4% e SPCA obteve quase 63%. Diferença de 9,4%.

5.1.1.3 Survival

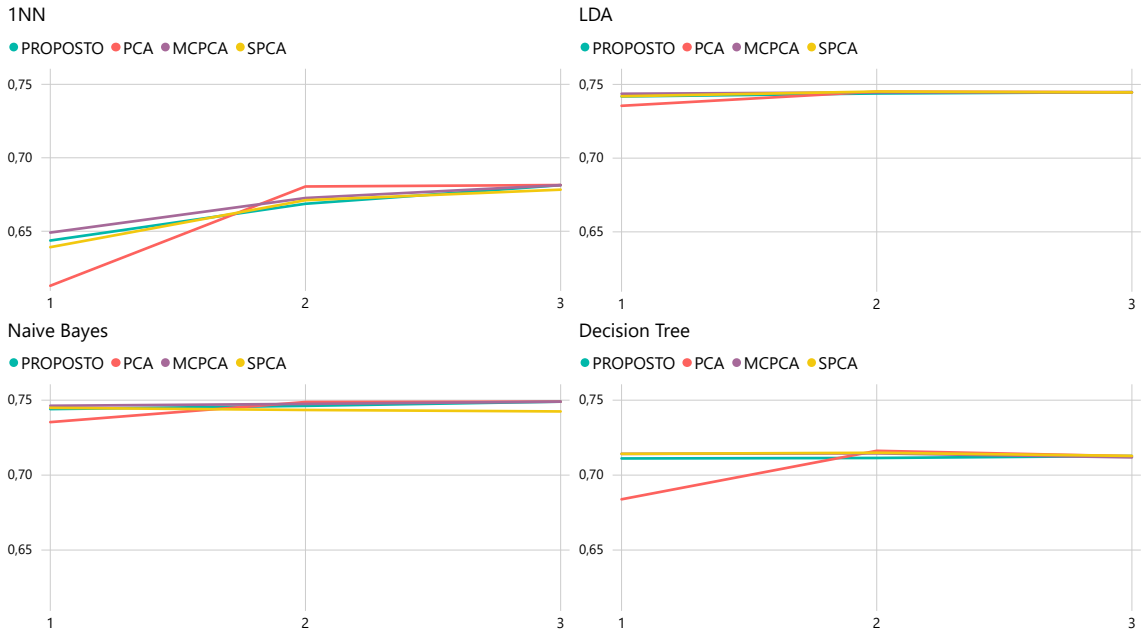


Figura 13 – Base Survival - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *Survival*, Figura 13, os métodos obtiveram resultados semelhantes, contudo, os métodos supervisionados conseguiram obter resultados superiores para 1 uma característica em relação ao PCA. Com 2 características o PCA obtém um resultado médio superior, mas, ainda dentro do intervalo de confiança que foi estabelecido, mesmo para o 1NN onde a diferença é mais notável, tornando os resultados equivalentes. A maior taxa de acerto foi de 74,9% obtida no *Naive Bayes* pelo PCA,SPCA e pelo método proposto. A maior diferença ocorre na árvore de decisão entre o SPCA e o SPCA ambos com 71,4% e o PCA com 68,4%. Diferença de 3,6%, neste mesmo intervalo o método proposto obteve 71,1% diferença de apenas 0.3% em relação ao maior resultado que cai no intervalo de confiança do método proposto que é de [70,4%, 71,7%].

5.1.1.4 Monk

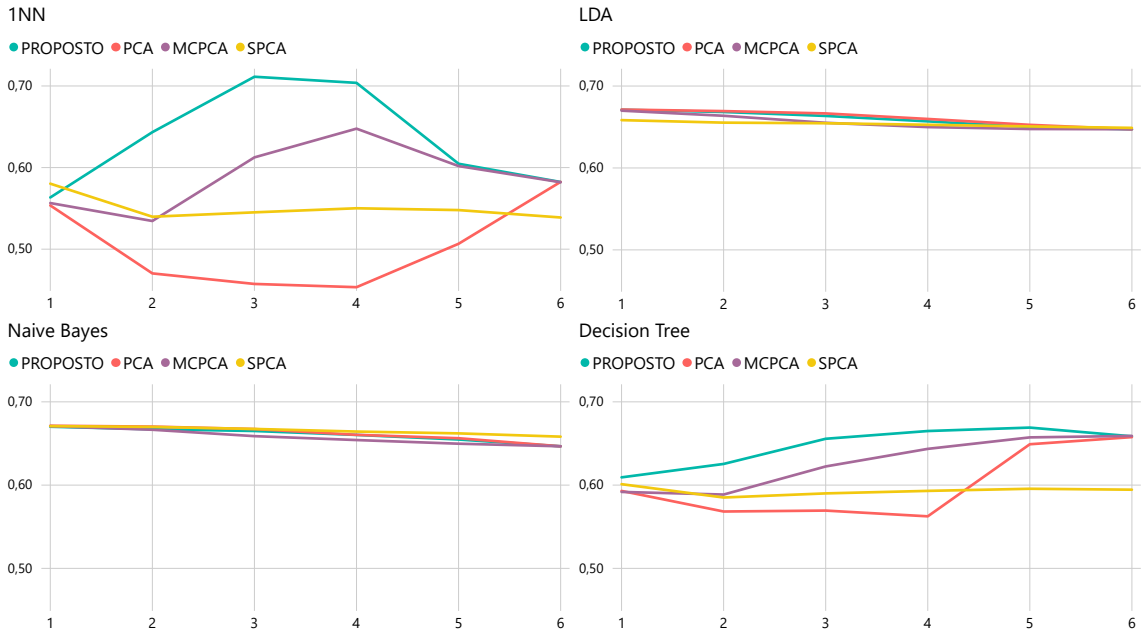


Figura 14 – Base Monk - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *Monk*, Figura 14, o método proposto conseguiu obter resultados ótimos com destaque para os classificadores 1NN e Decision Tree, em particular para o 1NN que para 3 características conseguiu uma diferença de quase 10% em relação ao MCPCA e de quase 26% para o PCA que apresentou uma queda neste intervalo. A maior taxa de acerto foi de 71,1% obtida no 1NN com 3 características, atingida somente pelo método proposto. Método proposto só não obteve taxa de acerto máxima no *Naive Bayes*, que foi de 67,1% em 1 característica com PCA, enquanto o proposto obteve 67% uma diferença ínfima que acaba caindo no intervalo de confiança [66,9%, 67,13%] tornando os resultados equivalentes.

