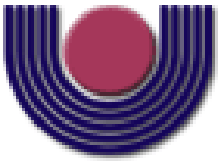


UNIOESTE – Universidade Estadual do Oeste do Paraná

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

Um Estudo Comparativo entre Métodos de Extração e de Seleção de Características

Victor Hugo Röhsig Silva

CASCAVEL

2012

VICTOR HUGO RÖHSIG SILVA

UM ESTUDO COMPARATIVO ENTRE MÉTODOS DE EXTRAÇÃO E DE SELEÇÃO DE CARACTERÍSTICAS

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel

Orientador: Prof. Dr. Clodis Boscarioli

CASCAVEL

2012

VICTOR HUGO RÖHSIG SILVA

UM ESTUDO COMPARATIVO ENTRE MÉTODOS DE EXTRAÇÃO E DE SELEÇÃO DE CARACTERÍSTICAS

Monografia apresentada como requisito parcial para obtenção do Título de *Bacharel em Ciência da Computação*, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Dr. Clodis Boscarioli (Orientador)
Colegiado de Ciência da Computação,
UNIOESTE

Prof. Dr. Jerry Adriani Johann
Centro de Ciências Exatas e Tecnológicas,
UNIOESTE

Prof.ª Dra. Rosangela Villwock
Colegiado de Matemática,
UNIOESTE

Cascavel, 21 de novembro de 2012.

DEDICATÓRIA

Dedico esta monografia a todos que contribuíram, direta ou indiretamente, para a sua elaboração, a minha família, meus colegas, amigos, banca examinadora, orientador e professores.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

BRD	Busca em Amplitude
DM	*Data Mining*
EB	Espaço de Busca
EC	Extração de Características
JAMA	*Java Matrix Package*
JDBC	*Java Database Connectivity*
KDD	*Knowledge Discovery in Databases*
LVQ	*Learning Vector Quantization*
MLP	*Multilayer Perceptron*
PCA	Análise de Componentes Principais
RBF	*Radial Basis Function*
PD	Problema da Dimensionalidade
RD	Redução de Dimensionalidade
SBS	Busca Sequencial Regressiva
SC	Seleção de Características
SFS	Busca Sequencial Progressiva
SVD	*Singular Value Decomposition*
YADMT	*Yet Another Data Mining Tool*

Sumário

Resumo

A redução da dimensionalidade de um conjunto de dados tem o intuito tanto de diminuir a quantidade de recursos utilizados em Reconhecimento de Padrões, bem como tentar aumentar a precisão de aplicações para tal. Nesta monografia fez-se um estudo comparativo entre métodos de Redução de Dimensionalidade, sendo comparados um método de Extração de Características, a Análise de Componentes Principais (PCA), e três métodos de Seleção de Características, a Busca Sequencial Progressiva (SFS), Busca Sequencial Regressiva (SBS) e a Busca em Amplitude (BRD). A comparação foi realizada em diferentes bases de dados, como critério de avaliação dos métodos adotou-se o tempo de execução do método e a precisão obtida pelo classificador. Por fim, uma discussão sobre a avaliação experimental é apresentada, onde se puderam observar os benefícios trazidos pela redução da dimensionalidade.

Palavras-chave: Redução de Dimensionalidade, Análise de Componentes Principais, *Yet Another Data Mining Tool.*

Capítulo 1

Introdução

As empresas e organizações vêm se mostrando muito eficientes na captura, organização e armazenamento de dados. Contudo, muitas ainda têm dificuldades no uso dessa enorme fonte de conhecimento. A mineração de dados vem para revelar conhecimento que possa guiar decisões com certo nível de precisão.

De acordo com (Michael; Berry, 1997) Mineração de Dados é a análise de dados, por meios automáticos ou semiautomáticos, em grandes quantidades de dados, com o objetivo de descobrir regras ou padrões interessantes. Para (Han; Kamber, 2001), o conceito de que um sistema de mineração de dados pode minerar automaticamente e encontrar todos os aspectos valiosos que estão ocultos em uma base de dados sem a intervenção ou direcionamento humano, está muito errado.

Neste trabalho é considerada a definição proposta por (Cortes; Porcaro; Lifschitz, 2002) de que "Mineração de dados é um processo altamente cooperativo entre homens e máquinas, que visa à exploração de grandes bancos de dados, com o objetivo de extrair conhecimentos através do reconhecimento de padrões e relacionamento entre variáveis, conhecimento esses que possam ser obtidos por técnicas comprovadamente confiáveis e validados pela sua expressividade estatística".

Mas somente métodos estatísticos não são suficientes para extrair o conhecimento de um banco de dados para a finalidade de mineração, surgindo então uma área de estudo denominada Descoberta de Conhecimento em Bases de Dados (*KDD – Knowledge Discovery in Databases*), que, de acordo com (Thomé, 2008), se caracteriza por ser um processo não trivial, que busca gerar conhecimento que seja novo e potencialmente útil para aumentar os ganhos, reduzir os custos ou melhorar o desempenho do negócio, por meio da procura e da identificação de padrões a partir de dados armazenados em bases de dados muitas vezes dispersas e inexploradas.

Segundo (Elmasri; Navathe, 2005), o *KDD* é composto de seis fases: seleção de dados, limpeza, enriquecimento, transformação ou codificação – etapas de pré-processamento –, *Data Mining* (DM) e construção dos relatórios de apresentação.

A fase de seleção de dados é onde itens específicos em um banco de dados são selecionados para o Processo de Descoberta do Conhecimento.

A fase de limpeza, também chamada de pré-processamento, corrige as inconsistências encontradas nos dados para garantir a confiabilidade nos dados que serão utilizados pela mineração. Segundo (Navega, 2002), as bases de dados são dinâmicas, incompletas, redundantes, ruidosas e esparsas, necessitando de um pré-processamento para "limpá-las".

A codificação ou transformação é o processo onde a quantidade de dados é reduzida, agrupando valores em outras categorias sumarizadas.

A fase de DM é o núcleo do processo. De acordo com Thomé (2008), a ideia central em DM é a de que seus algoritmos sejam capazes de identificar a existência de padrões e relacionamentos desconhecidos, que ao serem analisados posteriormente, possam suscitar e induzir a geração de hipóteses úteis e relevantes para o usuário.

E, por último, a fase de apresentação, é onde os conhecimentos extraídos do processo são apresentados de forma a serem compreendidos.

O pré-processamento mostra-se uma importante área de estudo. Nesta fase os dados são selecionados e/ou transformados para garantir um melhor desempenho do processo. Caso sejam selecionados dados inconsistentes ou fora do âmbito do objetivo, o conhecimento gerado pode ser falho ou não corresponder à realidade.

Outra forma de melhorar o desempenho do processo utilizando o pré-processamento é através da Redução de Dimensionalidade, pois para representar toda a variabilidade do banco de dados nem sempre é necessário utilizar o número total de dimensões (atributos/ características).

Este trabalho apresenta a comparação de duas abordagens para a Redução de Dimensionalidade, sendo elas Seleção de Características e Extração de Características, verificando se através de transformação linear (Extração Características) pode-se obter um conjunto reduzido de características que garanta uma maior precisão de um classificador do que um conjunto de mesma dimensão das características originais (Seleção de Características).

1.1 Justificativas

Utilizar um conjunto reduzido de dados que melhor expressem o seu comportamento é benéfico para:

- Reduzir o custo do aprendizado, pois o custo é diretamente proporcional ao tamanho do conjunto de dados;
- Aumentar a precisão do algoritmo;
- Gerar modelos mais compactos;
- Diminuir os recursos computacionais necessários.

1.2 Objetivos

O objetivo deste trabalho foi estudar a Redução de Dimensionalidade, fazendo um comparativo entre a Seleção de Características e a Extração de Características. Sendo os objetivos específicos:

- Estudo e implementação de método de Extração de Características através de transformação linear, sendo ele a Análise de Componentes Principais (PCA).
- Estudo e implementação de métodos de Seleção de Características.
- Avaliação Experimental dos métodos através da acuidade obtida pelo classificador *Naive Baye*.

1.3 Organização do Documento

O texto está organizado da seguinte maneira:

O Capítulo 2 apresenta o conceito de Redução de Dimensionalidade, também apresentando os métodos e modelos escolhidos de redução.

O Capítulo 3 apresenta a forma como foram desenvolvidos e implementados os métodos.

O Capítulo 4 apresenta a avaliação experimental, e expõe os resultados. E por fim, o Capítulo 5 apresenta as considerações finais do trabalho.

Capítulo 2

Redução de Dimensionalidade

O termo dimensionalidade é associado ao número de características de uma representação, ou seja, a dimensão do espaço de características (atributos). As principais razões para que a dimensão seja a menor possível são duas, o custo de medição e a precisão do classificador.

Segundo (Martins JR., 2004), a Redução de Dimensionalidade (RD) é necessária para evitar o Problema da Dimensionalidade (PD) que afeta a precisão do classificador. De acordo com (Jain; Duin; Mao, 2000), o PD ocorre quando o número de amostras de treinamento para que um classificador obtenha um bom desempenho é uma função monotonicamente crescente da dimensão dos padrões (o número de características). Em reconhecimento estatístico de padrões, o volume de amostras necessárias para a classificação cresce exponencialmente com a dimensionalidade (Perlovsky, 1998).

Na Figura 2.1 (adaptada de (De Campos, 2001)), pode-se observar o comportamento de um classificador à medida que se acrescente características, desde que estas sejam acrescentadas em ordem de relevância.

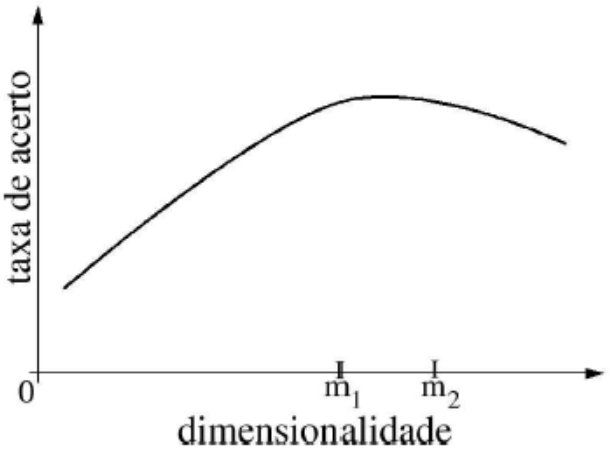


Figura 2.1 – Problema da Dimensionalidade

Em um primeiro momento (na Figura 2.1, dimensionalidade compreendida entre *0* e *m1*), a taxa de acerto é diretamente proporcional à dimensionalidade, ou seja, a adição de novas características melhora o desempenho do classificador, pois não foi alcançado o número mínimo de informações suficientes para distinguir as classes de padrões. (De Campos, 2001)

Em um segundo momento (na Figura 2.1, dimensionalidade compreendida entre *m1* e *m2*), o aumento da dimensionalidade não altera (ou altera sutilmente) a precisão do classificador, porém, características irrelevantes ou redundantes ao problema são também processadas causando um desperdício de recursos e aumentando o custo de medição (De Campos, 2001).

Em um terceiro momento (na Figura 2.1, dimensionalidade maior que *m2*), a adição de características prejudica a classificação. É neste ponto que o PD ocorre efetivamente, onde a quantidade amostral é insuficiente com relação à quantidade de características, fazendo com que a taxa de acerto (precisão) seja reduzida, ou seja, o desempenho do algoritmo tende a degradar-se. Assim, o ideal é que a quantidade de características utilizada seja escolhida de tal maneira que a precisão seja maximizada (De Campos, 2001).

A RD também é necessária para retirar (ou atenuar) dados irrelevantes e redundantes (Liu; Motoda, 1998), para que estes não influenciem na precisão e aumentem o custo da classificação. As características irrelevantes são aquelas que não possuem informação útil para o problema em questão, por exemplo, um valor constante para todos os exemplos em certo atributo.

Já as características redundantes possuem a mesma informação útil para a tarefa em questão, como exemplo, considere dois atributos contendo os mesmos valores para cada instância ou, que podem ser calculados um a partir do outro.

Faz-se também importante a RD em bases de dados muito grande, onde recursos computacionais tornam-se insuficientes, não sendo possível que algum algoritmo os processe em tempo hábil, ou, a falta de espaço para armazenar todos os dados de uma única vez (Liu; Motoda, 1998).

A RD pode ser realizada, basicamente, de duas formas: Seleção de Características (SC) e Extração de Características (EC). Em linhas gerais, a EC gera novas características por meio de transformações ou combinações lineares das características originais. Enquanto, os métodos de SC, como o próprio nome diz, selecionam, segundo algum critério, o melhor subconjunto das características originais.

A escolha entre SC e EC depende do conjunto de dados e do domínio da aplicação. A SC mantém a sua interpretação física original, mantendo as propriedades originais. Já a EC pode prover um conjunto de características de melhor discriminação que o subconjunto gerado pela SC. Entretanto, pode perder-se a interpretação física original. A seguir, a SC e a EC são apresentadas em mais detalhes.

2.1 Seleção de Características

A Seleção de Características (SC) tem como objetivo a RD. Segundo (Liu; Motoda, 1998), a seleção de características é um processo de redução de dimensionalidade que escolhe um subconjunto de características de acordo com um critério, ou seja, será resultado desse processo um subconjunto que melhor satisfaz certo critério.

Segundo (Santoro, 2005), existem dois modelos principais para SC: Filtro e *Wrapper*, existindo ainda o tipo Integrado que é menos abordado. O modelo Integrado ocorre quando o processo para SC é embutido ao classificador (Lee, 2000).

No modelo *Wrapper* a precisão preditiva é avaliada pelo próprio classificador, sendo assim os métodos de SC que utilizam *Wrapper* estão atreladas a aplicação. Segundo (Siedlecki; Sklansky, 1988) e (Kohavi, 1995), estes métodos obtém a garantia de melhores resultados, maximizando a precisão de um classificador, porém, demandam de um maior processamento.

Modelos para SC do tipo Filtro atuam apenas sobre os padrões determinados pelas características selecionadas. Por isso, modelos do tipo Filtro demandam de um menor processamento que os do tipo *Wrapper*, pois, inexiste o classificador que demandaria treinamento e posterior avaliação da precisão obtida a cada subconjunto candidato (gerado).

SC pode ser visto como um Problema de Busca (Lee, 2000), (Lee, 2005). O Espaço de Busca (EB) contém todos estados (configurações) possíveis. Em cada estado podemos ter de zero a todas as características selecionadas, isto é, o Espaço de Busca compreende todas as combinações possíveis das características. Pode se organizar o espaço por camadas, onde na primeira camada tem-se o estado contendo zero característica selecionada (C_N^0), na segunda camada temos o conjunto dos estados contendo uma característica cada (C_N^1), assim sucessivamente, até a última camada que terá um estado contendo todas as características (C_N^N). Totalizando assim um Espaço de Busca de $C_N^0+C_N^1+...+C_N^N = 2^N$ estados.

Conforme (Liu; Motoda, 1998), diferentes métodos de SC são combinações de quatro fatores: direção, estratégia, avaliação da solução e critério de parada, discutidos a seguir.

2.1.1 Direção da Busca

O Espaço de Busca (EB) contem pelo menos dois estados (conjuntos). Sendo eles o estado vazio, onde nenhuma característica é selecionada, e o estado completo, onde todas as características são selecionadas. Estes estados são os extremos e determinam o conjunto inicial de estados dos algoritmos de acordo com a direção utilizada. As possíveis direções são (Liu; Motoda, 1998):

- Progressiva (*forward*): Inicia pelo conjunto vazio e a cada iteração características são escolhidas e unidas ao conjunto anterior.
- Regressiva (*backward*): Começa a busca pelo conjunto completo e a cada iteração é escolhida uma característica e removida.
- Aleatória: Explora o espaço de busca sem ordem definida.
- Bidirecional: Ambas as direções são exploradas. Podem ser executadas concorrentemente ambas as direções, onde em cada direção metade do espaço de busca é explorado.

O número de iterações para que o conjunto seja encontrado depende do conjunto de dados e da direção de busca, pois caso o conjunto alvo contenha muitos atributos a busca regressiva encontra mais rapidamente este conjunto. Caso o conjunto alvo esteja próximo do conjunto vazio a busca progressiva a encontrará mais rapidamente. Quando a solução está próxima do centro do espaço de busca é indicado o uso da estratégia progressiva, pois, tende ser mais custoso avaliar um subconjunto com muitas características, conforme atestam (Jain; Zongker, 1997) e (Zongker; Jain, 1996).

2.1.2 Estratégia da Busca

Avaliar todo o espaço de busca pode tornar-se impraticável, pois este é exponencial com relação à dimensionalidade. Para contornar este problema surgem estratégias para controlar como o espaço é explorado. De acordo com (Liu; Motoda, 1998), as estratégias de busca são:

- Força Bruta: Todo o espaço de busca é explorado, podendo assim garantir o resultado ótimo do algoritmo de acordo com o critério utilizado.

- Completa: Algoritmos são ditos completos quando garante o resultado ótimo da solução. Mas não necessariamente é obtido pela força bruta.

- Heurística: Explora somente os caminhos mais promissores (nodos que obtiveram melhor avaliação). Buscam obter a melhor solução com o menor custo computacional.

- Não-determinísticos (aleatório): O espaço de busca é explorado de forma aleatória. Assim, em cada execução pode-se obter uma solução diferente. Esses algoritmos são indicados quando o conjunto de características é excessivamente grande.

Somente a força bruta é capaz de garantir a solução ótima, para qualquer caso. Mas, para maior desempenho pode-se ignorar estados que contiverem características não discriminantes (menos promissores).

2.1.3 Avaliação da Busca

A função critério é quem efetivamente qualifica uma solução (estado). Assim a solução esta intimamente ligada ao critério utilizado. Por mais que se varra todo o espaço de busca testando todas as possibilidades, se o critério não for condizente com o método de classificação a solução poderá ser um desastre.

Conforme (Liu; Motoda, 1998), critérios podem ser agrupados a partir da métrica utilizada. São elas:

- Medidas de Precisão: Faz o calculo da precisão preditiva, ou seja, utiliza-se um classificador para verificar a precisão. Pode-se utilizar o próprio classificador como critério, neste caso, configurando o modelo *Wrapper*.

- Medidas de Consistência: Procuram o subconjunto de características que mantenha a máxima consistência com relação ao conjunto. Uma inconsistência são duas amostras com os mesmos valores, mas designadas a classes diferentes. Usar essas métricas pode ser útil para remoção de características irrelevantes e redundantes.

- Medidas de Informação: Calculam o ganho de informação ao selecionar uma característica. O ganho de informação é a diferença entre a incerteza do conjunto original e a incerteza esperada pela adição (ou remoção) de uma característica.

- Medidas de Distância: Calculam as distâncias entre as classes ao selecionar uma característica, ou seja, quanto ficará distante as classes após a seleção (ou remoção) da característica.

- Medidas de Dependência: Calcula o quanto uma característica é relacionada ao valor da classe. Se estas forem estatisticamente independentes, a remoção da característica não alterará a separabilidade entre as classes. Se a cada valor da característica estiver associado a uma classe, a dependência será máxima, e a característica ajudará a determinar a classe.

2.1.4 Critério de Parada

Outro critério determinante para a solução é o quando o processo chega ao fim. Segundo (Liu; Yu, 2005), são utilizados quatro critérios:

- Todo o espaço de busca foi avaliado.

- O número alvo de características foi alcançado ou o número máximo de iterações.

- Adição (ou remoção) de características não produz um conjunto melhor.

- O conjunto suficientemente bom é encontrado. Por exemplo, se o conjunto encontrado produza uma taxa de erro na classificação abaixo de certo parâmetro.

2.1.5 Avaliação da Solução

Uma maneira simples para avaliação da solução é medir diretamente o resultado usando o conhecimento prévio sobre os dados. Caso saiba-se de antemão as características relevantes, como no caso de dados sintéticos, pode-se comparar este conjunto conhecido de características com as características selecionadas. O conhecimento sobre as características irrelevantes ou redundantes também pode ajudar. Espera-se que eles não sejam selecionados. Em aplicações do mundo real, no entanto, geralmente não se têm conhecimento prévio tal. Assim, há que confiar em alguns métodos indiretos, monitorando a mudança de desempenho

de mineração com a mudança de características. Por exemplo, se usar uma taxa de erro de classificação como um indicador de desempenho de uma tarefa de mineração, para um subconjunto de características selecionadas, pode-se simplesmente conduzir o "antes e depois" do experimento para comparar a taxa de erro do classificador sobre o conjunto completo de características e a taxa de erro no subconjunto selecionado (Liu; Yu, 2005).

2.2 Algoritmo Generalizado

Será apresentado um algoritmo generalizado (adaptado de (Liu; Yu, 2005)) que ilustra os conceitos expostos neste capítulo. Este algoritmo generalizado pode ser implementado utilizando-se de Filtro ou *Wrapper*.

```
Entrada:
      Dados   //os dados dispostos nas n dimensões
      S0      //conjunto inicial de características
      δ       //critério de parada
Saída:
      Sbest   //conjunto ótimo

01    begin
02          Sbest = S0;
03          γbest = avaliar(S0, Dados, M);          //avaliação de S0 por M
04          do begin
05                S = gerarSubConj(Dados);          //gerar subconjunto
06                γ = avaliar(S, Dados, M);         //avaliação de S  por M
07                if (γ melhor que γbest)
08                      γbest = γ;
09                      Sbest = S0;
10          end until (δ é encontrado);
11          return Sbest;
12    end
```

Algoritmo 2.1 – Seleção de Características Generalizado

Para um conjunto de dados *Dados*, o Algoritmo 2.1 começa a busca pelo conjunto dado S_0 (este conjunto pode ser um conjunto vazio, completo ou selecionado aleatoriamente) e segue de acordo com a estratégia particular. A estratégia define o comportamento da função *gerarSubConj*, esta função é responsável pela geração dos novos subconjuntos de características a ser avaliado. Cada subconjunto gerado é avaliado por um método *M*, podendo ser esse qualquer métrica apresentada anteriormente. A busca continua até ser alcançado o critério de parada (δ).

2.3 Métodos de Seleção de Características Abordados

Fazendo a combinações dos quatro fatores (direção, estratégia, avaliação da solução e critério de parada) podemos obter uma grande quantidade de métodos diferentes para a SC. Neste trabalho são utilizados três métodos por busca sequencial, sendo dois por heurística e outro por força bruta.

2.3.1 Busca Sequencial Progressiva (SFS)

A Busca Sequencial Progressiva (*Sequential Forward Seach* – SFS), faz uma busca progressiva, ou para frente, inserindo características em cada iteração no conjunto que, de acordo com o classificador, traz um melhor resultado, ou seja, insere características que mostram o maior ganho na precisão do classificador.

É um método de Subida de Encosta (*Hill Climbing*), assim, explora apenas o melhor caminho local, sem retrocesso. Pode sofrer o efeito *nesting*, ou seja, a solução fica presa ao máximo local descartando o máximo global, comprometendo a qualidade da solução. Por ser um método heurístico tem pouco consumo computacional.

O método parte de um conjunto inicial vazio e a cada iteração insere uma característica ao conjunto. A característica inserida é escolhida pela medida de precisão, ou seja, da avaliação da precisão encontrada pelo classificador dos possíveis conjuntos.

Na Figura 2.2 pode-se observar o espaço de busca percorrido pela SFS para base de dados Iris, que contém quatro características, utilizando o Classificador *Naive Bayes* como medida de precisão[1].

O método parte do conjunto vazio e então são gerados todos os conjuntos contendo apenas uma característica e é calculada a taxa de acuidade de cada conjunto. A que obtiver a maior taxa é selecionado. Na segunda iteração, são gerados conjuntos de características contendo duas características, sendo uma delas a previamente selecionada. Calcula-se a taxa de acuidade e novamente é selecionado o conjunto com a maior taxa de acuidade.

De forma análoga, são gerados os conjuntos sucessivos. O processo termina até o critério de parada ou o conjunto completo seja alcançado. No exemplo da Figura 2.2 este critério é obtido no conjunto contendo a característica x4, pois os conjuntos subsequentes apresentam resultados inferiores aos já encontrados.

[1] A base de dados Iris e o Classificador *Naive Bayes* serão utilizados em todos os exemplos desse capítulo.

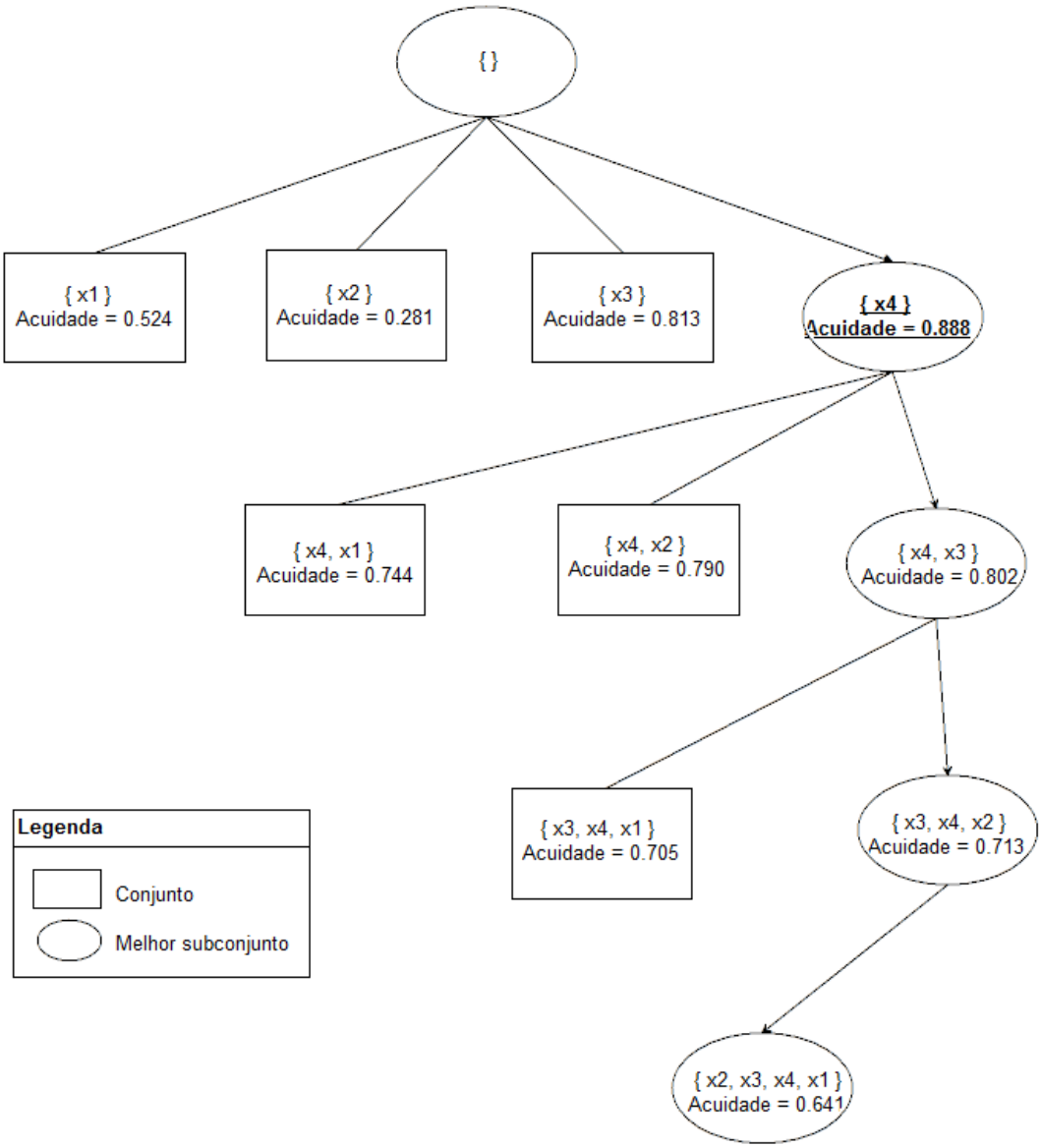


Figura 2.2 – Espaço de Busca SFS - Iris

2.3.2 Busca Sequencial Regressiva (SBS)

Na Busca Sequencial Regressiva (*Sequential Backward Seach* – SBS), é feita a busca regressiva, retirando-se características do conjunto que menos afetam a precisão do classificador. Seu funcionamento é similar ao SFS e também é suscetível ao efeito *nesting* (a solução fica presa ao máximo local descartando o máximo global) por não realizar retrocesso.

Partindo pelo conjunto completo, iterativamente retira-se a característica que, com sua remoção, o conjunto resultante obtém a melhor medida de precisão. Na Figura 2.3 pode-se observar o espaço de busca percorrido pela SBS para base de dados Iris. O método parte do conjunto completo e então são gerados todos os conjuntos possíveis retirando-se apenas uma

característica. Depois de calculada a taxa de acuidade de cada conjunto, o que obtiver a maior taxa é selecionado.

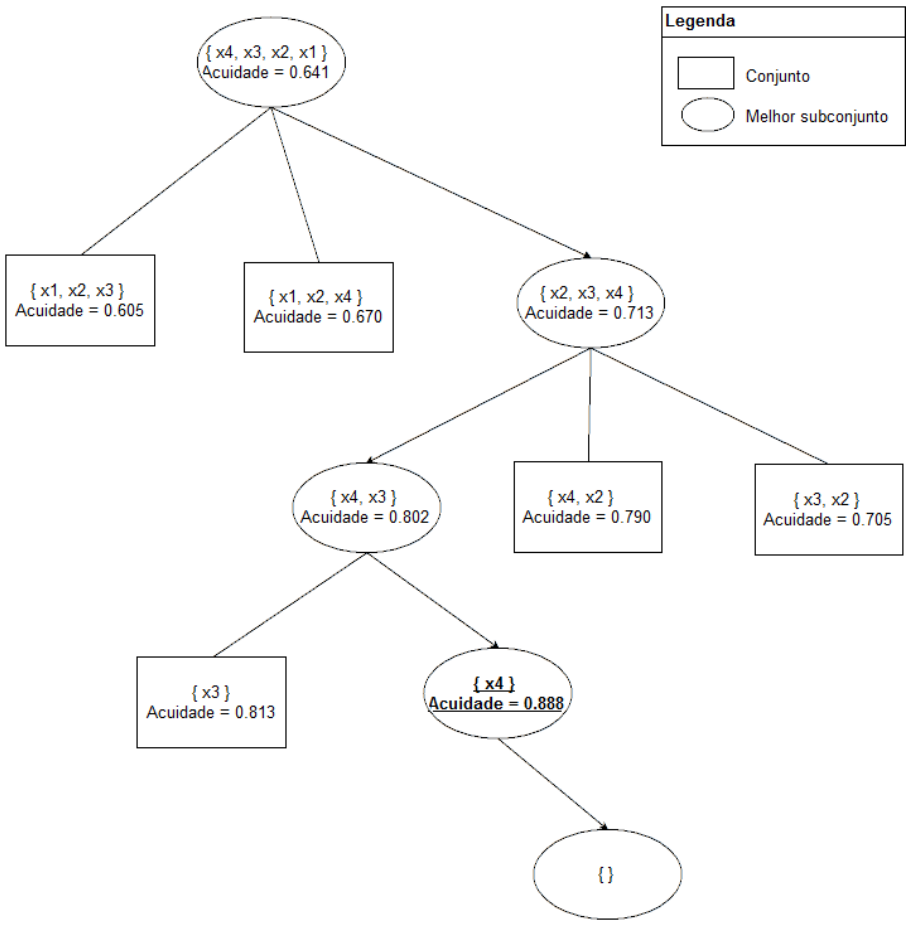


Figura 2.3 – Espaço de Busca SBS - Iris

Na segunda iteração, a partir do conjunto da iteração anterior, são gerados conjuntos de características removendo-se uma característica. Calcula-se a taxa de acuidade e novamente é selecionado o conjunto com a maior taxa de acuidade.

De forma análoga, são gerados os conjuntos sucessivos. O processo termina até o critério de parada ou o conjunto vazio seja alcançado. No exemplo apresentado na Figura 2.3 este critério é obtido no conjunto contendo a característica x4, pois é o conjunto que apresenta o melhor resultado.

2.3.3 Busca em Amplitude (BRD)

A Busca em Amplitude (*Breadth-first Search* – BRD), depois de definido o número alvo de características, varre o espaço de busca, fazendo combinações entre as possíveis

características, buscando o melhor resultado, ou seja, utiliza o classificador nas possíveis combinações e o conjunto que obtiver o melhor resultado é selecionado.

É um método exaustivo que varre o espaço de busca camada por camada, iniciando pelo conjunto vazio. Após, varre a camada que contém apenas uma característica, então a camada que contém duas características e assim sucessivamente até o conjunto completo. Nos métodos propostos pode-se definir o número alvo de características, e assim, limitar o número de iterações.

Na Figura 2.4 pode-se observar a espaço de busca percorrido pela BRD para base de dados Iris. A computação tem início com o conjunto vazio. São gerados todos os conjuntos possíveis contendo apenas uma característica. Calcula-se a taxa de acuidade de cada conjunto, a que obtiver a maior é selecionada.

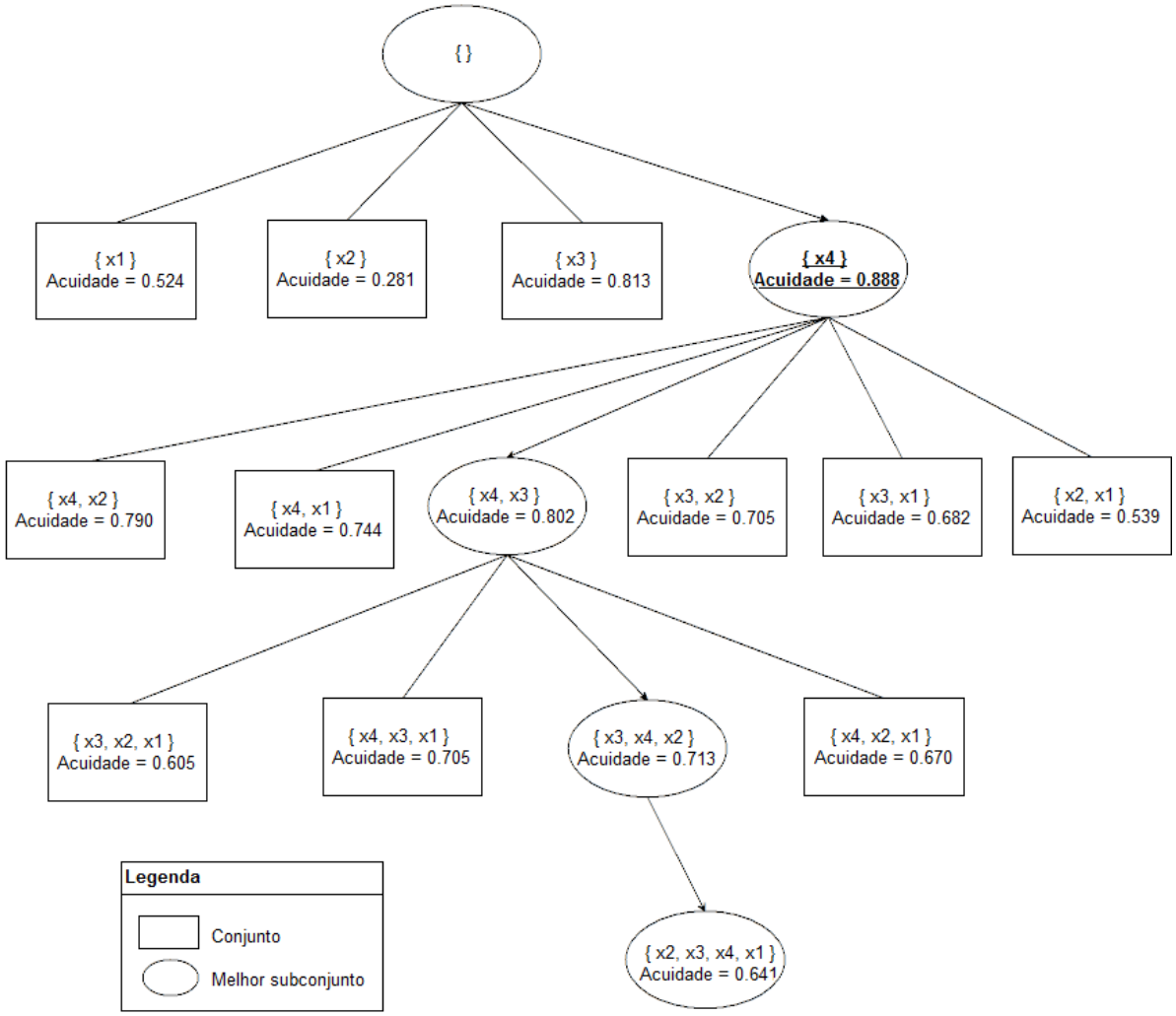


Figura 2.4 – Espaço de Busca BRD - Iris

Na segunda iteração são gerados todos os conjuntos possíveis contendo duas características, e novamente o conjunto que obtiver a maior taxa de acuidade é selecionado.

De forma análoga as próximas iterações são computadas, até que o critério de parada seja obtido. No exemplo apresentado na Figura 2.4 este critério é obtido no conjunto contendo a característica x4, pois é o conjunto que apresenta a maior taxa de acuidade.

2.2 Extração de Características

De acordo com (De Campos, 2001), a Extração de Características (EC) é o processo que cria novas características a partir de transformações ou combinações do conjunto de características inicial, ou seja, criam-se novas características que sejam mais expressivas e que melhor representam a variabilidade dos dados.

O funcionamento básico da EC é a escolha de uma transformação nos dados tal que haja uma alta condensação de informações em poucas características e que a redundância dos dados seja removida. Um processo similar ocorre na percepção humana. Nossa percepção do mundo é baseada em um número relativamente pequeno de relevantes características, que são geradas após processar uma grande quantidade de dados sensoriais, tais como a intensidade e cor dos pixels de imagens capturadas pelos olhos, e os sinais sonoros captados pelos ouvidos (Koutroumbas; Theodoridis, 2008).