5.1.1.5 Immunotherapy

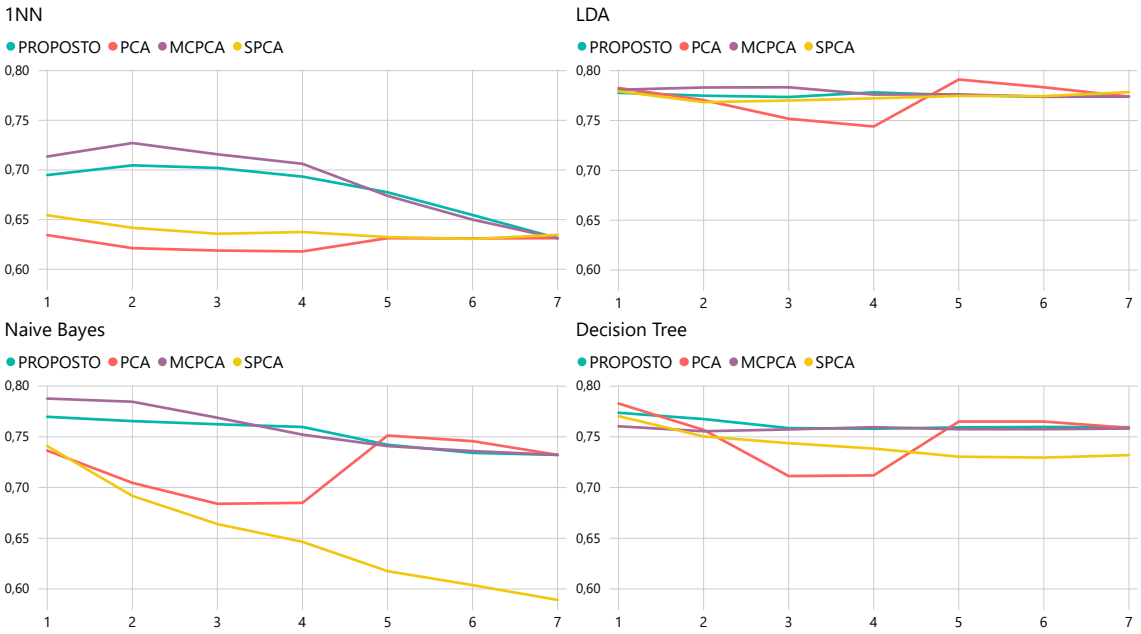


Figura 15 – Base Immunotherapy - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Para a base *Immunotherapy*, apresentada na Figura 15 o MCPCA e o proposto obtiveram taxas de acerto superiores para o 1NN e o *Naive bayes* com diferenças significativas em relação ao PCA e o SPCA. A maior taxa de acerto foi de 79,1% atingida pelo PCA no classificador LDA em 5 característica. O método proposto obteve 77,5% nestas mesmas condições. Mesmo o método não obtendo as taxas de acertos máximas nesta base ele conseguiu diferenças significativas em 2 dos 4 classificadores e não apresentou pior desempenho para nenhum dos casos.

5.1.1.6 Hillvalley

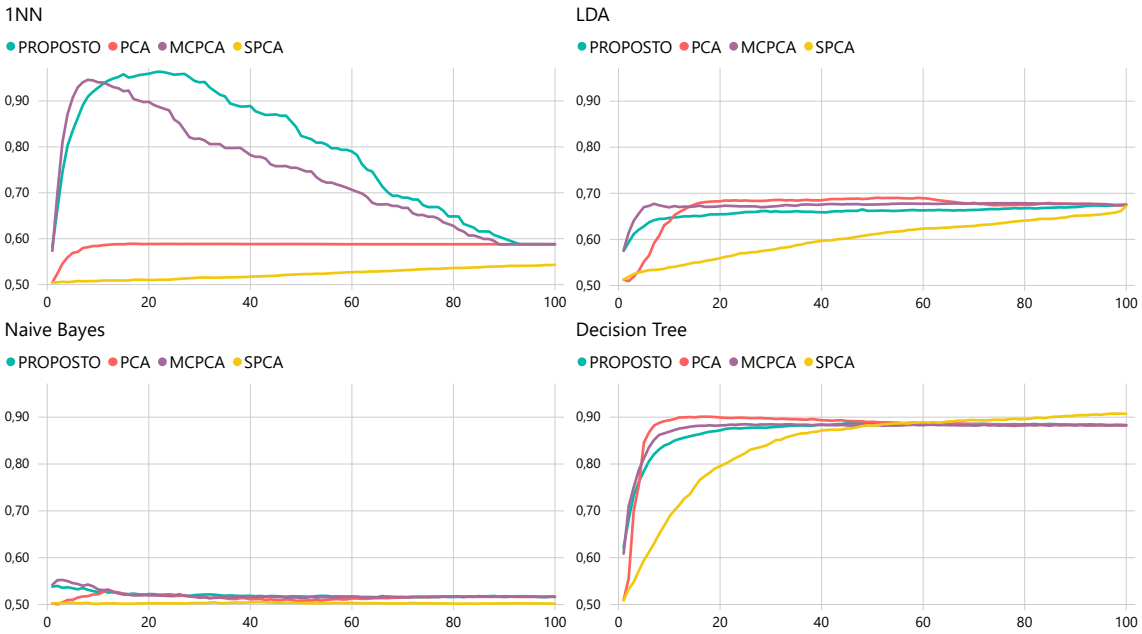


Figura 16 – Base Hillvalley - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *HillValley*, apresentada na Figura 16 o maior destaque ocorre no 1NN onde MCPCA e proposto obtiveram diferenças significativas em relação aos demais e ainda conseguiram as maiores taxas de acerto entre os classificadores. A maior taxa de acerto foi de obtida 96,4% pelo método proposto no classificador 1NN em 22 características. A maior diferença foi de 45,4% e também ocorreu com 22 características com o 1NN, onde PCA obteve 58,9% e SPCA obteve 51%. Tanto o MCPCA quanto o proposto também obtiveram maiores taxas de acerto com 1 característica em todos os classificadores.

5.1.1.7 Bank

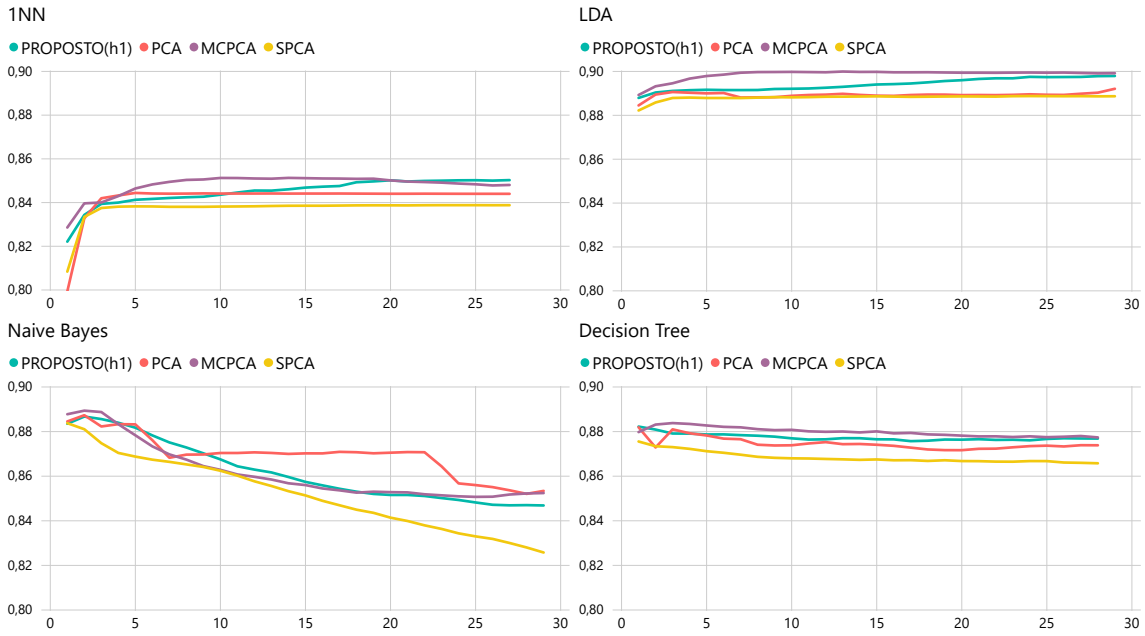


Figura 17 – Base Bank - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *Bank*, apresentada na Figura 17, diferente das demais bases mostradas anteriormente foi utilizada a Equação 2.6 por ter obtido resultados melhores, o que indica que os atributos dessas bases tem comportamentos mais próximos a uma distribuição gaussiana possuindo um desvio padrão grande. Nesta base todos os métodos conseguiram resultados semelhantes, com uma vantagem para o MCPCA seguido pelo método proposto. A maior taxa de acerto foi de 90% atingida pelo MCPCA no classificador LDA em 13 características, seguida pelo método proposto que obteve 89,3% neste mesmo cenário.

5.1.1.8 Titanic

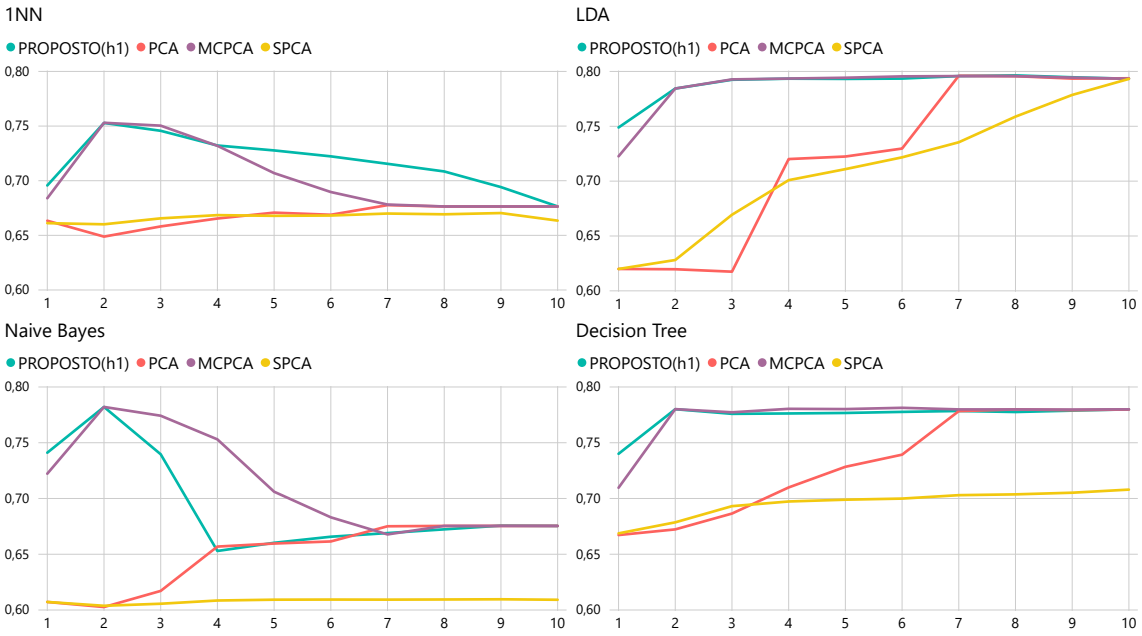


Figura 18 – Base Titanic - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *Titanic*, apresentada na Figura 18 o método proposto e o MCPCA obtiveram os melhores resultados, conseguindo atingir a maior taxa de acerto com o menor número de característica em todos os classificadores. A taxa de acerto foi de 79,6% obtida pelos: MCPCA, PCA e proposto, no LDA com 7 características, contudo, tanto o método proposto quanto o PCA obtiveram taxas de acerto expressivamente superiores no intervalo de 1 a 6 características (em todos os classificadores), onde, com 4 atributos o MCPCA e o proposto atingiram 79,3%, valor bem próximo da taxa de acerto máxima enquanto o PCA apenas 72%. É interessante notar que mesmo que MCPCA e proposto tendo obtidos as maiores taxas de acertos o proposto ainda conseguiu obter resultados melhores no espaço de uma característica.

5.1.2 Bases com mais de 2 classes

5.1.2.1 Letter

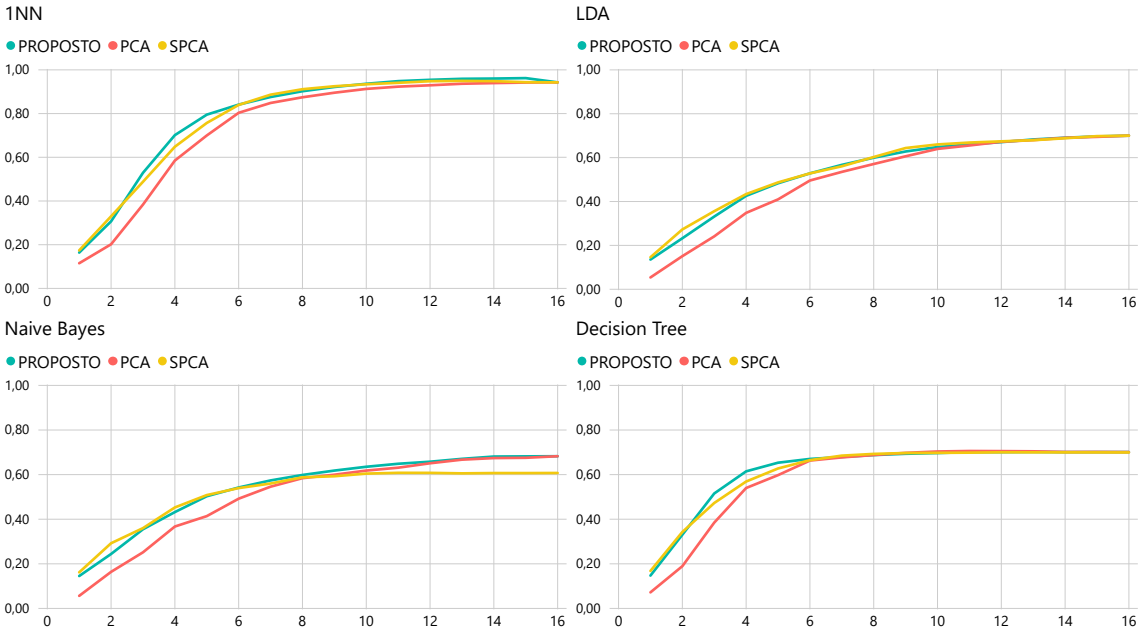


Figura 19 – Base Letter - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *Letter*, apresentada na Figura 19 os resultados dos métodos foram bem semelhantes, com destaque para o SPCA e o método proposto que conseguiram taxas de acerto maiores no espaço de menos características, sendo que o método proposto conseguiu manter uma curva estável em todos os classificadores, diferente do SPCA que apresentou perca de desempenho a partir de 8 características no classificador *Naive Bayes*, no qual outros métodos conseguiram máxima taxa de acerto.