As técnicas de EC são largamente utilizadas no reconhecimento de padrões, pois são capazes de extrair as características mais significativas nos dados, principalmente em aplicações onde a dimensionalidade dos dados é um fator limitante. Uma de suas principais aplicações está no reconhecimento de padrões em imagens.

A geração de novas características pode ser feita por meio de transformações lineares, ou seja, sem modificar a estrutura espacial dos dados, conservando assim as relações entre as observações. A Análise de Componentes Principais (PCA) é considerada a transformação linear ótima e tem como ideia a redução de dimensionalidade dos dados, gerando novas variáveis que são funções lineares das variáveis originais maximizando a proporção da variância do conjunto de dados expressa por sucessivas componentes principais que não são correlacionados entre si (Jolliffe, 2002).

2.2.1 Análise de Componentes Principais (PCA)

A Análise de Componentes Principais (PCA - do inglês *Principal Component Analysis*), também chamada de Transformada de Karhunen-Loève, ou ainda, Transformada de Hotelling, descrita pela primeira vez por Karl Person em 1901 (apud Fuzaro; Guliato, 2010), é o método mais popular para redução de dimensionalidade.

O método consiste, basicamente, em buscar o conjunto de eixos ortogonais que maximizem a variância dos dados, ou seja, o arranjo que melhor representa a distribuição dos dados. Esses novos eixos são chamados de Componentes Principais (Koutroumbas; Theodoridis, 2008).

Na Figura 2.5, pode-se observar a representação de um conjunto bidimensional em termos de seus dois atributos (Atributo 1 e Atributo 2). Ainda na Figura 2.5, observa-se a representação das Componentes Principais desse conjunto (PC 1 e PC 2), onde a maior variabilidade dos dados esta na direção da PC 1 e a variabilidade residual esta na direção da PC 2. Assim, fazendo a projeção dos dados nesses novos eixos (PC 1 e PC 2) obtém-se um novo conjunto de dados que mantem o mesmo comportamento espacial, mas com a maior parte da informação concentrada em uma componente (Koutroumbas; Theodoridis, 2008).

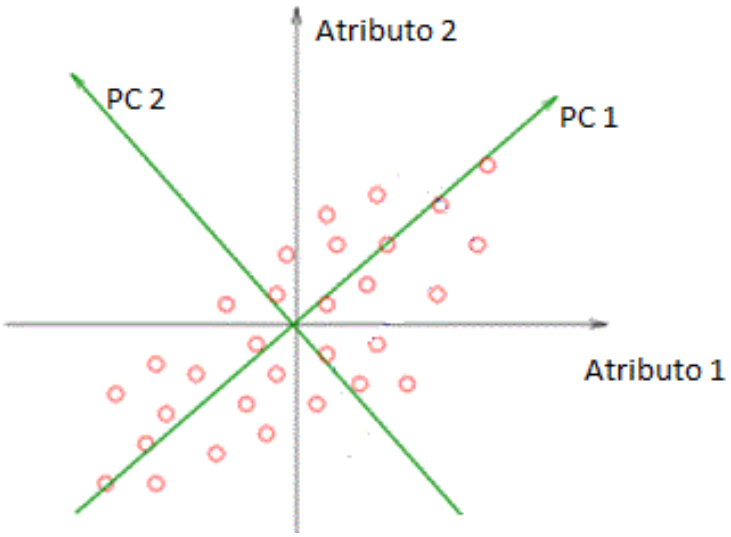


Figura 2.5 - Representação Dimensional PCA

Expandindo, de maneira análoga, para dimensões maiores, podem-se concentrar as informações contidas nos dados em poucas Componentes Principais.

Para isso, o método faz uso da matriz de covariância, que é o resultado da média do produto de cada subtração por ela mesma. Sendo que a mesma forma uma matriz quadrada (n x n), real e simétrica sendo sempre possível encontrar um conjunto de autovalores e ortonormais (Anton; Rorres, 2004).

Um autovetor representa uma direção que é preservada por uma transformação linear. Sendo $\boldsymbol{v}$ um vetor, $\boldsymbol{M}$ uma matriz quadrada e $\boldsymbol{\lambda}$ um escalar segue a propriedade apresentada na Equação 2.1.

$$\boldsymbol{M}.\boldsymbol{v} = \boldsymbol{\lambda}.\boldsymbol{v} \tag{2.1}$$

A multiplicação da matriz $\boldsymbol{M}$ pelo vetor $\boldsymbol{v}$ resulta num múltiplo de $\boldsymbol{v}$.

Nesse caso, $\boldsymbol{\lambda}$ é um autovalor. E, $\boldsymbol{v}$ é chamado de autovetor associado ao autovalor.

Os auvetores são perpendiculares (ortogonais) entre si, sendo possível projetar os dados em termos dos autovetores. O autovetor com o maior autovalor associado representa a direção da maior variabilidade dos dados.

Projetando os dados a partir dos autovetores é gerado um conjunto de dados descorrelacionado. A porção da variabilidade que cada autovetor representa pode ser calculada pelo autovetor ao qual é associado, através da Equação 2.2.

$$\frac{\lambda_i}{\sum_{k=1}^{n} \lambda_k} \tag{2.2}$$

onde λ_i é o autovalor associado ao autovetor v_i e $\sum_{k=1}^{n} \lambda_k$ é o somatório de todos os auvalores. Utilizando essa equação pode-se definir o número de características a qual a base de dados será reduzida.

Definido o número de componentes é gerada uma matriz $\boldsymbol{A}$**,** que contém os autovetores selecionados, organizados de forma decrescente. Os novos dados são gerados através de sua multiplicação pela transposta de $\boldsymbol{A}$.

Para encontrar os autovalores e autovetores pode ser utilizado a Decomposição em Valores Singulares. A decomposição em valores singulares ou *Singular Value Decomposition* (SVD) é a fatoração de uma matriz real ou complexa, com diversas aplicações importantes em processamento de sinais e estatística.

A decomposição em valores singulares de uma matriz $m \times n$ real ou complexa M é uma fatoração na forma apresentada na Equação 2.3.

$$M = U.\sum.V^T \tag{2.3}$$

onde U é uma matriz unitária $m \times m$ real ou complexa, Σ é uma matriz retangular diagonal $m \times n$ com números reais não-negativos na diagonal, e V^T (a conjugada transposta de V) é uma matriz unitária $n \times n$ real ou complexa. As entradas diagonais $\Sigma_{i,i}$ de Σ são os chamados valores singulares de M. As m colunas de U e as n colunas de V são os chamados vetores singulares à esquerda e vetores singulares à direita de M, respectivamente (Koutroumbas; Theodoridis, 2008).

A decomposição em valores singulares e a decomposição em autovalores são intimamente relacionadas. Os vetores singulares à esquerda de M são autovetores de MM^T. Os vetores singulares à direita de M são autovetores de M^TM. E, os valores singulares não nulos de M (ao longo da diagonal de Σ) são as raízes dos autovalores não nulos de MM^T ou M^TM.

A decomposição em valores singulares pode ser usada para calcular a pseudoinversa (*Moore–Penrose pseudoinverse*) de uma matriz, a qual é útil como uma forma de resolver problemas de mínimos quadrados lineares (Koutroumbas; Theodoridis, 2008).

Outra aplicação da SVD é que a mesma representa explicitamente a imagem e o núcleo de uma matriz M. Os vetores singulares à direita que correspondem a valores singulares nulos de M geram o núcleo (*kernel*) de M. Os valores singulares à esquerda correspondendo aos valores singulares não nulos de M geram a imagem de M. Assim, o posto de M é igual ao número de valores singulares não nulos que é igual ao número de elementos não-diagonais (Koutroumbas; Theodoridis, 2008).

Também é amplamente utilizada em estatística, onde se relaciona com a Análise de Componentes Principais, pois é capaz de encontrar os autovalores e autovetores da matriz de covariância. Outro aspecto interessante para sua aplicação na PCA é que os autovetores encontrados já saem ordenados pelo autovalor associado.

Capítulo 3

Materiais e Métodos

Os métodos aqui apresentados (Extração e Seleção de Características) necessitam de um parâmetro de entrada que indique o número de características presentes no conjunto alvo. Propõe-se a comparação dos resultados obtidos pelas duas classes de métodos para cada dimensão alvo. Desta forma, pode-se identificar a diferença entre os conjuntos de características obtidos pelos métodos propostos, bem como o seu comportamento nas diferentes dimensões alvos. Com este objetivo, estes foram implementados na ferramenta *Yet Another Data Mining Tool* (YADMT).

3.1 Yet Another Data Mining Tool

A YADMT, conforme descrição apresentada em (Benfatti *et al.*, 2011), é uma ferramenta de *Data Mining* que vem sendo desenvolvida na UNIOESTE (Universidade Estadual Oeste do Paraná) de forma modular, possibilitando assim que cada técnica pertencente às etapas do processo seja desenvolvida de forma independente e possam ser agrupadas na ferramenta principal como se fossem peças, pacotes, ou objetos autocontidos.

A solução encontrada para atender tal requisito foi utilizar a linguagem de programação Java, que permite o carregamento de módulos. Mas a linguagem por si só não é suficiente para deixar os módulos coerentes com o programa principal, por isso, também foram definidas regras/especificações/restrições para a construção de cada módulo e também se utilizou de meta informações adicionais ao código-fonte, a fim de fornecer informações que possam ser recuperadas em tempo de execução.

A elaboração da ferramenta é realizada em mais de uma etapa. Em um primeiro momento foi desenvolvida a base do programa e definidas as regras/especificações/restrições que os subsistemas devem possuir e seguir para fazerem parte do programa principal.

Na primeira etapa, foram implementadas técnicas que possibilitam a coleta de dados, os quais são necessárias às demais etapas de KDD. A ferramenta faz a coleta de dados do SGBD PostgreSQL utilizando os serviços JDBC (*Java Database Connectivity*) e de arquivos de dados do padrão ARFF (*Attribute-Relation File Format*). O PostgreSQL foi escolhido como primeiro SGBD devido ao fato de ser um SGBD livre, que possibilita a utilização do padrão JDBC, sem custo, robusto e de grande uso. A escolha de dados de arquivo no padrão ARFF se deve por ser um formato utilizado na ferramenta (Weka, 2002), uma ferramenta bastante utilizada em mineração de dados.

Os métodos de classificação presentes na ferramenta são k-NN, C4.5 e *Naive Bayes* (Benfatti, 2010). Também conta com mais três RNA para classificação, sendo elas *Multilayer Perceptron* (MLP), *Radial Basis Function* (RBF) e *Learning Vector Quantization* (LVQ) (Bonifácio, 2010). Estes métodos de classificação serão utilizados para a avaliação dos resultados obtidos pelos métodos de redução de dimensionalidade previamente apresentados.

3.2 Implementando Extração de Características

A implementação de PCA foi dividida em seis etapas distintas, conforme proposta de (Smith, 2002), sendo elas: aquisição de dados, centralização na média, cálculo da matriz de covariância, cálculo dos autovetores e autovalores da matriz de covariância, escolha das componentes e transformação dos dados.

Optou-se pelo uso do pacote JAMA (*Java Matrix Package*), disponível em http://math.nist.gov/javanumerics/jama/, que oferece uma série de operações com matrizes, como multiplicações, somas e a Decomposição em Valores Singulares, que encontra os autovetores e autovalores associados a uma matriz, passo fundamental para PCA. Segue a descrição dos passos:

Etapa 1 - Aquisição de dados: Nesta etapa são selecionados os dados que serão aplicados à técnica. A ferramenta YADMT já possui um módulo de aquisição de dados, sendo assim, não foi necessária sua implementação. O módulo de aquisição presente na ferramenta suporta a entrada de dados via arquivo, através do formato ARFF, e diretamente de um Sistema Gerenciador de Banco de Dados (*PostgreSQL*).

Pela interface é possível escolher um arquivo ARFF, selecionar as colunas desejadas e carregá-lo para ferramenta (Figura 3.1).

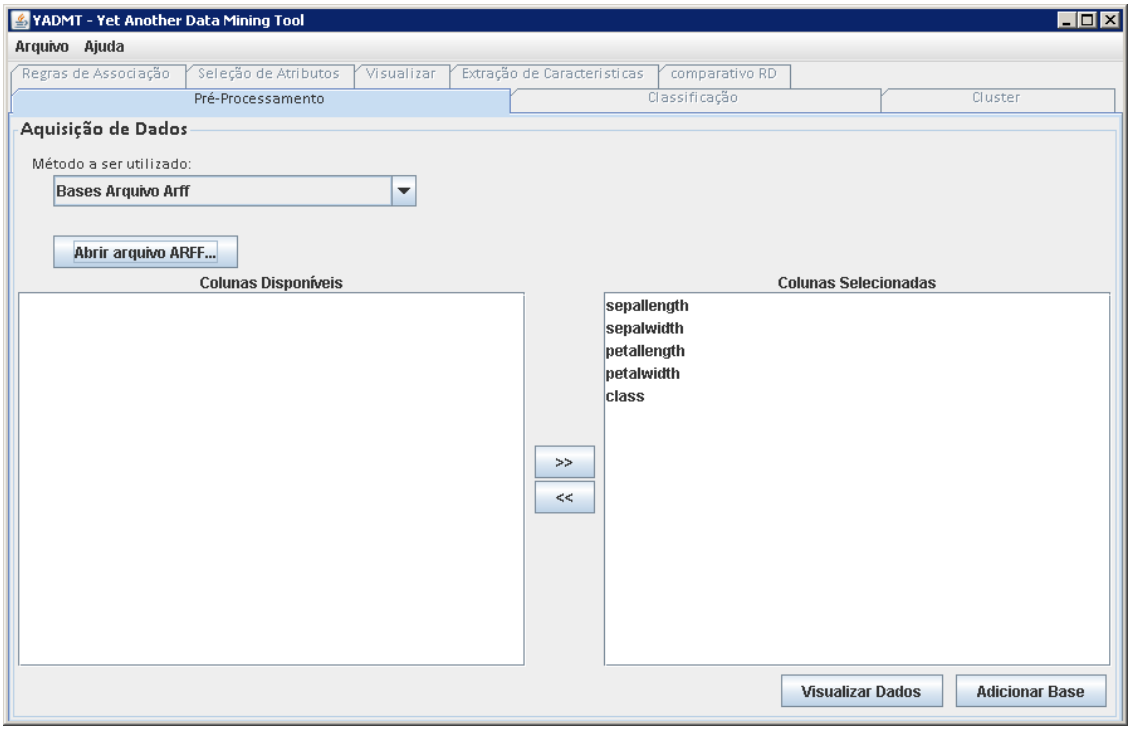


Figura 3.1 – Exemplo de Aquisição ARFF

Através de uma conexão via *socket* com o SGBD PostgreSQL é possível selecionar a base de dados, as colunas de interesse e carrega-la , como pode ser visto na Figura 3.2.

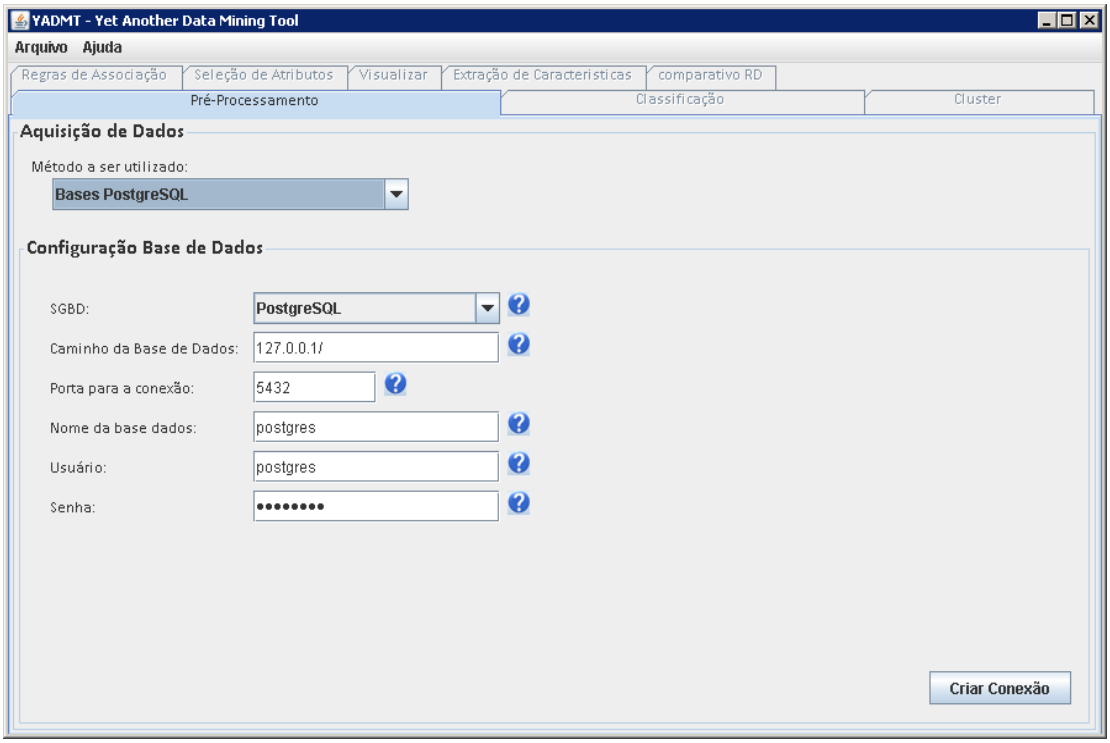


Figura 3.2 – Exemplo de Aquisição PostgreSQL

Etapa 2 - Centralização na média: Para que o PCA funcione corretamente, o valor da média em cada atributo deve ser igual à zero. É calculado o valor médio de cada atributo, então esse valor é subtraído em cada amostra, fazendo com que a média resultante, por atributo, seja igual a zero.

Etapa 3 – Cálculo da matriz de covariância: A matriz de covariância é o resultado da média do produto de cada subtração por ela mesma e terá dimensão $n\ x\ n$. Onde n é o número de atributos da base de dados.

A covariância sempre é medida entre duas dimensões, calcular a covariância entre uma dimensão e ela mesma resulta na sua variância. O cálculo da covariância entre duas dimensões (X e Y) se da pela Equação 3.1.

$$cov(X,Y) = \frac{\sum_{i=1}^{n}[(X_i - \bar{X}).(Y_i - \bar{Y})]}{n} \tag{3.1}$$

onde X e Y são os vetores de dados, $\bar{X}$ e $\bar{Y}$ são as médias das variáveis e n é o numero de itens (usa-se $n - 1$ quando utiliza-se amostra).

Se os dados tiverem mais de duas dimensões, é necessário obter a covariância entre todos os pares de dimensões. A partir dessa ideia surge a matriz de covariância.

A diagonal principal da matriz contém as variâncias e as demais posições a correlação entre as direções. Essa matriz é simétrica e real, de modo que é sempre possível encontrar um conjunto de autovetores ortonormais (Anton; Rorres, 2004).