5.1.2.2 Obs Network

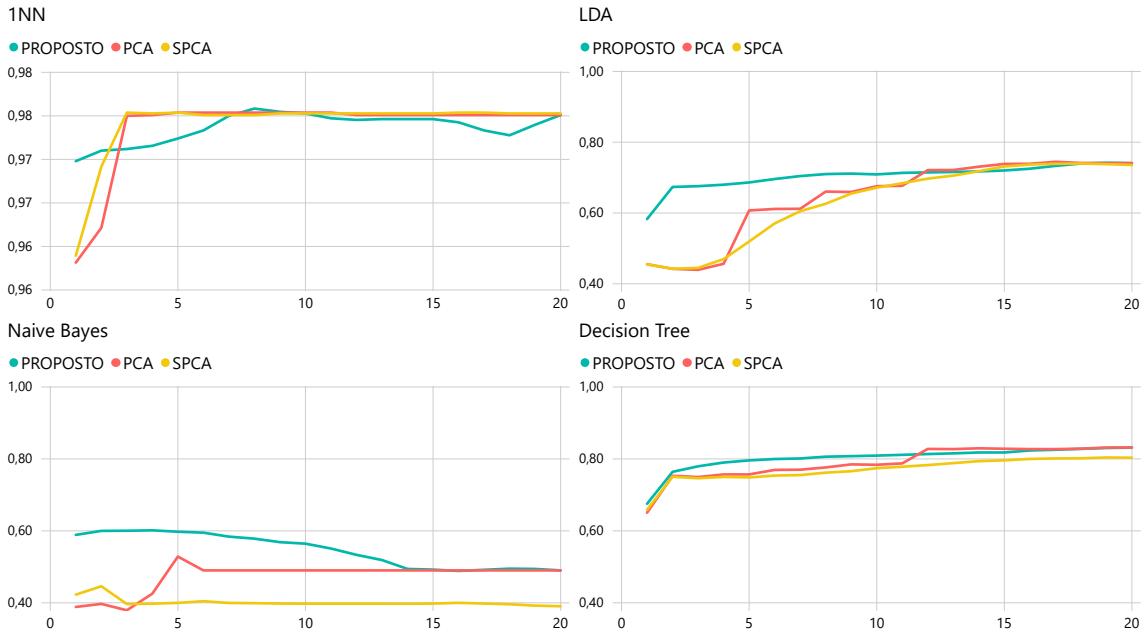


Figura 20 – Base Obs Network - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

A Figura 20 apresenta os resultados para a base *Obs Network*. O método se destacou nos classificadores *Naive Bayes*, LDA e *Decision Tree*, onde conseguiu obter resultados superiores para intervalo de 1 até 10 características. A maior taxa de acerto média foi de 97,6% no 1NN atingida pelo método proposto em 8 características. Já a maior diferença ocorreu no *Naive Bayes* em 1 característica onde o método proposto obteve 58,8% e enquanto o PCA 38,8%, uma diferença de 20%. Para o classificador 1NN o método apresentou resultados inferiores aos demais métodos para 3 e 4 características, contudo, essa diferença é de menos de 0.4%. A partir de 5 característica os resultados se equivalem no intervalo de confiança. A escala do classificador 1NN foi alterada em relação aos demais para melhor visualização dos gráficos.

5.1.2.3 Dermatology

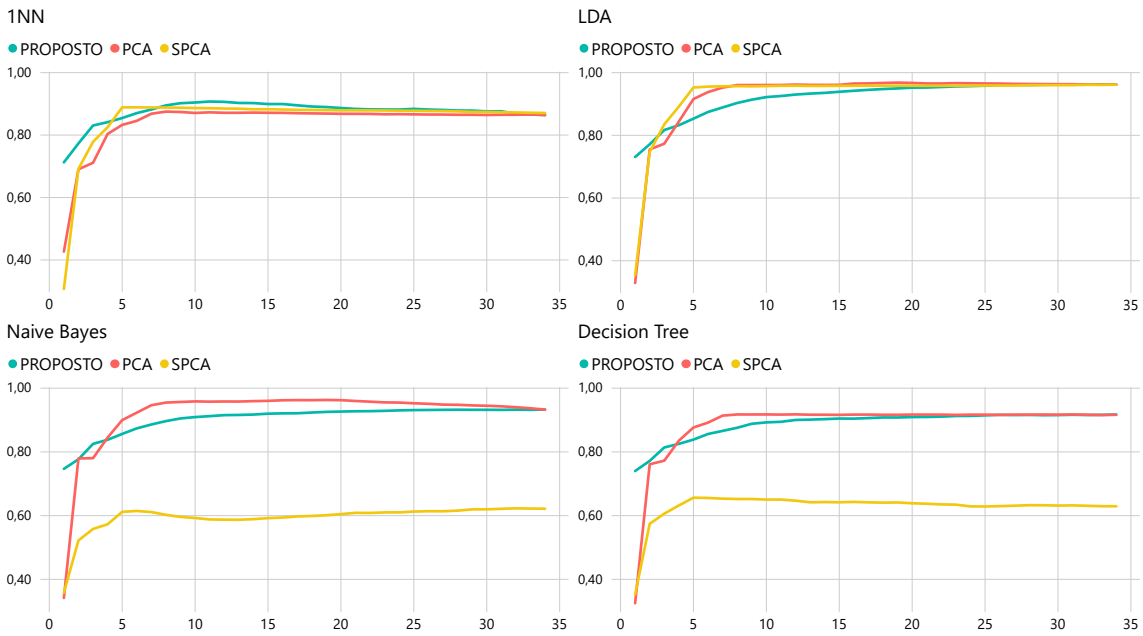


Figura 21 – Base Dermatology - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *Dermatology*, apresentada na Figura 21 o método proposto apresentou resultados positivos, mesmo não obtendo o valor máximo conseguiu resultados superiores na seleção de apenas uma característica, conseguindo uma diferença superior a 30% em relação as demais técnicas neste mesmo espaço e mantendo resultados muito próximo dos melhores nas demais características. A maior taxa de acerto foi de 96,8% atingida no LDA pelo PCA em 19 características, o método proposto obteve 95%. A maior diferença foi de 42,8% na *decision tree* com 1 característica onde o método proposto obteve 74% enquanto o PCA obteve 32,5% e o SPCA 35,2%.