Etapa 4 - Cálculo dos autovetores e autovalores da matriz de covariância: Diz-se que um autovetor é um vetor v que multiplicado por uma matriz quadrada M resulta em um múltiplo de v (Equação 3.2).

$$M.v = \lambda.v \tag{3.2}$$

onde M é uma matriz quadrada, v um vetor e λ um escalar.

O autovalor é o valor associado ao autovetor, ou seja, ele é o λ. Para a realização deste cálculo é utilizada a biblioteca JAMA, que traz procedimentos para o calculo do mesmo através da Decomposição de Valores Singulares.

Etapa 5 - Escolha das componentes: O autovetor com o maior autovalor associado corresponde à componente principal do conjunto de dados usado. Isso significa que essa é o relacionamento mais significativo entre as dimensões dos dados. Assim, os autovetores são ordenados de acordo com o autovalor associado, fazendo com que, fiquem ordenados do mais significante ao menos significante.

Com base em cada autovetor pode-se explicar certa quantidade da variabilidade dos dados, e com os autovetores ordenados fica fácil à identificação do número de componentes principais necessárias para explicar a maior variabilidade dos dados. Podendo assim reduzir o número de atributos com pouca perda de informações.

Na ferramenta, são apresentadas as componentes e então o usuário seleciona o número de componentes principais, como apresentado na Figura 3.3.

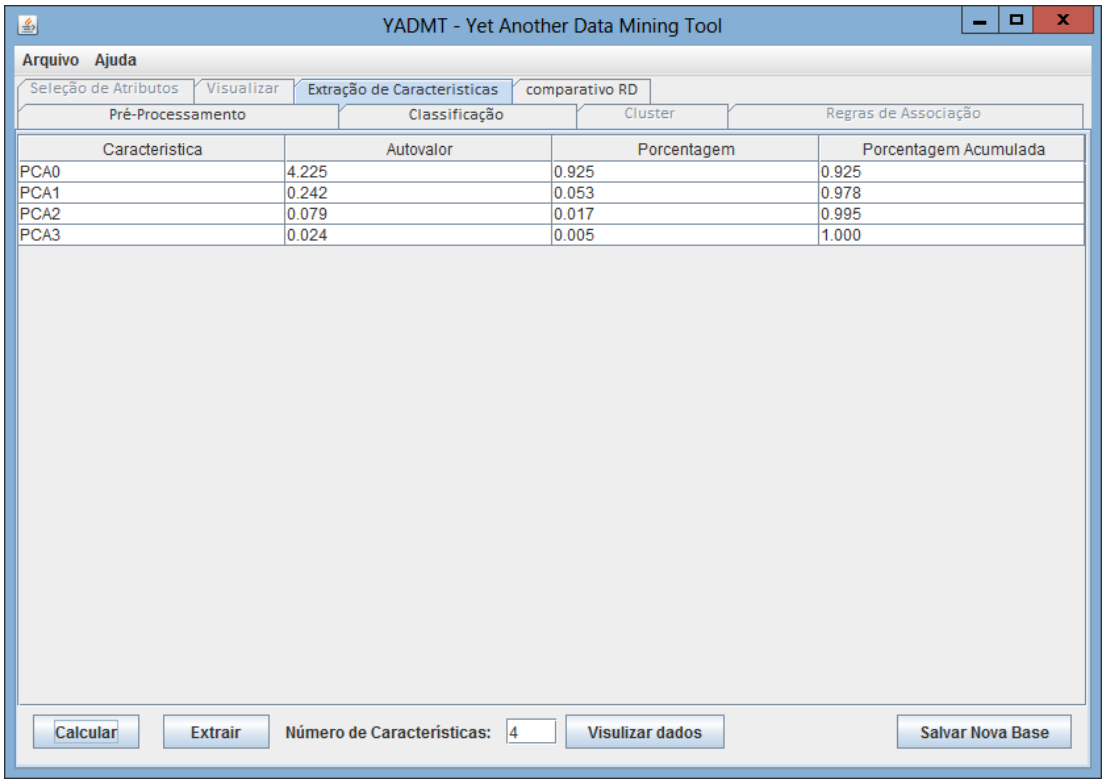
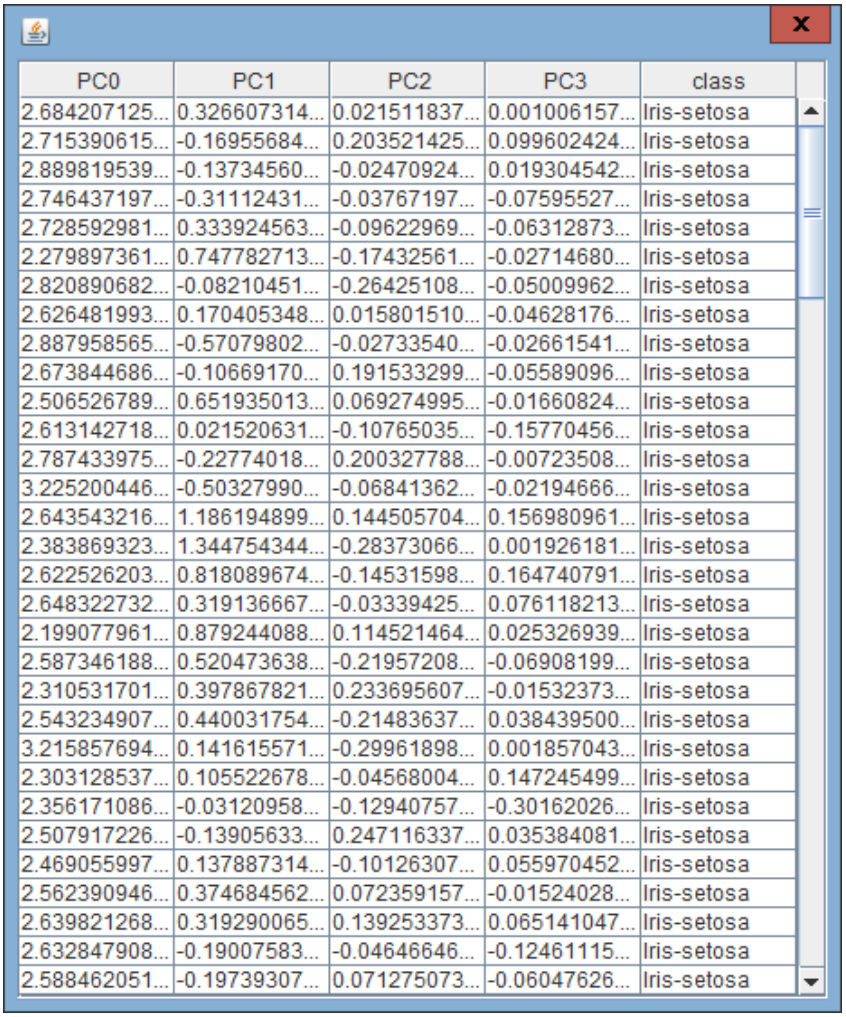


Figura 3.3 – Exemplo de Extração PCA

Etapa 6 - Transformação dos dados: A transformação dos dados se dá pela multiplicação dos autovetores pela matriz de dados. Onde a multiplicação do autovetor mais expressivo resultará na primeira componente principal e assim sucessivamente.

Na ferramenta, os valores estão armazenados em memória, esses valores são multiplicados pelos autovetores e adicionados a uma nova base, a qual será utilizada. A nova base de dados pode ser visualizada através da interface apresentada na Figura 3.4.



PC0	PC1	PC2	PC3	class
2.684207125...	0.326607314...	0.021511837...	0.001006157...	Iris-setosa
2.715390615...	-0.16955684...	0.203521425...	0.099602424...	Iris-setosa
2.889819539...	-0.13734560...	-0.02470924...	0.019304542...	Iris-setosa
2.746437197...	-0.31112431...	-0.03767197...	-0.07595527...	Iris-setosa
2.728592981...	0.333924563...	-0.09622969...	-0.06312873...	Iris-setosa
2.279897361...	0.747782713...	-0.17432561...	-0.02714680...	Iris-setosa
2.820890682...	-0.08210451...	-0.26425108...	-0.05009962...	Iris-setosa
2.626481993...	0.170405348...	0.015801510...	-0.04628176...	Iris-setosa
2.887958565...	-0.57079802...	-0.02733540...	-0.02661541...	Iris-setosa
2.673844686...	-0.10669170...	0.191533299...	-0.05589096...	Iris-setosa
2.506526789...	0.651935013...	0.069274995...	-0.01660824...	Iris-setosa
2.613142718...	0.021520631...	-0.10765035...	-0.15770456...	Iris-setosa
2.787433975...	-0.22774018...	0.200327788...	-0.00723508...	Iris-setosa
3.225200446...	-0.50327990...	-0.06841362...	-0.02194666...	Iris-setosa
2.643543216...	1.186194899...	0.144505704...	0.156980961...	Iris-setosa
2.383869323...	1.344754344...	-0.28373066...	0.001926181...	Iris-setosa
2.622526203...	0.818089674...	-0.14531598...	0.164740791...	Iris-setosa
2.648322732...	0.319136667...	-0.03339425...	0.076118213...	Iris-setosa
2.199077961...	0.879244088...	0.114521464...	0.025326939...	Iris-setosa
2.587346188...	0.520473638...	-0.21957208...	-0.06908199...	Iris-setosa
2.310531701...	0.397867821...	0.233695607...	-0.01532373...	Iris-setosa
2.543234907...	0.440031754...	-0.21483637...	0.038439500...	Iris-setosa
3.215857694...	0.141615571...	-0.29961898...	0.001857043...	Iris-setosa
2.303128537...	0.105522678...	-0.04568004...	0.147245499...	Iris-setosa
2.356171086...	-0.03120958...	-0.12940757...	-0.30162026...	Iris-setosa
2.507917226...	-0.13905633...	0.247116337...	0.035384081...	Iris-setosa
2.469055997...	0.137887314...	-0.10126307...	0.055970452...	Iris-setosa
2.562390946...	0.374684562...	0.072359157...	-0.01524028...	Iris-setosa
2.639821268...	0.319290065...	0.139253373...	0.065141047...	Iris-setosa
2.632847908...	-0.19007583...	-0.04646646...	-0.12461115...	Iris-setosa
2.588462051...	-0.19739307...	0.071275073...	-0.06047626...	Iris-setosa

Figura 3.4 – Apresentação dos dados

O módulo de extração também permite ao usuário salvar a nova base de dados gerada pela técnica em arquivo seguindo o padrão ARFF.

3.3 Implementação Seleção de Características

Para a seleção de características foi proposto a abordagem de três métodos, dois utilizando a busca progressiva e um utilizando a busca regressiva. Todos os métodos utilizam de medidas de precisão, caracterizando assim o modelo *Wrapper*, ou seja, utilizam-se do próprio classificador para a escolha do conjunto de características. Nos dois primeiros são utilizadas heurísticas, fazendo com que nem todo o espaço de busca seja varrido. Já o terceiro método, utiliza-se de força bruta, onde todo o espaço de busca é varrido. É interessante a utilização deste método por garantir o melhor conjunto de características, mas há uma limitação pelo excesso de memória utilizada.

3.3.1 Busca Sequencial Progressiva (SFS)

O método foi implementado na YADMT, utilizado os módulos de aquisição de dados e de classificação. Este último, utilizado para calcular a taxa de acuidade do conjunto.

A ferramenta trabalha com a ideia de bases de dados em forma matricial. Assim, os conjuntos de características devem ser convertidos para forma matricial.

Tem início criando uma base de dados vazia, após são geradas as possíveis bases de dados contendo uma característica. Utilizando o módulo de classificação, são calculadas as respectivas taxas de acuidade. A base de dados que obtiver a maior taxa de acuidade é mantida e as demais são descartadas.

A partir da base de dados preservada, novas são geradas com a combinação da base de dados e as características ainda não selecionadas. Novamente, utilizando o módulo de classificação são calculadas as taxas de acuidade, mantendo a que obteve a maior taxa e descartando as demais. Posteriormente, são geradas bases de dados adicionando uma característica, e o processo segue de forma análoga. A cada iteração é exibida, em terminal, o número de características, a taxa de acuidade e o tempo necessário para encontra-la.

3.3.2 Busca Sequencial Regressiva (SBS)

Como o método anterior, o SBS foi implementado na YADMT e utiliza-se da ideia de bases de dados na forma matricial.

Sua execução tem início com o conjunto completo, sendo esta a base de dados proveniente do módulo de aquisição de dados da ferramenta. É calculada a taxa de acuidade, utilizando o módulo de classificação, para fins comparativos.

Após, são gerados os conjuntos de bases de dados retirando-se uma característica. São calculadas as taxas de acuidade e a base de dados que obter a maior taxa é mantida. De forma análoga, são geradas outras bases de dados retirando-se uma característica. Para cada base de dados é calculada a taxa de acuidade e mantem-se a base de dados com a maior taxa. O processo segue de mesma maneira até encontrar o conjunto vazio. Como no método anterior, a cada iteração é exibida, por meio do terminal, o número de características, a taxa de acuidade e o tempo necessário para encontra-la.

3.3.3 Busca em Amplitude (BRD)

Como os outros métodos, o BRD foi implementado na YADMT e utiliza-se de bases de dados na forma matricial. Este método foi implementado para agir de forma progressiva, gerando todas as possíveis bases de dados de uma determinada dimensão a cada iteração.

A cada iteração são calculadas as taxas de acuidade das bases de dados e melhor taxa é apresentada, bem como o tempo gasto para encontra-la. O espaço de busca cresce de forma exponencial à medida que aumenta a dimensionalidade, assim, inviabiliza-se a utilização deste método em bases de dados que contenham altas dimensionalidades pela sua alta utilização de memória.

3.4 Interface de Execução

Foi desenvolvida uma interface de execução onde é possível selecionar o método de Redução de Dimensionalidade e suas configurações, como pode ser visto na Figura 3.5.

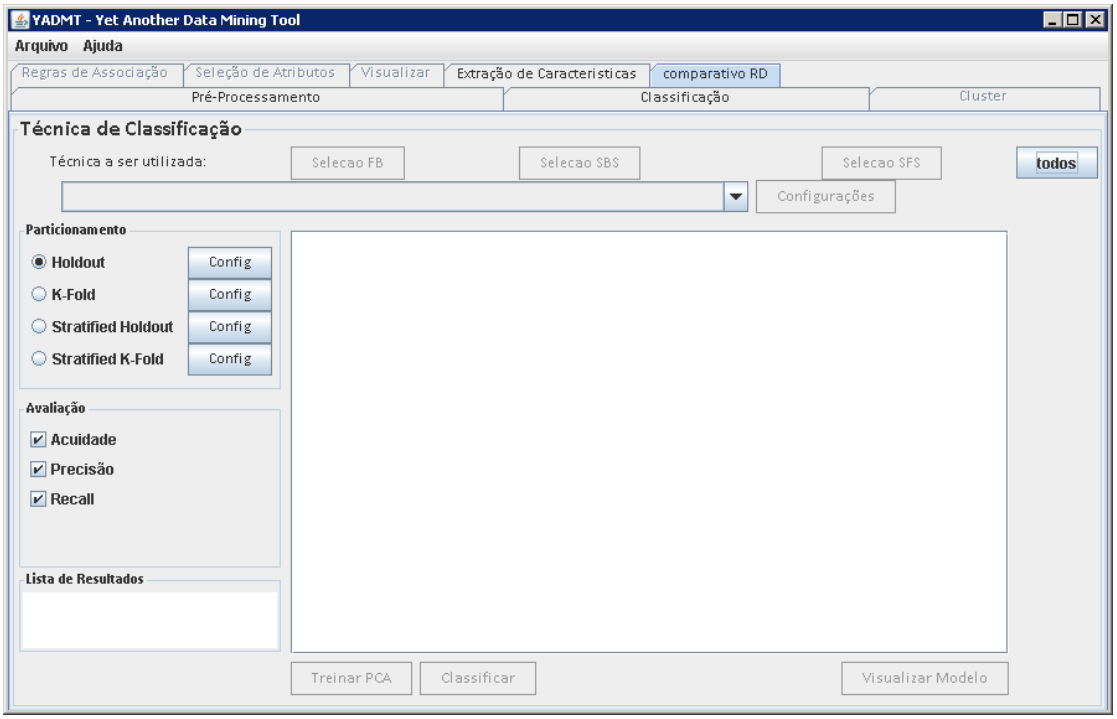


Figura 3.5 – Interface de Execução

Capítulo 4

Avaliação Experimental

Para a realização da avaliação foi utilizada uma máquina Supermicro X8DT3 equipada com dois Processadores Intel Xeon E5620 2.40GHz totalizando 16 *threads*, mas somente 8 dedicados ao Sistema Operacional Windows Server 2008, com 32 GB de memória. O número de núcleos dedicados não interfere diretamente ao ganho de desempenho já que a ferramenta não trabalha de forma paralela, ou seja, todas as operações são realizadas de forma sequencial.

Os métodos propostos nessa sessão foram implementados e acoplados a YADMT seguindo as especificações descritas por (Benfati *et al.*, 2010). Nos métodos de seleção de características (SFS, SBS e BRD) foi utilizado o classificador *Naive Bayes* presentes na ferramenta, utilizando a acurácia (acuidade) como medida de precisão.

4.1 Classificador *Naive Bayes*

Naive Bayes, também conhecido por Classificador Bayesiano, é dito um classificador ingênuo (*naive*) por assumir que os atributos são condicionalmente independentes, ou seja, considera que o efeito do valor de um atribuído sobre uma determinada classe é independente dos valores dos outros atributos, o que simplifica os cálculos envolvidos (Fix; Jr, 1952). Apesar dessa atitude pessimista e simplista, reporta bons desempenhos em inúmeras tarefas, como afirmam (Oguri, 2006) e (Han; Kamber, 2005).

O algoritmo tem como objetivo calcular a probabilidade de que uma amostra desconhecida pertença a cada uma das classes possíveis, ou seja, predizer a classe mais provável. Este tipo de predição é chamado de classificação estatística, pois é completamente baseada em probabilidades.

Este método está inserido em muitas áreas de conhecimento dentre as quais, a título de exemplificação citam-se projetos socioambientais (Zanotta, 2009), supervisão elétrica (Borges; Klautau, 2007), acompanhamento de redes computacionais (Danziger, 2010), entre outros. Este classificador esta presente na ferramenta no módulo de classificação desenvolvido por (Benfati *et al.*, 2010).

4.2 Métricas de Avaliação

O classificador utiliza o particionamento *stratified k-fold-cross-validation* (Witten; Frank, 2005), no qual, as tuplas são divididas em *k* partições (*folds*), com aproximadamente o mesmo número de tuplas. Essas partições são estratificadas, ou seja, as tuplas são selecionadas de forma que as classes estejam proporcionalmente presentes em ambos os conjuntos, garantindo assim a representatividade das classes. Em cada iteração, uma partição é utilizada como conjunto de teste, e as demais utilizadas no processo de treinamento. Esse procedimento é repetido tantas vezes quanto for o número de partições, de modo que dada partição seja utilizada uma vez no conjunto de teste e todas as demais no conjunto de treinamento.