5.1.2.4 Leaf

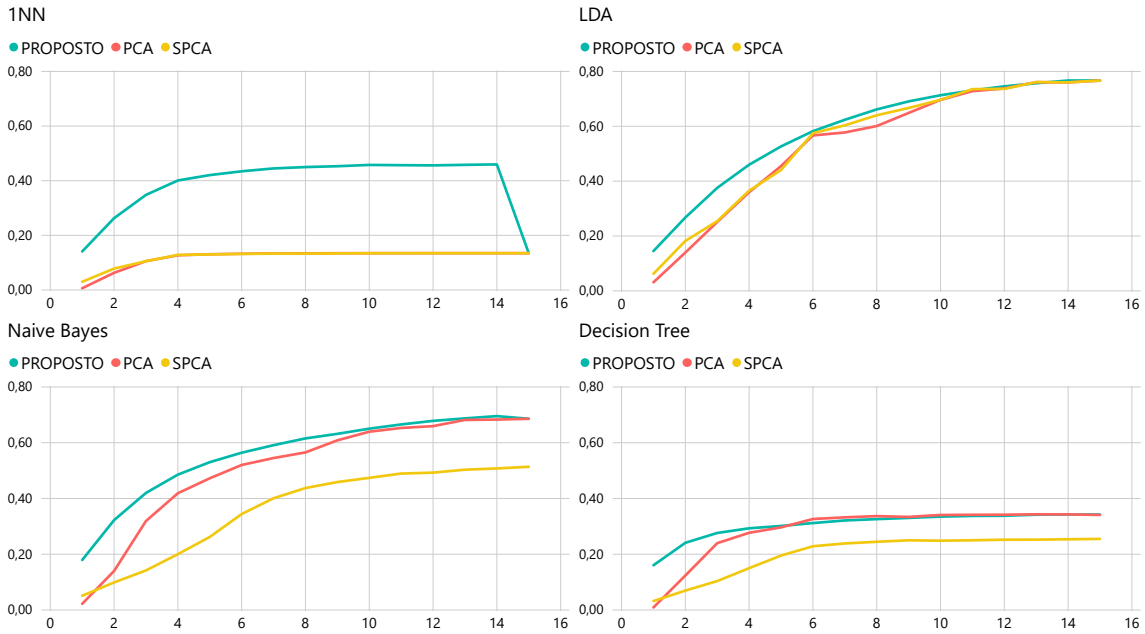


Figura 22 – Base Leaf - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *Leaf*, apresentada na Figura 22 o método proposto obteve ótimos resultados, onde conseguiu atingir a taxa de acerto máxima com menos características em todos os classificadores e manteve diferenças significativas nas seleções quem mantém menos de 4 características. A maior taxa de acerto foi de 77,7% no classificador LDA obtida por todos os métodos. Já a maior diferença ocorreu no 1NN em 14 características onde o método proposto obteve 46% enquanto o PCA e o SPCA obtiveram 13,5%. Uma diferença de 32,5%. O método só não atingiu valores superiores na árvore de decisão no intervalo de 6 a 8 características com uma diferença de cerca de 1,1%, contudo, obteve uma diferença significativa para 1 e 2 características, com pouco mais de 10% e além de ter atingido a taxa de acerto máxima neste classificador junto do PCA de 34,3% em 14 atributos.

5.1.2.5 Wine

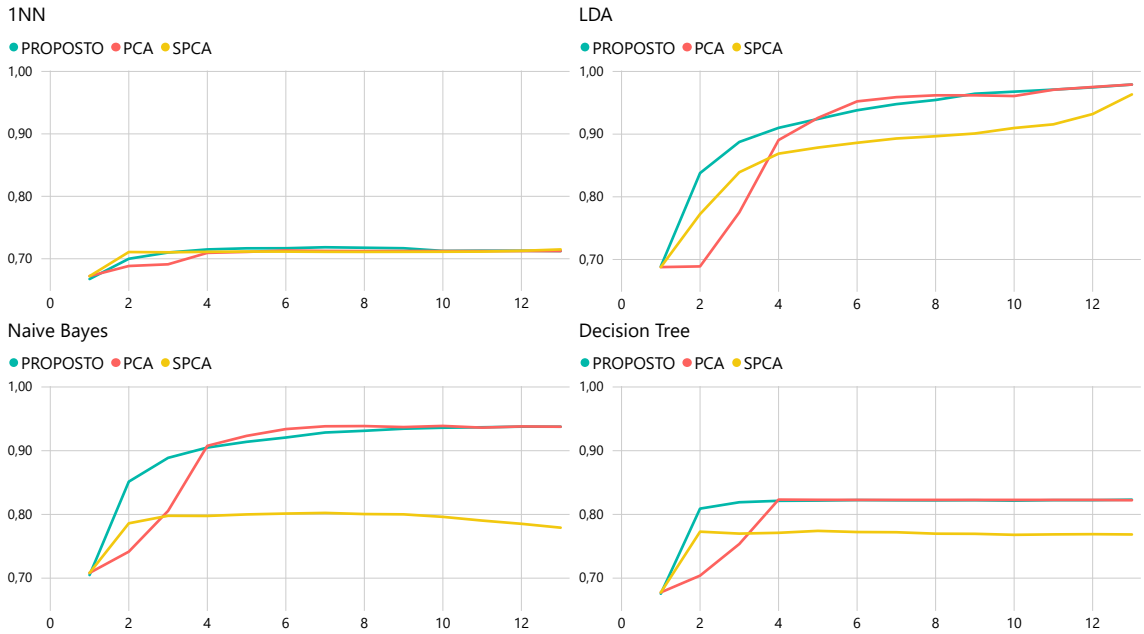


Figura 23 – Base Wine - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *Wine*, apresentada na Figura 23 o método proposto conseguiu taxas de acerto expressivas para as 3 primeiras características em 3 dos 4 classificadores. Para o Naive Bayes e LDA o PCA conseguiu maior taxa de acerto entre 6 e 8 características. A maior taxa de acerto foi de 97,9% no LDA em 13 características, atingida pelo PCA e o método proposto. A maior diferença de resultados comparando somente o método proposto com PCA foi de 14,8% onde obteve 83,8% enquanto o PCA conseguiu 69%. A maior diferença global ocorreu entre o PCA e SPCA, contudo, não apresentamos essa comparação, pois, o SPCA apresentou resultados bem inferiores aos demais métodos em 3 dos classificadores como podemos ver na Figura 23.

5.1.2.6 Wine Quality

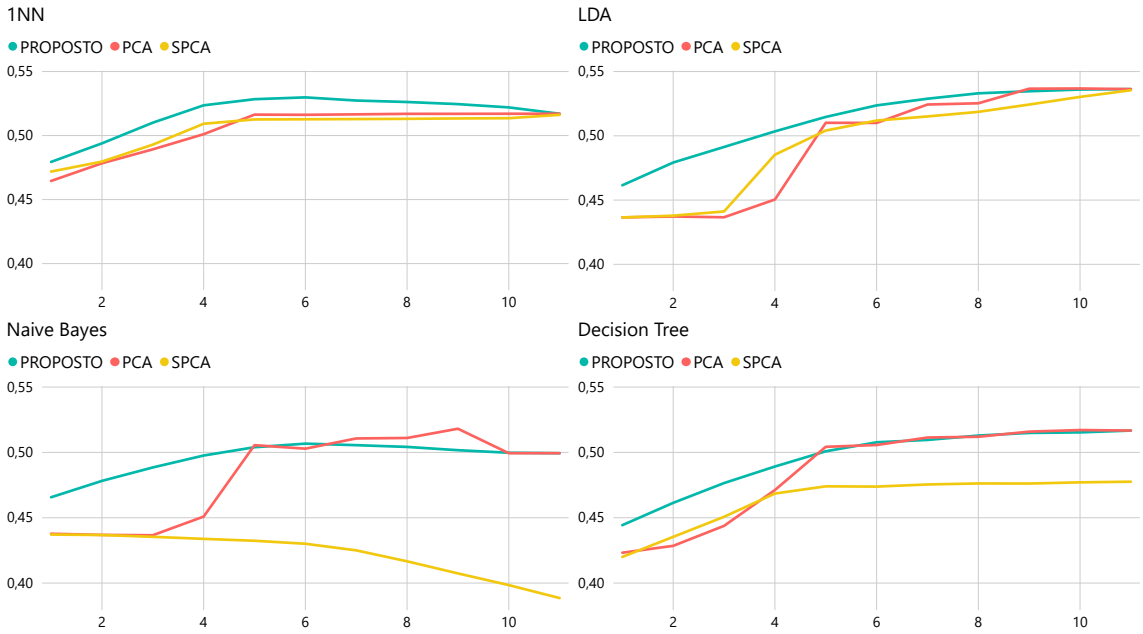


Figura 24 – Base Wine Quality Red - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na Base *Wine Quality Red*, apresentada na Figura 24 o método obteve resultados bastante favoráveis no intervalo de características abaixo de 5 em todos os classificadores. Tendo ainda conseguido obter taxas de acertos médias superiores para todas as características para o 1NN e LDA. A maior taxa de acerto foi de 53% obtida no classificador 1NN pelo método proposto em 6 características. A maior diferença entre o método proposto e os demais ocorreu no classificador LDA em 3 características, onde o proposto obteve 49,1%, o SPCA 44,1% e o PCA 43,6 %, uma diferença de 5% em relação ao SPCA e 5,5% para o PCA.

5.1.2.7 Mice

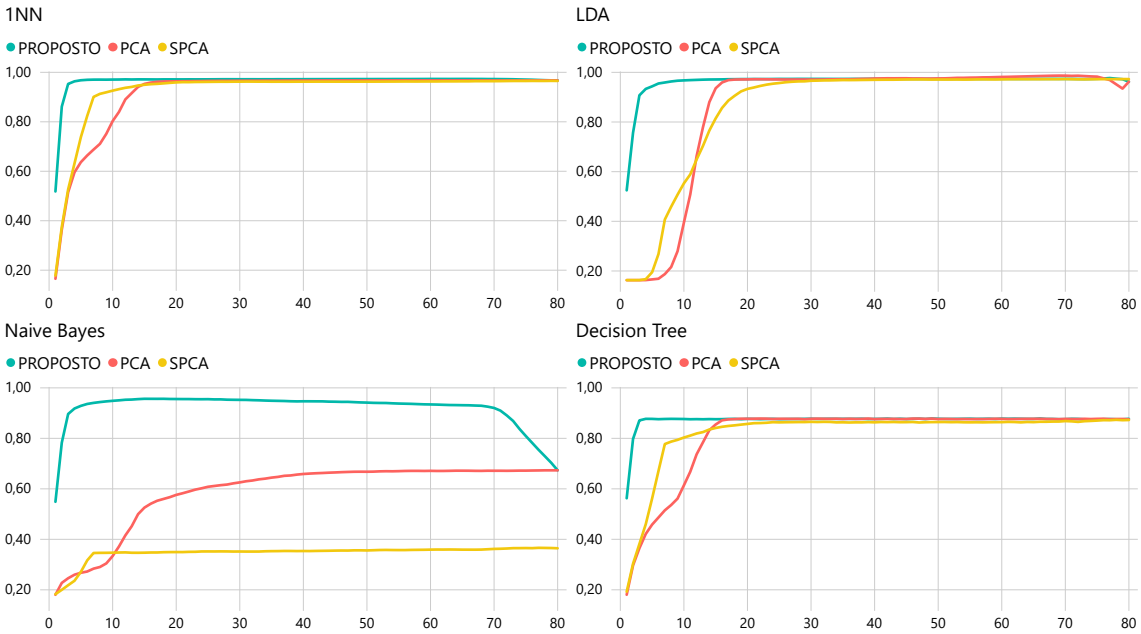


Figura 25 – Base Mice - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

A base *Mice*, apresentada na Figura 25 obteve resultados excelentes para o método proposto conseguindo resultados muito superiores no intervalo até 10 características e conseguindo atingir a maior taxa de acerto em 3 dos 4 classificadores, com exceção do LDA onde o PCA teve uma leve vantagem a partir de 59 características. A Maior diferença entre método proposto e PCA ocorreu no LDA em 6 características onde o proposto obteve 95,5% enquanto o PCA atingiu apenas 16,9% uma diferença de 78,6%. Já a diferença em relação ao supervised PCA foi de 76,7% e também ocorreu no LDA onde o proposto obteve 93,3% enquanto o supervised apenas 16,6% em 4 características.

5.1.2.8 UserKnowledge

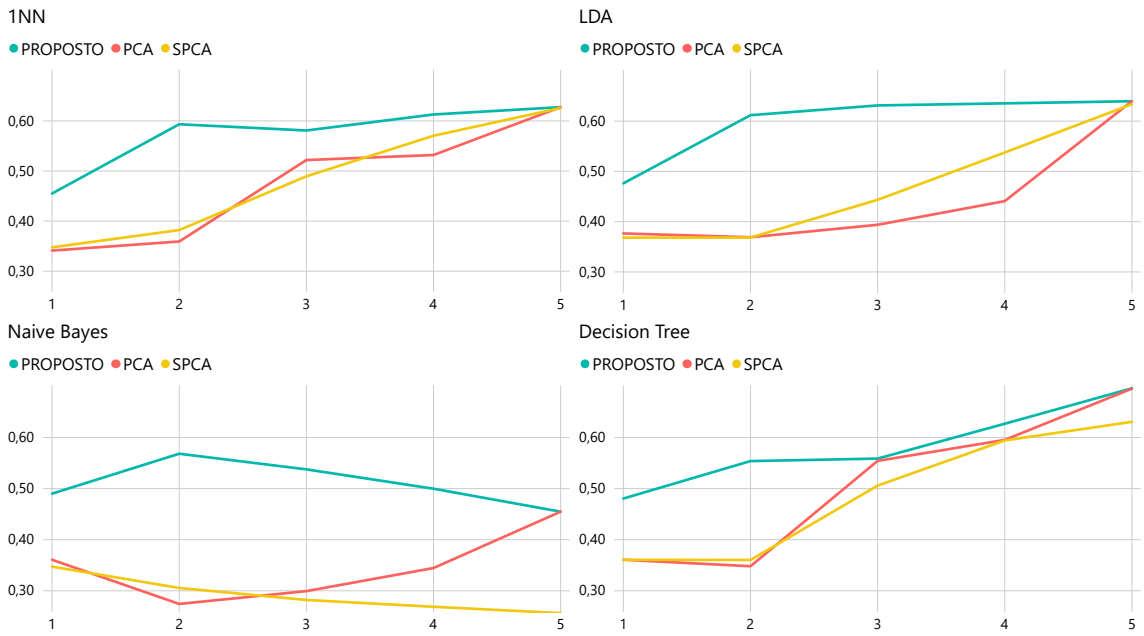


Figura 26 – Base UserKnowledge - comparação entre PCAs, taxa de acerto (eixo Y) por quantidade de características extraídas (eixo X). Fonte: Autor