Para a comparação dos resultados é utilizada a taxa de acerto, também conhecida como acurácia (*accuracy*) ou acuidade, que avalia o quão efetivo um algoritmo é, por meio da probabilidade do algoritmo fazer predições corretas (Costa, 2008).

As soluções são submetidas ao classificador e comparadas pela precisão encontrada pelo classificador. Assim o conjunto que obtém uma melhor avaliação é aquele que ter uma maior precisão pelo classificador.

Outra forma de comparação é o tempo necessário para encontrar a solução. É medido o tempo desprendido por cada método para encontrar a solução, e depois comparado entre os métodos, assim o que obtiver o conjunto em um menor tempo obtém uma melhor avaliação. Para a resolução de problemas reais mostram-se mais interessantes estudos em relação ao tempo de execução, já que este pode ser um fator limitante.

Os métodos foram avaliados utilizando-se de diferentes bases de dados. As bases de dados utilizadas estão presentes no repositório UCI (Frank; Asuncion, 2010) que atualmente mantém 211 conjuntos de dados largamente utilizados em soluções de Aprendizado de Máquinas.

A fim de avaliar o desempenho das técnicas de classificação do módulo proposto, foram utilizadas sete bases de dados. O conjunto de bases de dados selecionado possibilitou analisar o comportamento dos algoritmos em bases de dados pequenas, médias e grandes quanto a número de registro, número de atributos e número de classes, além de bases de dados com atributos tanto categóricos como numéricos apresentados na Tabela 4.1.

Tabela 4.1- Bases de Dados Utilizadas

Nome	Número de Atributos	Número de Amostras	Número de Classes
Iris	5	150	3
Ecoli	8	336	8
Acute Inflammations	6	120	2
Blood Transfusion Service Center	5	748	2
Glass Identification	9	214	6
Ionosphere	35	351	2
LIBRAS *Movement*	91	360	15

De acordo com (Zheng, 2006), um classificador tem alta taxa de acuidade quando a porcentagem da taxa de acertos for maior que 75%, média quando estiver entre 40 e 75% e baixa quando for menor que 40%.

Os algoritmos foram treinados utilizando a estratégia *k-fold-cross-validation,* com k = 10 para particionamento dos conjuntos treinamento e teste. A partir destes, foram considerada como medida de avaliação a acuidade. Os valores encontrados foram submetidos a teste para igualdade de proporções com um nível de significância de 5%.

Segue a descrição das bases de dados, bem como a apresentação de resultados obtidos. Nas tabelas os valores em **destaque** são os melhores resultados obtidos e as células contendo * indicam que não foi possível encontrar uma solução.

Soluções não foram possíveis utilizando o método BRD em bases de dados de alta dimensionalidade, pois houve um "estouro" de memória, ou seja, durante a sua execução foi excedido o limite de 20 Gb de memória (limite imposto pela capacidade da máquina utilizada).

4.3 Iris

Contém 150 amostras em três classes, sendo uma delas linearmente separável e as outras duas não. Cada amostra é composta com quatro características, que representam o tamanho e a largura da sépala e da pétala de uma flor.

Na Tabela 4.2, observa-se que o PCA obteve alta taxa acuidade (acima de 75%) para todos os conjuntos de características, o que não ocorre com os métodos de SC. Nos conjuntos com 3 e 4 características os métodos de SC obtiveram uma taxa de acuidade média (entre 40% e 75%). Porém, a taxa de acuidade encontrada pelos métodos de SC é superior ao PCA. Em relação ao tempo para encontrar a solução, o SFS obteve a solução ótima em menor tempo

(31 milissegundos), mas PCA obteve soluções em tempos muito inferiores, de 2 a 9 milissegundos. Os valores encontrados utilizando PCA são estatisticamente iguais utilizando significância de 5%. Nos métodos de SC o valor ótimo difere estatisticamente dos demais.

Tabela 4.2- Avaliação Experimental na base de dados Iris

	BRD		**SFS**		**SBS**		**PCA**	
N	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)
1	**0,888^{A}**	62	**0,888^{A}**	31	**0,888^{A}**	64	**0,811^{aA}**	2
2	0,802^{aBA}	93	0,802^{aA}	47	0,802^{aA}	63	0,794^{aA}	3
3	0,713^{abA}	109	0,713^{abA}	62	0,713^{abA}	32	0,824^{a}	5
4	0,641^{bA}	124	0,641^{bA}	63	0,641^{bA}	16	0,812^{a}	9

Nota: Letras minúsculas representam médias estatisticamente iguais nas colunas, e letras maiúsculas representam médias estatisticamente iguais nas linhas a 5% de significância. Valores em destaque representam os melhores resultados.

No gráfico da Figura 4.1 pode-se observar a precisão obtida pelos métodos propostos em relação ao número de características. Observa-se que quando utilizada uma única característica os métodos de seleção obtém uma precisão maior que o PCA. Já com duas características todos os métodos obtêm valores similares, e a partir que se inserem novas características o PCA obtém uma maior precisão.

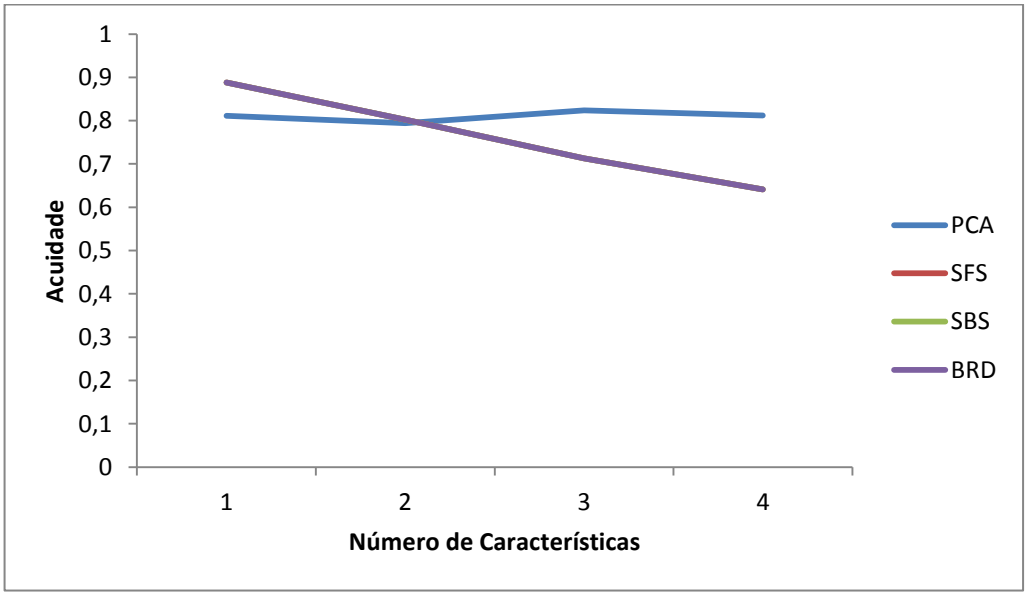


Figura 4.1 - Precisão *versus* o Número de Características IRIS

No gráfico da Figura 4.2 pode-se observar o tempo necessário para encontrar as soluções. O PCA tem custo fixo para encontrar a transformação linear, mas para realizar a transformação dos dados, conforme o aumento da dimensão requer um número maior de operações aumentando assim o tempo necessário para gerar a base de dados. O BRD tem um custo muito superior aos outros métodos. Os métodos SFS e SBS encontram soluções rapidamente quando perto do extremo em que se inicia, mas conforme se insere (ou retira-se) características seu custo aumenta.

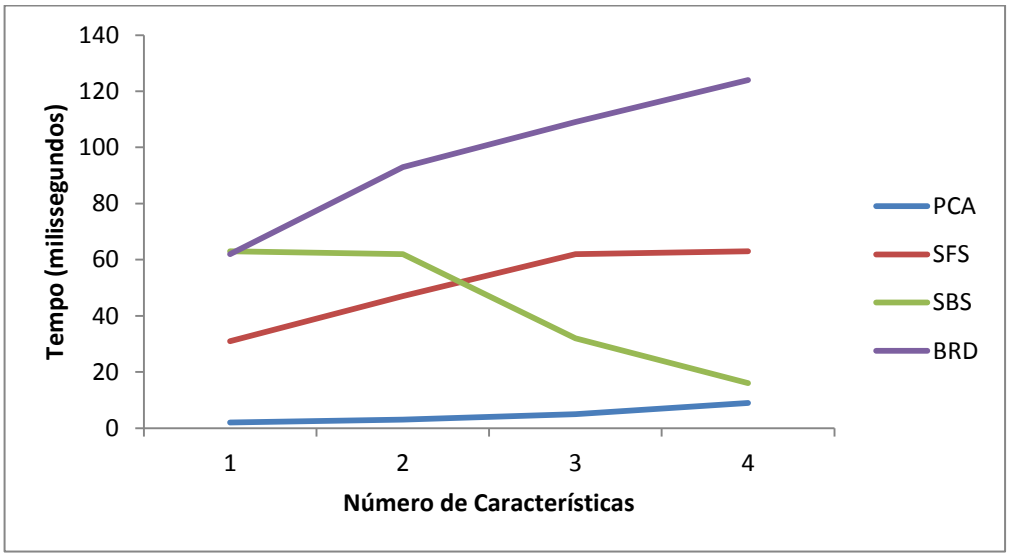


Figura 4.2 - Tempo *versus* o Número de Características IRIS

4.4 Ecoli

A base de dados Ecoli foi criada por Kenta Nakai do Instituto de Biologia Celular e Molecular da Universidade de Osaka em 1996. Contêm dados sobre a localização de proteínas em células. Conta com um atributo categórico (*Sequence Name*, em inglês), sete atributos numéricos e um último atributo que define a classe do objeto, neste caso, a localização da célula que tem a maior concentração de proteína. Nesta base de dados há 336 amostras.

A Tabela 4.3 apresenta os resultados da avaliação na base de dados Ecoli. Nela observa-se que a taxa de acuidade encontrada pelo PCA em todos os conjuntos de características é superior ao dos demais métodos. A partir do conjunto contendo três características, PCA encontra uma taxa de acuidade média (entre 40% e 75%), sendo que nenhum outro método obteve tal resultado. O tempo necessário para encontrar tais soluções é inferior ao dos demais. Os conjuntos utilizando PCA contendo seis, sete e oito características são estatisticamente iguais, mas pelo menor número de características o conjunto contendo seis características

representa o melhor resultado. Os métodos de SC encontram o seu melhor resultado logo na segunda iteração, sendo os seus respectivos resultados na próxima iteração estatisticamente iguais. Porém o método SBS encontra uma solução inferior às demais.

Tabela 4.3 - Avaliação Experimental na base de dados Ecoli

	BRD		**SFS**		**SBS**		**PCA**	
n	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)
1	$0{,}159^{aA}$	172	$0{,}159^{aA}$	47	$0{,}159^{aA}$	795	$0{,}226^{a}$	1
2	**$0{,}219^{bA}$**	593	**$0{,}219^{bA}$**	110	$0{,}219^{bA}$	780	$0{,}289^{a}$	2
3	$0{,}219^{bA}$	1482	$0{,}219^{bA}$	172	$0{,}219^{bA}$	749	$0{,}416^{b}$	3
4	$0{,}217^{bA}$	2746	$0{,}217^{bA}$	266	$0{,}184^{abA}$	671	$0{,}424^{b}$	4
5	$0{,}202^{bcA}$	3947	$0{,}202^{bcA}$	375	$0{,}183^{abA}$	561	0,513	4
6	$0{,}182^{abcA}$	4649	$0{,}182^{abcA}$	453	$0{,}182^{abA}$	421	**$0{,}620^{c}$**	6
7	$0{,}145^{acA}$	4868	$0{,}145^{acA}$	515	**$0{,}145^{aA}$**	249	$0{,}625^{c}$	8
8	$0{,}0193^{A}$	4899	$0{,}019^{A}$	531	$0{,}019^{A}$	31	$0{,}611^{c}$	24

Nota: Letras minúsculas representam médias estatisticamente iguais nas colunas, e letras maiúsculas representam médias estatisticamente iguais nas linhas a 5% de significância. Valores em destaque representam os melhores resultados.

No gráfico da Figura 4.3 pode-se observar a precisão obtida pelos métodos propostos em relação ao número de características. Nesta base de dados os resultados obtidos pela PCA foram superiores aos outros métodos. Pode-se observar o efeito do Problema da Dimensionalidade nos métodos de SC, onde a inserção de novas características faz com que caia a taxa de acuidade. Isso não ocorre com o PCA, pois as características são transformadas a fim de reduzir a redundância e a correlação.

No gráfico da Figura 4.4 pode ser observado o tempo necessário para encontrar as soluções. O tempo para encontrar a solução utilizando PCA é muito inferior ao dos outros métodos. Por percorrer todo o espaço de busca, o BRD tem um tempo de execução muito superior.

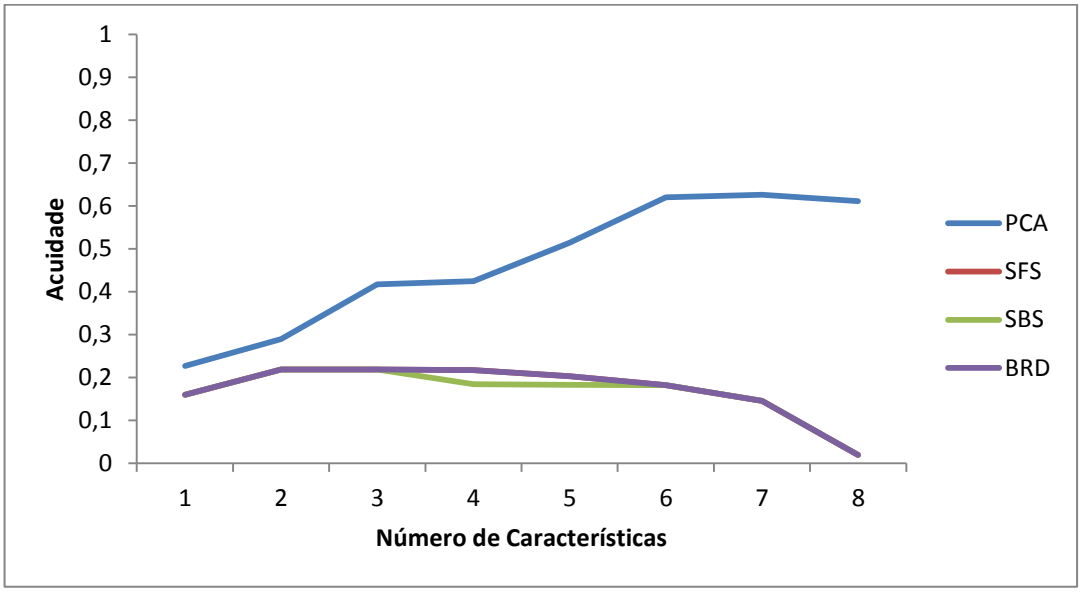


Figura 4.3 - Precisão *versus* o Número de Características ECOLI

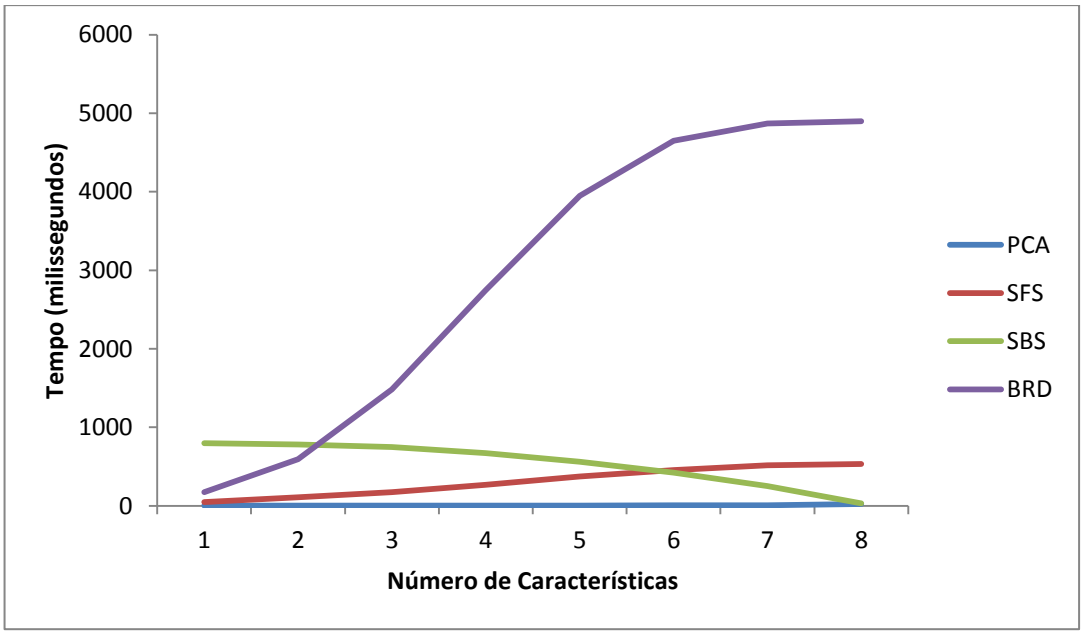


Figura 4.4 - Tempo *versus* o Número de Características ECOLI

4.5 Acute Inflammations

A base de dados *Acute Inflammations* com 120 amostras foi criada por Jacek Czerniak, do Instituto de Pesquisa de Sistemas da Academia Polonesa de Ciências, e está disponível na UCI desde 2009. Apresentam dados criados para testar um sistema especialista, que deveria realizar o diagnóstico presumível de doenças do sistema urinário. Esta base de dados possui 8 atributos, sendo: 1 atributo numérico, indicando a temperatura do paciente; 7 categóricos

(ocorrência de náuseas, dor lombar, necessidade constante de urinar, dores 58 demicção, dores de uretra, ocorrência de inflamação urinária na bexiga e nefrite de origem pelve renal), que podem assumir valores "*yes*"e "*no*", destes; os dois últimos atributos representam atributos de decisão de classe baseado nos seis primeiros atributos.

Na Tabela 4.4, pode se observar que BRD encontrar seu melhor resultado com apenas três características, SFS com apenas uma característica e PCA encontram com quatro características, SBS encontra seu melhor resultado na segunda iteração (com seis características). Apenas SFS não encontra a taxa de acuidade máxima (100%).