Na base *UserKnowledge*, apresentada na Figura 26 o método proposto obteve resultados excelentes, obtendo taxas de acerto superiores com menos características e a maior taxa de acerto em todos os classificadores. A maior taxa de acerto foi de 69,6% atingida no LDA pelo método proposto em 5 características. Já maior diferença foi de 29,4% no *Naive Bayes* em 2 características onde o método proposto obteve 56,8% enquanto o PCA e o SPCA obtiveram 27,4% e 30,5% respectivamente.

5.2 Análise geral dos resultados

Nesta seção apresentaremos os resultados mais amplamente, utilizando tabelas para formatar os dados. As tabelas contabilizam a quantidade de vezes que um método obteve a taxa de acerto máxima com menos características em cada classificador, logo, a quantidade máxima que uma técnica pode receber é 4 por base, pois, é quantidade de classificadores utilizados. Separamos os resultados em dois intervalos: com todas as características e com uma característica. É importante salientar que pode haver empate entre os métodos, nessas situações ambos os métodos serão contabilizados. Lembrando que um resultado só é considerado superior se não houver sobreposição entre os intervalos de confiança. Para validar os resultados obtidos utilizamos o teste do sinal com grau de

significância de 5% e sempre partindo da hipótese nula de que não existe diferença de desempenho entre os métodos.

5.2.1 Todas as características

Base	Proposto	MCPCA	PCA	SPCA
BankNote	4	4	0	0
Pima	3	4	0	0
Survival	4	4	4	4
Monk	4	2	2	0
Immunotherapy	1	2	2	0
Hillvalley	1	1	2	0
Bank	1	4	1	1
Titanic	4	2	0	0
Letter	3	-	0	1
Obs Network	3	-	2	1
Dermatology	1	-	3	1
Leaf	3	-	1	0
Wine	3	-	1	0
Wine Quality	3	-	2	0
Mice	4	-	0	0
UserKnowledge	4	-	0	0
Total	46	23	20	8

Tabela 1 – Número de vezes que cada método obteve taxa de acerto máxima com qualquer número de características

Com todas as características o método proposto se saiu melhor que os demais, obtendo maior taxa de acerto com menos características em 46 casos, sendo seguido pelo MCPCA com 23, o PCA com 20 e o SPCA com 8. A Tabela 1 mostra os resultados mais detalhadamente. Para problemas de 2 classes o MCPCA obteve o melhor resultado apresentado 23 casos contra 22 do método proposto. Aplicamos o teste do sinal para avaliar se o MCPCA é melhor que o proposto utilizando a quantidade de acerto por base como métrica obtendo a seguinte situação: 3 empates, 3 resultados favoráveis a hipótese e 2 contra, portando temos que:

$$p - valor = p(x \geq 3) = 0.5001. \tag{5.1}$$

Como o grau de significância estabelecido foi de 5% e 0.5001 é maior que 0.05 não se rejeita a hipótese nula, logo, os métodos tem desempenhos iguais.

Aplicando agora o teste do sinal para o PCA e SPCA, tendo como hipótese alternativa que o método proposto possui maior desempenho, temos que:

$$p - valor_{pca} = p(x \geq 11) = 0.0288. \tag{5.2}$$

$$p - valor_{spca} = p(x \geq 13) = 0.0001. \tag{5.3}$$

Ambos os $p - valore$ foram menores que o grau de significância de 5%, portando, rejeita-se a hipótese nula, que afirma que os métodos possuem desempenhos iguais e aceita-se a alternativa, onde o método proposto tem desempenho superior.

5.2.2 Com 1 característica

Base	Proposto	MCPCA	PCA	SPCA
BankNote	4	4	0	0
Pima	3	4	0	0
Survival	4	4	0	4
Monk	3	2	2	2
Immunotherapy	4	3	2	2
Hillvalley	4	4	0	0
Bank	2	3	1	0
Titanic	4	0	0	0
Letter	0	-	0	4
Obs Network	4	-	0	0
Dermatology	4	-	0	0
Leaf	4	-	0	0
Wine	4	-	4	4
Wine Quality	4	-	0	0
Mice	4	-	0	0
UserKnowledge	4	-	0	0
Total	57	24	9	16

Tabela 2 – Número de vezes que cada método obteve maior taxa de acerto com apenas uma característica.

No intervalo de apenas uma característica o método proposto se destacou significativamente, obtendo 57 melhores taxa de acerto de 64 casos. Se limitarmos a bases que apresentam duas classes a qualidade dos resultados se mantém, onde o proposto conseguiu obter 29 resultados favoráveis contra 24 do MCPCA o segundo colocado. A Tabela 2 exibe os resultados nesse intervalo detalhadamente.

5.3 Considerações finais do capítulo

Método	com todas características	com 1 característica
Proposto	46	57
PCA	20	9
MCPCA	23	24
SPCA	8	16

Tabela 3 – Número de vezes que cada métodos obteve taxa de acerto máxima, com todas características e com apenas uma característica

Neste capítulo discorremos acerca dos resultados obtidos das comparações entre os PCAs. Como podemos observar pela Tabela 3 o método proposto obteve resultados bastante positivos conseguindo atingir o ótimo (maior taxa de acerto com menos características) em 46 dos 64 casos e em particular para o cenário de apenas uma características conseguiu obter 57 resultados favoráveis. Também aplicamos o teste de hipótese do sinal para avaliar os resultados confirmando a evidência que o método proposto se iguala ao MCPCA e tem desempenho superior ao PCA e SPCA.

6 Conclusões e trabalhos futuros

6.1 Conclusão

Este trabalho teve como objetivo desenvolver uma versão supervisionada do *Principal Component Analysis* (PCA) utilizando classificação Bayesiana para selecionar as características, ao invés da variância como o PCA. Utilizamos como base o *Minimum Classification Error PCA* (MCPCA) proposto por Carvalho, Tsang e Cavalcanti (2017) que utiliza uma métrica semelhante, mas possuindo diversas limitações para base de dados como: ser binária, gaussiana e ter prioris iguais para ambas as classes. Essas limitações se devem a utilização da distancia de mahalanobis para estimar o erro Bayesiano, sendo assim é utilizamos a classificação Bayesiana em conjunto da janela de parzen para estimar erro como é detalhado no Capítulo 3. Essa abordagem permitiu ao método contornar as limitações do MCPCA mantendo o erro Bayesiano com critério para seleção das características.