Tabela 4.43 - Avaliação Experimental na base de dados Acute Inflammations

	BRD		**SFS**		**SBS**		**PCA**	
n	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)
1	$0{,}817^{aA}$	63	**$0{,}817^{aA}$**	31	$0{,}714^{aA}$	110	$0{,}747^{aA}$	1
2	$0{,}845^{aA}$	172	$0{,}845^{aA}$	46	$0{,}714^{aB}$	110	$0{,}761^{aAB}$	0
3	**1^{b}**	328	$0{,}845^{aA}$	62	$0{,}918^{bA}$	94	$0{,}918^{A}$	1
4	1^{bA}	484	$0{,}845^{a}$	78	1^{cA}	78	**1^{bA}**	1
5	1^{bA}	578	$0{,}845^{a}$	93	1^{cA}	63	1^{bA}	2
6	1^{bA}	624	$0{,}845^{a}$	93	**1^{cA}**	32	1^{bA}	6
7	$0{,}845^{aA}$	624	$0{,}845^{aA}$	109	$0{,}845^{bA}$	6	1^{b}	13

Nota: Letras minúsculas representam médias estatisticamente iguais nas colunas, e letras maiúsculas representam médias estatisticamente iguais nas linhas a 5% de significância. Valores em destaque representam os melhores resultados.

No gráfico da Figura 4.5 pode-se observar a precisão obtida pelos métodos propostos em relação ao número de características. Observa-se que quando utilizada poucas característica os métodos de seleção obtém uma precisão maior que o PCA. A partir que se inserem novas características o PCA obtém uma maior precisão. BRD e SBS alcançam valor máximo nos conjuntos contendo de quatro a seis características, mas ao inserir-se a última característica a precisão decai, o que não ocorre utilizando PCA.

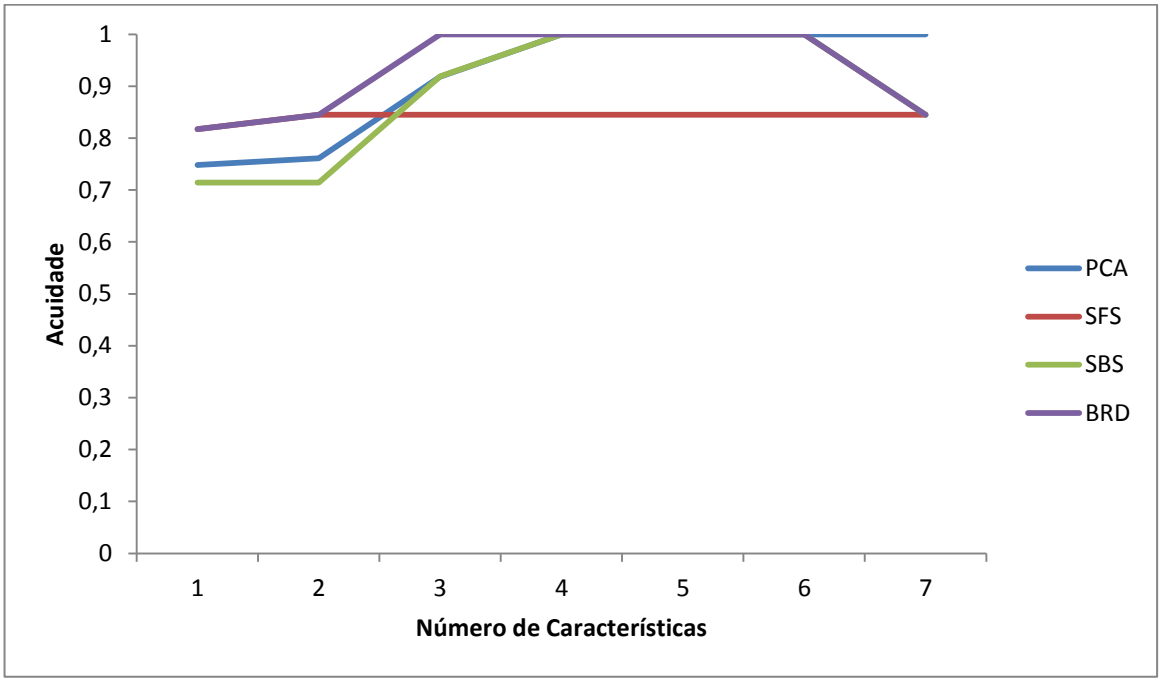


Figura 4.5 - Precisão *versus* o Número de Características ACUTE

No gráfico da Figura 4.6 pode-se observar o tempo necessário para encontrar as soluções. O PCA demonstra encontrar soluções muito mais rapidamente que os outros métodos apresentados. Observa-se que por não utilizar heurística, consequentemente varrendo todo o espaço de busca, o método BRD, consome um tempo muito maior em comparação ao demais.

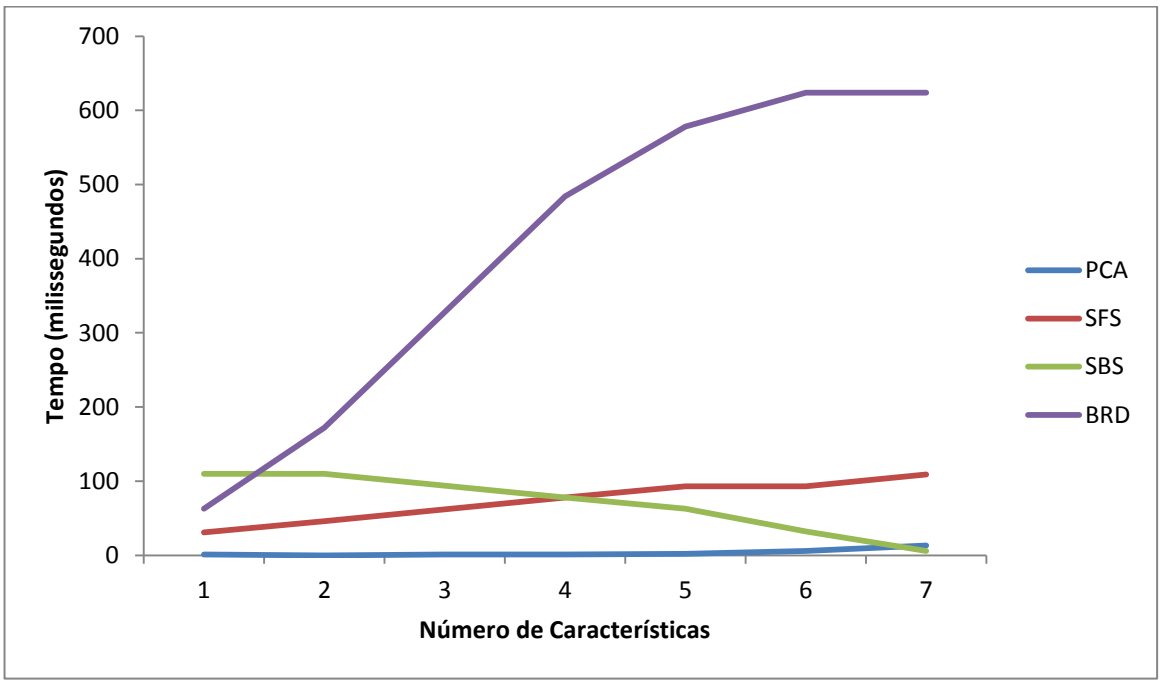


Figura 4.6 - Tempo *versus* o Número de Características ACUTE

4.6 Blood Transfusion Service Center

Base de dados contém 748 amostras divididas em duas classes, problema binário, que indicam se uma pessoa doou ou não sangue em março de 2007. Todas as características são discretas, são elas: *R*, meses desde a última doação; *F*, quantidade de doações; *M*, total de sangue doado no centro clínico e *T*, meses desde a primeira doação.

A Tabela 4.5 apresenta os resultados obtidos na avaliação experimental dos métodos propostos. Nela, observa-se que todos os métodos obtiveram taxa de acuidade média para todas as dimensões. Os resultados encontrados em todas as dimensões pelos métodos mostraram-se estatisticamente iguais utilizando uma significância de 5%.

Tabela 4.5- Avaliação Experimental na base de dados *Blood Transfusion Service Center*

	BRD		**SFS**		**SBS**		**PCA**	
n	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)
1	**0,426^{aA}**	181	**0,426^{aA}**	88	0,426^{aA}	380	**0,432^{aA}**	4
2	0,455^{aA}	398	0,455^{aA}	194	0,455^{aA}	334	0,477^{aA}	5
3	0,448^{aA}	587	0,448^{aA}	282	0,448^{aA}	232	0,481^{aA}	12
4	0,450^{aA}	647	0,450^{aA}	335	**0,450^{aA}**	55	0,481^{aA}	24

Nota: Letras minúsculas representam médias estatisticamente iguais nas colunas, e letras maiúsculas representam médias estatisticamente iguais nas linhas a 5% de significância. Valores em destaque representam os melhores resultados.

No gráfico da Figura 4.7, observa-se a representação das taxas obtidas pelos métodos nas diferentes dimensões. Pode-se observar que PCA obteve um pequeno ganho em relação aos demais.

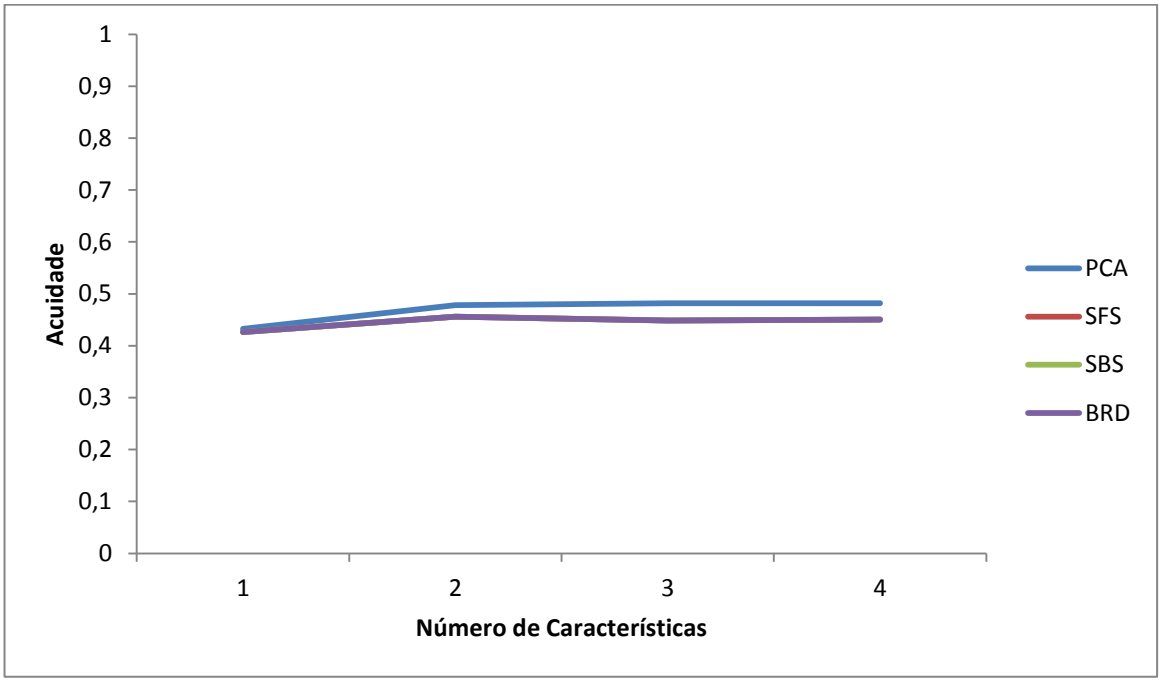


Figura 4.7 - Precisão *versus* o Número de Características *Blood Transfusion Service Center*

No gráfico da Figura 4.8 apresenta o tempo necessário para cada método encontrar as soluções. Pode-se observar a diferença entre os tempos para encontrar os conjuntos solução, mesmo obtendo uma taxa de acuidade estatisticamente igual PCA obteve em um menor tempo.

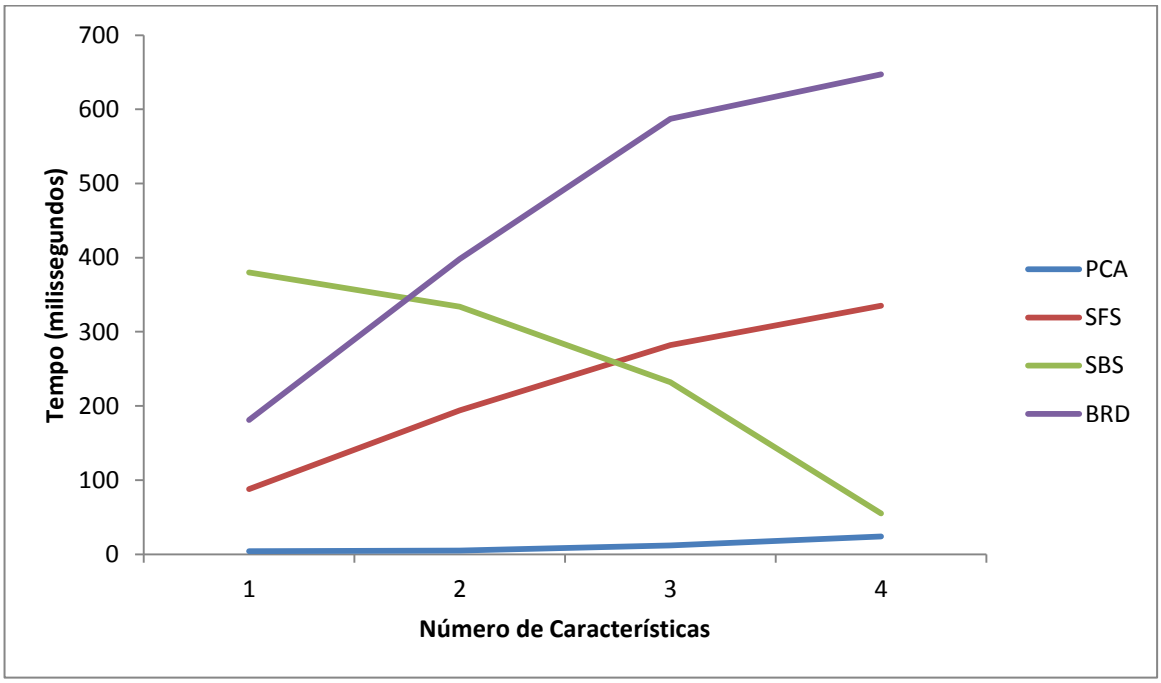


Figura 4.8 - Tempo *versus* o Número de Características *Blood Transfusion Service Center*

4.7 Glass Identification

Contém 214 padrões em 6 classes que correspondem a tipos de vidros, identificados por: 0 (*building windows float processed*), 1 (*building windows non float processed*), 2 (*vehicle windows float processed*), 3 (*containers*), 4 (*tableware*), 5 (*headlamps*), duas classes predominam o que é de interesse. Cada padrão é formado por nove características representando quantidades de constituintes químicos presentes nas amostras mais um índice de refrativo. As características são: *RI* (índice refrativo), *Na* (sódio), *Mg* (magnésio), *Al* (alumínio), *Si* (silício), *K* (potásio), *Ca* (cálcio) e *Ba* (bário), todas as características são contínuas.

Na Tabela 4.6, observa-se que PCA consegue obter uma taxa de acuidade média (acima de 40%), sendo que os demais métodos obtêm taxas baixas (inferiores a 25%). Utilizando PCA, no conjunto contendo quatro características obtém-se uma taxa de acuidade de 33,78% caracterizando o melhor resultado, pois é estatisticamente igual a maior taxa encontrada (42,70%).

Tabela 4.6- Avaliação Experimental na base de dados *Glass Identification*

	BRD		**SFS**		**SBS**		**PCA**	
n	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)
1	**0,169^{aA}**	275	**0,1695^{aA}**	27	0,1695^{aA}	401	0,086454	1
2	0,245^{aA}	620	0,245^{aA}	57	0,245^{aA}	394	0,192318^{aA}	1
3	0,239^{aA}	1279	0,239^{aA}	99	0,239^{aA}	377	0,184414^{aA}	2
4	0,175^{aA}	2288	0,175^{aA}	148	**0,175^{aA}**	351	**0,337846^{b}**	3
5	0,065^{bA}	3403	0,065^{bA}	192	0,065^{bA}	314	0,364085^{b}	3
6	0,047^{bcA}	4216	0,047^{bcA}	232	0,035^{bcA}	263	0,37632^{b}	3
7	0,027^{bcA}	4598	0,027^{bcA}	265	0,027^{bcA}	198	0,386236^{b}	4
8	0,023^{cA}	4701	0,023^{cA}	290	0,023^{cA}	115	0,361183^{b}	5
9	0,018^{cA}	4713	0,018^{cA}	303	0,018^{cA}	12	0,427042^{b}	26

Nota: Letras minúsculas representam médias estatisticamente iguais nas colunas, e letras maiúsculas representam médias estatisticamente iguais nas linhas a 5% de significância. Valores em destaque representam os melhores resultados.

No gráfico da Figura 4.9, observa-se que, a partir da segunda dimensão, a precisão dos métodos SFS, SBS e BRD decai com o acréscimo de características, caracterizando o problema da dimensionalidade. Utilizando PCA isso não ocorre devido ao fato de a correlação e a redundância das características serem minimizadas.

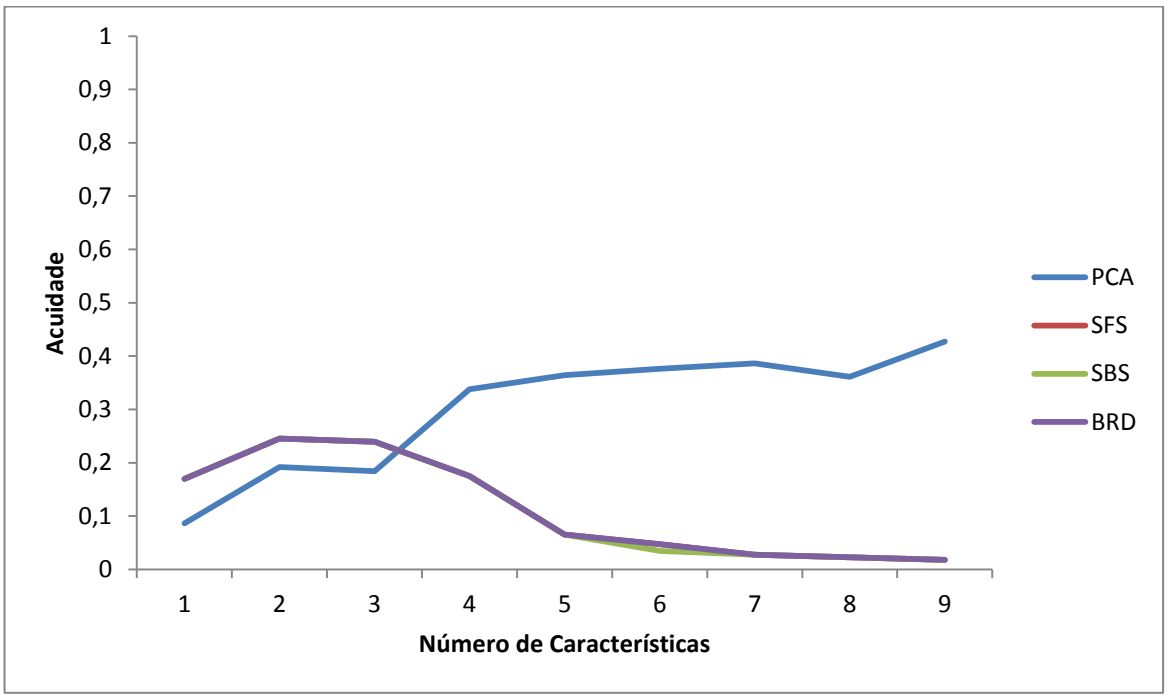


Figura 4.9 - Precisão *versus* o Número de Características *Glass Identification*

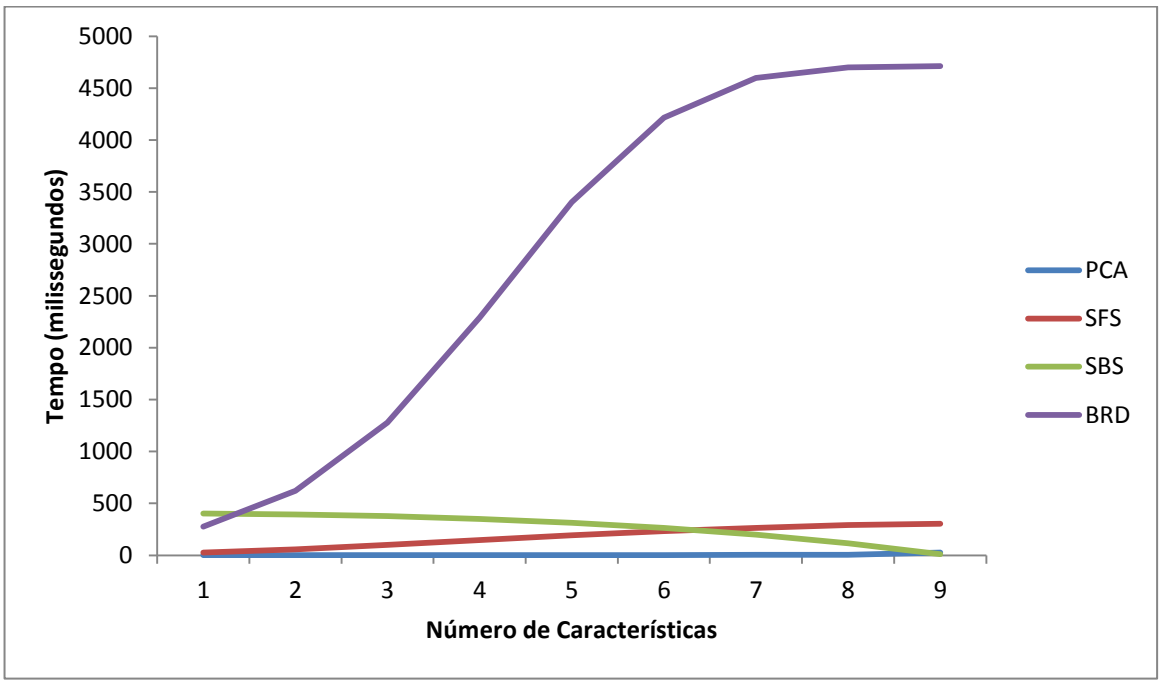


Figura 4.10 - Tempo *versus* o Número de Características *Glass Identification*

No gráfico da Figura 4.10, pode observar-se que tempo gasto pelo BRD é superior aos demais. PCA obtém seu conjunto resposta em um tempo muito inferior aos demais.

4.8 Ionosphere

Contém 351 padrões em duas classes: *b* (*bad*) e g (*good*). Os padrões representam medições recebidas por 16 antenas de alta frequência de um sistema de radar. Cada padrão é formado por 17 pulsos complexos, e existem duas características para cada pulso (coeficientes do número complexo), totalizando 34 características contínuas. Quando o sinal recebido é fraco (*bad*), significa que o sinal passa direto pela ionosfera, quando o sinal retornado é forte (*good*), significa que o sinal é refletido de volta por um objeto qualquer.

Na Tabela 4.7, observa-se que com poucas dimensões os métodos encontram seus conjuntos resposta. A taxa de acuidade encontrada pela PCA é superior em dez pontos percentuais dos resultados dos demais métodos.

Tabela 4.7- Avaliação Experimental na base de dados *Ionosphere*

	BRD		**SFS**		**SBS**		**PCA**	
n	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)	Acuidade	Tempo (ms)
1	**0,735^{aA}**	400	**0,735^{aA}**	140	0,697^{aA}	39577	0,501^{a}	0
2	0,735^{aA}	4891	0,735^{aA}	343	0,708^{aA}	39562	0,500^{a}	16
3	0,735^{aA}	60224	0,735^{aA}	686	0,656^{abB}	39546	0,649^{bB}	0
4	0,716^{aA}	597302	0,716^{abA}	1201	**0,635^{abcB}**	39515	0,636^{bB}	0
5	0,687^{aA}	4402594	0,687^{abcA}	1747	0,614^{bcd}	39453	0,778^{c}	0
6	*	*	0,666^{bcdA}	2355	0,598^{bcdeA}	39375	0,779^{c}	16
7	*	*	0,659^{bcde}	3026	0,587^{bcde}f	39250	**0,836^{cd}**	16
8	*	*	0,652^{bcdef}	3791	0,582^{cdefg}	39109	0,837^{d}	16
9	*	*	0,640^{cdefgA}	4617	0,577^{cdefghA}	38907	0,822^{cd}	0
10	*	*	0,629^{cdefghA}	5507	0,577^{cdefghA}	38641	0,812^{cd}	15
11	*	*	0,619^{cdefghiA}	6474	0,572^{cdefghiA}	38329	0,823^{cd}	0
12	*	*	0,608^{defghijA}	7441	0,566^{cdefghiA}	37939	0,823^{cd}	0
13	*	*	0,598^{defghijkA}	8439	0,566^{cdefghiA}	37471	0,818^{cd}	0

14	*	*	$0,593^{efghijklA}$	9407	$0,566^{cdefghiA}$	36910	$0,818^{cd}$	16
15	*	*	$0,582^{fghijklmA}$	10405	$0,566^{cdefghiA}$	36286	$0,827^{cd}$	0
16	*	*	$0,572^{ghijklmnA}$	11403	$0,566^{cdefghiA}$	35553	$0,833^{cd}$	0
17	*	*	$0,566^{hijklmnoA}$	12386	$0,566^{cdefghiA}$	34648	$0,823^{cd}$	16
18	*	*	$0,551^{ijklmnopA}$	13353	$0,566^{cdefghiA}$	33665	$0,824^{cd}$	0
19	*	*	$0,541^{jklmnopA}$	14289	$0,56^{6cdefghiA}$	32511	$0,809^{cd}$	16
20	*	*	$0,530^{klmnopA}$	15210	$0,561^{defghijA}$	31247	$0,825^{cd}$	15
21	*	*	$0,520^{lmnopA}$	16099	$0,556^{defghijA}$	29827	$0,830^{cd}$	16
22	*	*	$0,515^{mnopA}$	16988	$0,551^{defghijkA}$	28345	$0,825^{cd}$	16
23	*	*	$0,510^{mnopA}$	17815	$0,546^{defghijkA}$	26692	$0,836^{cd}$	15
24	*	*	$0,510^{mnopA}$	18626	$0,541^{defghijkA}$	24976	$0,840^{d}$	16
25	*	*	$0,510^{mnopA}$	19359	$0,535^{efghijkA}$	23104	$0,835^{cd}$	16
26	*	*	$0,510^{mnopA}$	20061	$0,530^{efghijkA}$	20982	$0,825^{cd}$	16
27	*	*	$0,510^{mnopA}$	20701	$0,525^{efghijkA}$	18876	$0,825^{cd}$	16
28	*	*	$0,505^{nopA}$	21278	$0,520^{fghijkA}$	16630	$0,820^{cd}$	15
29	*	*	$0,505^{nopA}$	21777	$0,515^{fghijkA}$	14181	$0,825^{cd}$	16
30	*	*	$0,5^{nopA}$	22214	$0,510^{ghijkA}$	11669	$0,820^{cd}$	15
31	*	*	$0,494^{opA}$	22557	$0,505^{hijkA}$	8970	$0,831^{cd}$	16
32	*	*	$0,489^{pA}$	22838	$0,5^{ijkA}$	6193	$0,826^{cd}$	16
33	*	*	$0,484^{pA}$	23025	$0,489^{jkA}$	3183	$0,826^{cd}$	32
34	*	*	$0,479^{pA}$	23119	$0,479^{kA}$	94	0,320	94

Nota: Letras minúsculas representam médias estatisticamente iguais nas colunas, e letras maiúsculas representam médias estatisticamente iguais nas linhas a 5% de significância. Valores em destaque representam os melhores resultados. * não foi possível obter resultados (estouro de pilha).

No gráfico da Figura 4.11, se pode observar que somente PCA encontra taxas de acuidade altas (superiores a 75%), com acréscimo da ultima característica essa taxa decai drasticamente, pois com a inserção dessa característica insere-se alta correlação e redundância entre as variáveis. Observa-se também que as taxas obtidas pelos métodos SFS, SBS e BRD decaem com o acréscimo de características, caracterizando o problema da dimensionalidade.

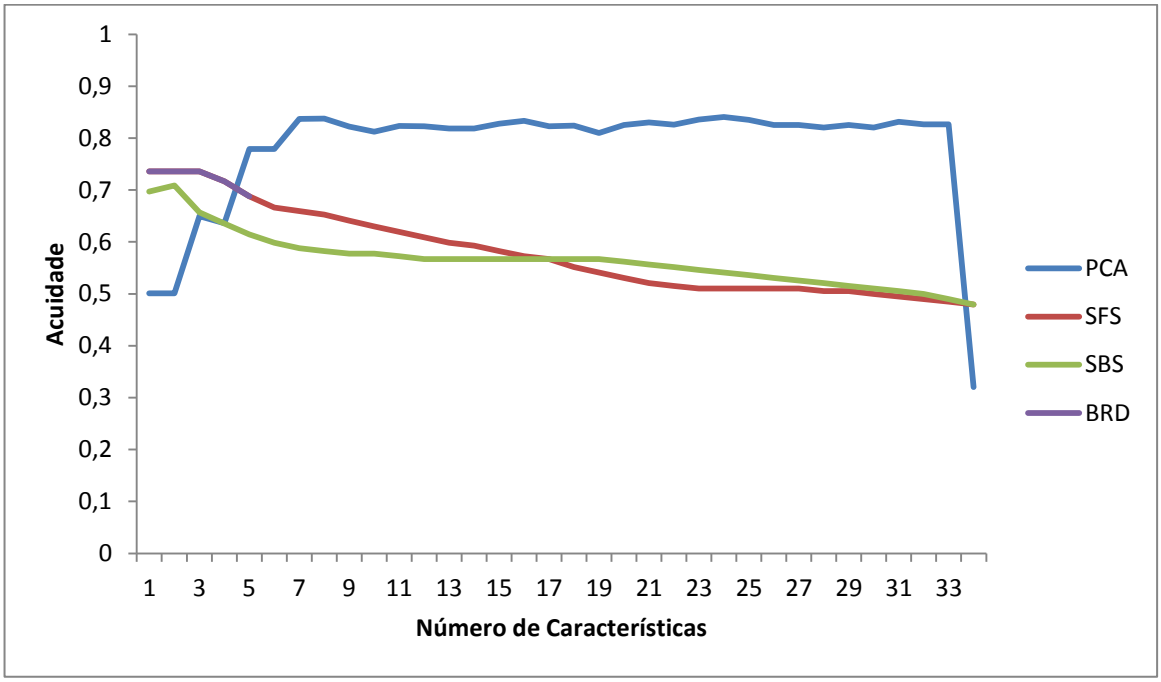


Figura 4.11 - Precisão *versus* o Número de Características *Ionosphere*

No gráfico da Figura 4.12, observa-se que o tempo necessário para encontrar uma solução utilizando BDR cresce de forma exponencial. O tempo gasto pelo PCA mantém-se muito abaixo dos valores encontrados pelos outros métodos.

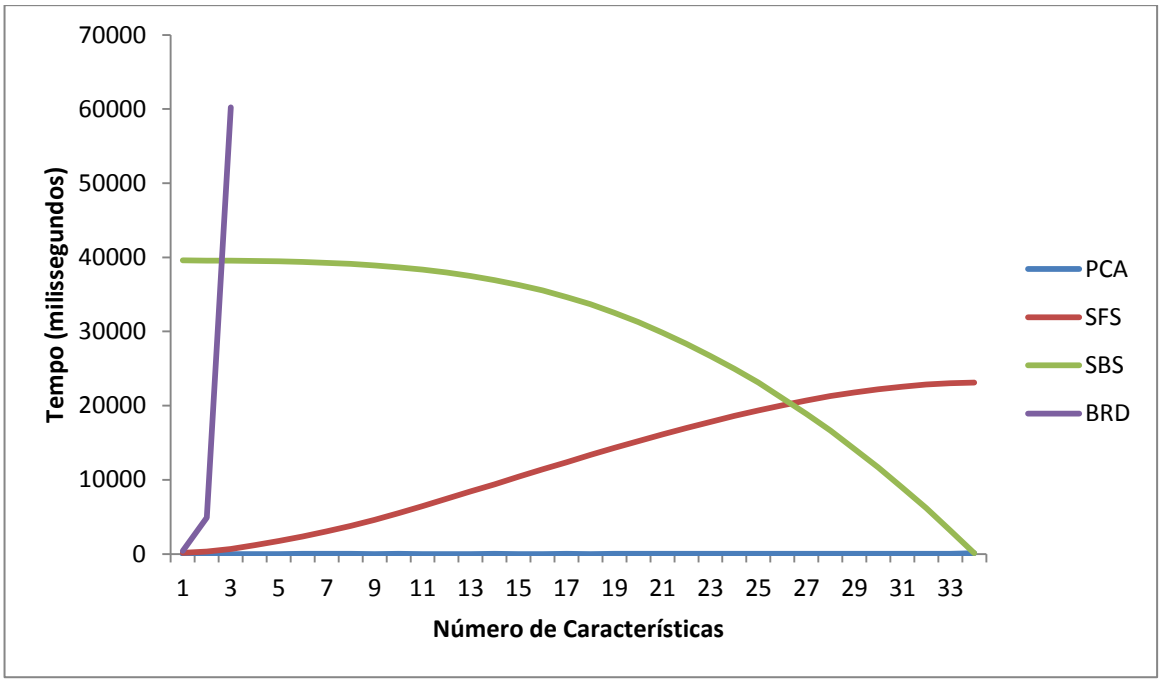


Figura 4.12 - Tempo *versus* o Número de Características *Ionosphere*

4.9 LIBRAS *Movement*

A base de dados LIBRAS *Movement* foi criada por Daniel Baptista Dias, Sarajane Marques Peres e Helton Hideraldo Bíscaro, da Universidade de São Paulo - USP em 2009. Esta base de dados contém informações de movimentos da mão que representam o tipo de sinal LIBRAS (Língua Brasileira de Sinais). É composta de 90 atributos numéricos que representam a posição da mão em cada instante de tempo em um vídeo de 45 frames, além de um atributo categórico que define a classe do objeto, neste caso, um dos 15 tipos de movimento LIBRAS. Esta base de dados é composta por 360 amostras com a igualitária distribuição das classes.

Na Tabela 4.8, observa-se que os resultados encontrados pelos métodos SFS, SBS e BRD são encontrados na primeira iteração. E que todos os resultados encontrados são estatisticamente iguais a uma significância de 5% entre si. Esse valor não ultrapassa os 25%, sendo assim uma taxa de acerto baixa. PCA encontra seu resultado com 10 características, com uma taxa de acuidade de 64,8% aproximando-se assim de taxas de alta precisão.

Tabela 4.8 - Avaliação Experimental na base de dados Libras *Movement*

	BRD		**SFS**		**SBS**		**PCA**	
n	Acuidade	Tempo	Acuidade	Tempo	Acuidade	Tempo	Acuidade	Tempo
1	**0,147^{aA}**	1185	**0,147^{aA}**	572	0,140^{aA}	28449	0,087	1
2	0,181^{aA}	36987	0,180^{aA}	1547	0,160^{aA}	28437	0,133^{A}	2
3	0,177^{aA}	1271652	0,166^{aA}	2600	0,155^{aA}	28412	0,257	2
4	*	*	0,163^{aA}	3919	0,155^{aA}	28379	0,325^{a}	3
5	*	*	0,160^{aA}	5514	0,155^{aA}	28340	0,360^{a}	3
6	*	*	0,160^{aA}	7392	0,155^{aA}	28293	0,531^{b}	4
7	*	*	0,158^{aA}	9581	0,155^{aA}	28240	0,552^{b}	3
8	*	*	0,158^{aA}	12111	0,155^{aA}	28178	0,592^{bc}	4
9	*	*	0,158^{aA}	15102	0,155^{aA}	28110	0,596^{bc}	5
10	*	*	0,158^{aA}	18001	0,155^{aA}	28030	**0,648^{cd}**	5
11	*	*	0,158^{aA}	21179	0,155^{aA}	27947	0,668^{de}	6
12	*	*	0,158^{aA}	24721	0,155^{aA}	27850	0,713^{def}	8
13	*	*	0,158^{aA}	28414	0,155^{aA}	27749	0,711^{def}	6

14	*	*	0,155^{aA}	32307	0,155^{aA}	27633	0,731^{ef}	8
15	*	*	0,155^{aA}	36374	0,155^{aA}	27516	0,730^{ef}	8
16	*	*	0,155^{aA}	40713	0,155^{aA}	27386	0,733^{ef}	9
17	*	*	0,155^{aA}	45304	0,155^{aA}	27256	0,721^{ef}	9
18	*	*	0,155^{aA}	49710	0,155^{aA}	27111	0,732^{f}	10
19	*	*	0,155^{aA}	54171	0,155^{aA}	26963	0,722e^{f}	10
20	*	*	0,155^{aA}	58664	0,155^{aA}	26807	0,708^{def}	11
21	*	*	0,155^{aA}	63251	0,155^{aA}	26641	0,718^{df}	13
22	*	*	0,155^{aA}	67931	0,155^{aA}	26457	0,723^{ef}	12
23	*	*	0,155^{aA}	72876	0,155^{aA}	26291	0,715^{def}	13
24	*	*	0,155^{aA}	77743	0,155^{aA}	26111	0,739^{f}	13
25	*	*	0,155^{aA}	82719	0,155^{aA}	25930	0,736^{f}	16
26	*	*	0,155^{aA}	87612	0,155^{aA}	25735	0,739^{f}	15
27	*	*	0,155^{aA}	92664	0,155^{aA}	25537	0,745^{f}	18
28	*	*	0,155^{aA}	97812	0,155^{aA}	25334	0,739^{f}	17
29	*	*	0,155^{aA}	102991	0,155^{aA}	25136	0,732^{ef}	17
30	*	*	0,155^{aA}	108326	0,155^{aA}	24915	0,730^{ef}	23
31	*	*	0,155^{aA}	113802	0,155^{aA}	24695	0,721^{ef}	19
32	*	*	0,155^{aA}	119092	0,155^{aA}	24462	0,728^{ef}	19
33	*	*	0,155^{aA}	124386	0,155^{aA}	24238	0,725^{ef}	21
34	*	*	0,155^{aA}	129624	0,155^{aA}	23981	0,720^{ef}	52
35	*	*	0,155^{aA}	134902	0,155^{aA}	23745	0,719^{ef}	31
36	*	*	0,155^{aA}	140231	0,155^{aA}	23488	0,700^{def}	31
37	*	*	0,155^{aA}	145565	0,155^{aA}	23244	0,711^{def}	31
38	*	*	0,155^{aA}	150894	0,155^{aA}	22974	0,723^{ef}	15
39	*	*	0,155^{aA}	156326	0,155^{aA}	22706	0,707^{def}	32
40	*	*	0,155^{aA}	161680	0,155^{aA}	22424	0,689^{def}	16
41	*	*	0,155^{aA}	167205	0,155^{aA}	22161	0,709^{def}	32
42	*	*	0,155^{aA}	172537	0,155^{aA}	21868	0,701^{def}	47
43	*	*	0,155^{aA}	177840	0,155^{aA}	21581	0,716^{ef}	31
44	*	*	0,155^{aA}	183311	0,155^{aA}	21274	0,718^{ef}	31
45	*	*	0,155^{aA}	188708	0,155^{aA}	20979	0,707^{de}	31