Comparamos o método proposto com o PCA, MCPCA e com *Supervised PCA* proposto por Barshan et al. (2011) que, como o nome sugere, é uma outra abordagem supervisionada do PCA, mas que não utiliza o erro de Bayes como critério de seleção. Os experimentos foram realizados em 16 bases, onde 8 eram binarias (devido as limitações do MCPCA) e as demais multi classe utilizando um *holdout* de 100 dividindo as bases em 50% para treino e 50% para teste. Em cada base foram utilizados 4 classificadores: 1NN, *Naive Bayes*, Árvore de decisão e Analise discriminante Linear (LDA) coletando a taxa de acerto por quantidade de características extraídas. Apresentamos os resultados individuas em gráficos na seção 5.1.

Obtivemos resultados bastante favoráveis ao nosso método, onde o mesmo obteve maior taxa de acerto com qualquer número de características em 46 dos 64 casos, enquanto o PCA, MCPCA e SPCA obtiveram 20, 23 e 8 respectivamente como podemos observar na Tabela 1. O método também se destacou de no cenário de apenas uma característica, onde obteve a maior taxa de acerto em 57 dos 64 casos, enquanto o PCA, o MCPA e o SPCA obtiveram 9, 24 e 16 respectivamente apresentados na Tabela 2. Acreditamos que o sucesso do método proposto neste cenário se deve ao fato que a primeira características é, em teoria, a que possui menor erro Bayesiano, portanto sendo mais discriminante que as demais.

Notável observar a diferença nos resultados obtidos pelo método proposto e MCPCA (este mesmo sendo aplicados somente as bases binárias, ainda obteve resultados melhores que o PCA e SPCA) demonstrando que o erro Bayesiano é um critério de seleção bastante

discriminante. Para validar os resultados aplicamos o teste de hipóteses dos sinais, que demonstrou que o método proposto é equivalente ao MCPCA mesmo este tendo obtido mais maiores taxas de acerto para qualquer característica nas bases binárias, 23 contra 22 do proposto. Já para o PCA e SPCA o teste evidenciou uma diferença significativa entre eles e o proposto, o que indica superioridade do método. Com isso podemos definir a pesquisa como bem sucedida atingindo todos os objetivos definidos.

6.2 Trabalhos futuros

Para trabalhos futuros testar novas métricas de avaliação de erros e outras formas de se obter os parâmetros de suavização que melhor se ajustem aos dados observados, conseguindo assim obter taxas de acerto ainda maiores para as situações onde não se atingiu o melhor resultado. Também desejamos validar o método como técnica de avaliação de características, combinando-o com outras técnicas de extração de características como o *Supervised PCA* (SPCA) e a Análise Discriminante Linear (LDA) avaliando se haverá melhoras nas taxas de acertos para esses métodos também.

É desejável também encontrar alternativas que permitam estimar o erro de Bayes sem utilizar a janela de Parzen, pois, a mesma gera um custo considerável ao tempo de execução do método dependendo do tamanho da base de dados e da quantidade de características tornando a etapa de treinamento do algoritmo mais lenta.

Referências

BARSHAN, E. et al. Supervised principal component analysis:visualization, classication and regression on subspaces and submanifolds. *Pattern Recognition*, n. 10, p. 9–11, 2011. Citado 3 vezes nas páginas 15, 21 e 54.

BELLMAN, R. *Dynamic Programming*. [S.l.]: Courier Corporation, 1957. Citado na página 18.

BISHOP, C. M. *Pattern Recognition And Machine Learning*. [S.l.]: Springer, 2006. Citado 2 vezes nas páginas 17 e 19.

CARVALHO, T. B. A.; TSANG, I. R.; CAVALCANTI, G. D. da C. Principal component analysis for supervised learning: a minimum classication error approach. *Journal of Information and Data Management*, v. 8, p. 131–145, 2017. Citado 2 vezes nas páginas 15 e 54.

DEEPAI. *Feature Extraction*. 2018. Last accessed 20 December 2018. Disponível em: <https://deepai.org/machine-learning-glossary-and-terms/feature-extraction>. Citado na página 19.

DIAS, M. S. *Regressão Construtiva Em Variedades Implícitas*. Tese (Doutorado) — Pontifícia Universidade Católica Do Rio De Janeiro - PUC-RIO, 2013. Citado 2 vezes nas páginas 14 e 18.

DY, J. G. Unsupervised feature selection. p. 19–39, 2007. Citado na página 14.

KEINOSUKE, F. *Introduction to Statistical Pattern Recognition Second Edition*. 2th. ed. [S.l.]: Academic Press, 1990. Citado 2 vezes nas páginas 24 e 26.

KINGSFORD, C.; SALZBERG, S. L. What are decision trees? *Nat. Biotechnol*, v. 26, p. 1011–1013, 2008. Citado na página 32.

KOTSIANTIS, S. B. Supervised machine learning: A review of classification techniques. *Informatica*, v. 31, p. 249–268, 2007. Citado na página 17.

MACEDO, D. C. D. *Comparação Da Redução De Dimensionalidade De Dados Usando Seleção De Atributos E Conceito De Framework: Um Experimento No Domínio De Clientes*. 2012. Citado na página 14.

MAćKIEWICZ, W. R. A. Principal components analysis (pca). *Computers Geosciences*, v. 19, p. 118–173, 1993. Citado na página 19.

SILVERMAN, B. W. *Density Estimation for Statistics and Data Analysis*. [S.l.]: Monographs on Statistics and Applied Probability, 1986. Citado na página 23.

SPRUYT, V. *The Curse of Dimensionality in classification*. 2014. Último acesso em 18/01/2019. Disponível em: <http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/>. Citado 2 vezes nas páginas 9 e 18.

SPRUYT, V. *Feature extraction using PCA*. 2014. Disponível em: <http://www.visiondummy.com/2014/05/feature-extraction-using-pca/>. Citado 2 vezes nas páginas 9 e 20.

TAN, P.-N. et al. *Introduction to Data Mining*. 2th. ed. [S.l.]: Pearson, 2018. Citado na página 14.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining*. 1th. ed. [S.l.]: Addison-Wesley Longman Publishing Co, 2005. Citado na página 17.

TERRELL, G. R.; SCOTT, D. W. The variable kernel density estimation. *Nat. Biotechnol*, v. 20, p. 1236–1265, 1992. Citado na página 22.

TORKKOLA, K. Linear discriminant analysis in document classification. *IEEEICDM Workshop on Text Mining*, p. 800–806, 2001. Citado na página 31.

VARGAS, S. A. *Previsão da distribuição da densidade de probabilidade da Geração de Energia Eólica usando técnicas não paramétricas*. Tese (Doutorado), 2015. Citado na página 23.

WANDERLEY, M. F. B. *Estudos em Estimação de Densidade por Kernel: Métodos de Seleção de Características e Estimação do Parâmetro Suavizador*. Tese (Doutorado), 2013. Citado na página 32.