46	*	*	0,155^{aA}	194106	0,155^{aA}	20647	0,708^{def}	31
47	*	*	0,155^{aA}	199425	0,155^{aA}	20336	0,713^{def}	31
48	*	*	0,155^{aA}	204823	0,155^{aA}	20003	0,706^{def}	31
49	*	*	0,155^{aA}	210127	0,155^{aA}	19670	0,713^{def}	31
50	*	*	0,155^{aA}	215322	0,155^{aA}	19315	0,710^{def}	46
51	*	*	0,155^{aA}	220532	0,155^{aA}	18981	0,717^{ef}	47
52	*	*	0,155^{aA}	225711	0,155^{aA}	18591	0,720^{ef}	47
53	*	*	0,155^{aA}	230906	0,155^{aA}	18244	0,720^{ef}	47
54	*	*	0,155^{aA}	236148	0,155^{aA}	17822	0,722^{ef}	47
55	*	*	0,155^{aA}	241093	0,155^{aA}	17449	0,721^{ef}	32
56	*	*	0,155^{aA}	245984	0,155^{aA}	17065	0,730^{ef}	47
57	*	*	0,155^{aA}	250812	0,155^{aA}	16697	0,722^{ef}	47
58	*	*	0,155^{aA}	255539	0,155^{aA}	16299	0,725^{ef}	46
59	*	*	0,155^{aA}	260195	0,155^{aA}	15914	0,727^{ef}	47
60	*	*	0,155^{aA}	264742	0,155^{aA}	15473	0,732^{ef}	46
61	*	*	0,155^{aA}	269176	0,155^{aA}	15076	0,733^{ef}	46
62	*	*	0,155^{aA}	273534	0,155^{aA}	14647	0,737^{f}	47
63	*	*	0,155^{aA}	277806	0,155^{aA}	14241	0,728^{ef}	47
64	*	*	0,155^{aA}	281921	0,155^{aA}	13805	0,724^{ef}	46
65	*	*	0,155^{aA}	285970	0,155^{aA}	13359	0,726^{ef}	63
66	*	*	0,155^{aA}	289856	0,155^{aA}	12902	0,723^{ef}	47
67	*	*	0,155^{aA}	293659	0,155^{aA}	12417	0,737^{f}	62
68	*	*	0,155^{aA}	297294	0,155^{aA}	11949	0,739^{f}	47
69	*	*	0,155^{aA}	300866	0,155^{aA}	11481	0,734^{ef}	62
70	*	*	0,155^{aA}	304312	0,155^{aA}	11013	0,737^{f}	62
71	*	*	0,155^{aA}	307682	0,155^{aA}	10561	0,740^{f}	62
72	*	*	0,155^{aA}	311043	0,155^{aA}	10077	0,736^{f}	63
73	*	*	0,155^{aA}	314413	0,155^{aA}	9609	0,733^{ef}	63
74	*	*	0,155^{aA}	317423	0,155^{aA}	9094	0,729^{ef}	63
75	*	*	0,155^{aA}	320294	0,155^{aA}	8580	0,729^{ef}	62
76	*	*	0,155^{aA}	323039	0,155^{aA}	8049	0,724^{ef}	78
77	*	*	0,155^{aA}	325582	0,155^{aA}	7566	0,721^{ef}	63

78	*	*	0,155^{aA}	327969	0,155^{aA}	7020	0,718^{ef}	78
79	*	*	0,155^{aA}	330184	0,155^{aA}	6520	0,718^{ef}	62
80	*	*	0,155^{aA}	332259	0,155^{aA}	5959	0,725^{ef}	62
81	*	*	0,155^{aA}	334147	0,155^{aA}	5444	0,714^{def}	78
82	*	*	0,155^{aA}	335909	0,155^{aA}	4882	0,716^{ef}	78
83	*	*	0,155^{aA}	337469	0,155^{aA}	4321	0,732^{ef}	78
84	*	*	0,155^{aA}	338858	0,155^{aA}	3728	0,740^{f}	78
85	*	*	0,155^{aA}	340003	0,155^{aA}	3182	0,732^{ef}	78
86	*	*	0,155^{aA}	341002	0,155^{aA}	2589	0,744^{f}	78
87	*	*	0,155^{aA}	341782	0,155^{aA}	2012	0,744^{f}	78
88	*	*	0,155^{aA}	342390	0,155^{aA}	1404	0,739^{f}	93
89	*	*	0,155^{aA}	342811	0,155^{aA}	811	0,732e^{f}	94
90	*	*	0,155^{aA}	343014	**0,155^{aA}**	202	0,737^{f}	203

Nota: Letras minúsculas representam médias estatisticamente iguais nas colunas, e letras maiúsculas representam médias estatisticamente iguais nas linhas a 5% de significância. Valores em destaque representam os melhores resultados. * não foi possível obter resultados (estouro de pilha).

No gráfico da Figura 4.13, observa-se a grande diferença nas taxas encontrada pelo PCA aos demais métodos.

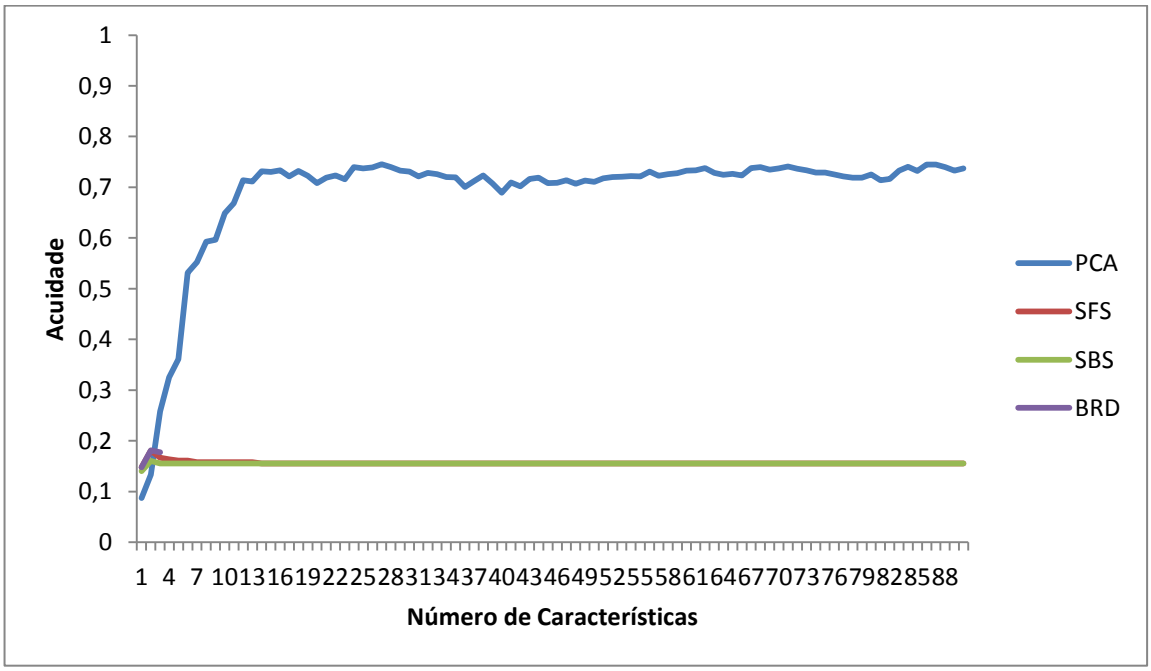


Gráfico 4.13 - Precisão *versus* o Número de Características - Libras *Movement*

No gráfico da Figura 4.14, observa-se o crescimento exponencial do tempo necessário pelo BRD para encontrar uma solução. Também se observa o baixo tempo necessário pela PCA para encontrar sua solução.

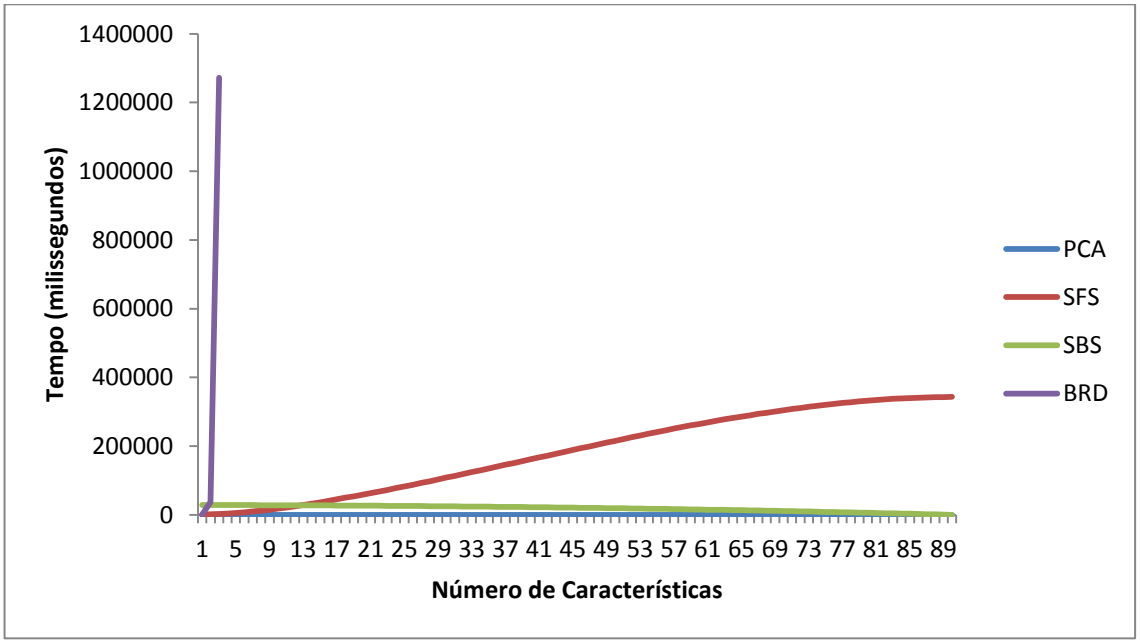


Gráfico 4.14 - Tempo *versus* o Número de Características - Libras *Movement*

Capítulo 5

Considerações Finais

Este trabalho estudou método de Extração de Características por transformação linear utilizando a Análise de Componentes Principais e Métodos de Seleção de Características utilizando o modelo *Warper* com o classificador *Naive Bayes*.

Após breve descrição dos métodos, apresenta-se uma avaliação experimental dos mesmos utilizando bases de dados do repositório UCI e como métrica de avaliação a taxa de acerto (acuidade) e o tempo utilizado pelo método para encontrar o conjunto solução.

A partir dessa avaliação pode-se observar o comportamento dos métodos e a importância dos métodos de Redução de Dimensionalidade, pois reduzem o Problema da Dimensionalidade aumentando a taxa de acerto do classificador.

Nas bases de dados apresentadas, os métodos BRD e SFS obtiveram melhor resultados em relação ao SBS ao que se diz respeito ao problema da dimensionalidade. Isso se deve ao fato de que o Problema da Dimensionalidade começa a ocorrer mais próximo do conjunto vazio do que ao conjunto completo, assim os métodos BRD e SFS necessitam de um menor número de iterações.

PCA demonstrou-se ser uma técnica de baixo custo computacional, executando de um tempo inferior às demais, e que apresenta bons resultados na redução da dimensionalidade. Nos testes realizados gerou conjuntos de dados que obtiveram uma taxa de acuidade superior ao SFS, SBS e BRD isso com um custo computacional (tempo de execução) inferior.

Pode-se observar que com dimensionalidades elevadas torna-se praticamente inviável explorar todo o espaço de busca, pois necessita de um maior espaço em memória e um maior tempo de execução, impossibilitando assim a utilização de métodos de força bruta (BRD) em bases de dados de alta dimensionalidade.

A partir dos resultados apresentados, pôde-se concluir que, nas bases estudadas e utilizando o classificador *Naive Bayes*, PCA obtém um desempenho, tanto em tempo de execução quanto em taxa de acuidade, superior aos demais métodos de Redução de Dimensionalidade aqui apresentados.

A partir dos estudos realizados durante a elaboração desta monografia, outros trabalhos mostram-se de interesse:

- Implementação de outros métodos de Seleção de Características, a exemplo de Busca Oscilatória, Busca Sequencial Flutuante Progressiva, Ramificar e Podar, entre outros.
- Implementação de outros métodos de Extração de Características, como Análise de Componentes Independentes, *Kernel* PCA, *Wavelets*, entre outros.
- Avaliação dos métodos e Extração e Seleção utilizando outros classificadores, por exemplo, C4.5, KNN, Redes Neurais Artificiais, Máquina de Comitê, entre outros.
- Avaliação dos métodos e Extração e Seleção utilizando técnicas de agrupamento de dados, como os algoritmos baseados em Colônia de Formigas, K-média, entre outros.
- Utilização dos métodos para resolução de problemas reais.

Referências

BENFATTI, E. W.; BONIFACIO, F. N.; GIRARDELLO, A. D.; BOSCARIOLI, C. *Descrição da Arquitetura e Projeto da Ferramenta YADMT - Yet Another Data Mining Tool*. Relatório Técnico nº 01 do Curso de Ciência da Computação, UNIOESTE, Campus de Cascavel, 2011.

BENFATTI, E. W. *Um estudo sobre a aplicação dos algoritmos KNN, C45 e redes de Bayes na classificação de dados*. Curso de Ciência da Computação, UNIOESTE, Campus de Cascavel, 2010.

BONIFACIO, F. N. *Comparação entre as Redes Neurais Artificiais MLP, RBF e LVQ na Classificação de Dados*. Curso de Ciência da Computação, UNIOESTE, Campus de Cascavel, 2010.

BORGES, J.; KLAUTAU, A. *Inteligência artificial aplicada à supervisão da qualidade de energia.* Revista Científica da UFPA, Belém, PA, v. 6, n. 1, Janeiro 2007. ISSN 1981-6014.

CORTES, S. C.; PORCARO, R. M.; LIFSCHITZ, S.; *Mineração de dados – Funcionalidades, Técnicas e Abordagens*, PUC Rio Inf. MCC10/02, Maio, 2002.

COSTA, E. de P. *Investigação de técnicas de classificação hierárquica para problemas de bioinformática*. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) — UFCG - Universidade Federal de Campina Grande, Campina Grande, PB, Março 2008.

DANZIGER, M. *Sistema Híbrido de Detecção de Intrusão em Redes IEEE 802.11 Baseado na Teoria do Perigo e Sistemas Multi-agentes* (MAWIDS-DT). Dissertação (Mestrado em Ciência da Computação) – UPE - Universidade de Pernambuco, Recife, PE, Março 2010.

DE CAMPOS, T. E. *Técnicas de Seleção de Características com Aplicações em Reconhecimento de Faces*. Dissertação de Mestrado em Ciência da Computação, Instituto de Matemática e Estatística. Universidade de São Paulo – IME/ USP, São Paulo, SP: Maio, 2001.

ELMASRI, R.; NAVATHE, S.; *Conceitos de Data Mining.* In: Sistemas de Banco de Dados. São Paulo: Pearson Addison Wesley, 2005.

FIX, E.; JR, H. *Discriminatory Analysis - Nonparametric Discrimination: Small Sample Performance.* Berkeley, California, 1952. 1-43 p.

FRANK, A.; ASUNCION, A. *UCI Machine Learning Repository*, Irvine, CA: University of California, School of Information and Computer Science, 2010. Disponível em: http://archive.ics.uci.edu/ml Data de último acesso: 26/03/2012.

FUZARO, M. Jr, GULIATO, D., *Desenvolvimento de métodos para redução de dimensionalidade do espaço de características para reconhecimento de padrões*, Horizonte Científico, VOL. 4, Nº 2, 2010.

HAN, J.; KAMBER, M. *Data Mining – Concepts and Techniques*, Morgan Kaufmann Publishers, Inc, 2001.

HAN, J.; KAMBER, M. *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. ISBN 1-55860-901-6.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. *Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence*, Washington, DC, USA, v.22, n.1, p.4-37, Janeiro, 2000.

JAIN, A. K.; ZONGKER, D. *Feature selection: Evaluation, application, and small sample performance.* IEEE Transactions on Pattern Analysis and Machine Intelligence, [S.1.], v.19, n.2, p.153-158, Fevereiro, 1997.

JOLLIFFE, I. T. *Principal Component Analysis*, 2nd Edition, New York: Springer-Verlag, New York, 2002.

KOHAVI, R. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. Stanford, CA, USA: Stanford University, 1995.

KOUTROUMBAS, K.; THEODORIDIS, S.; *Pattern Recognition*, Ed Academic Press, 2008.

LEE, H. D. *Seleção de Atributos Importantes para a Extração de Conhecimentos de Bases de Dados*. São Carlos, SP: Instituto de Ciências Matemáticas e de Computação de São Paulo – ICMC- USP, Dezembro, 2005.

LEE, H. D. *Seleção e Construção de Features Relevantes para o Aprendizado de Máquina.* São Carlos, SP: Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo – ICMC-USP, Março, 2000.

LIU, H.; MOTODA, H. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer international series in engineering and computer science. 2. Ed. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

LIU, H.; YU, L. *Toward integrating feature selection algorithms for classification and clustering,* Knowledge and Data Engineering, IEEE Transactions on , vol.17, no.4, pp. 491-502, April 2005.

MARTINS JR., D. C. *Redução de Dimensionalidade Utilizando Entropia Condicional Média Aplicada a Problemas de Bioinformática e de Processamento de Imagens*. São Paulo, SP: Instituto de Matemática e Estatística da Universidade de São Paulo – IME-USP, Dezembro, 2004.

MICHAEL J. A.; BERRY, G. L. *Data Mining Techniques for Marketing, Sales, and Customer Support*, John Wiley & Sons, Inc., 1997.

NAVEGA, S. *Princípios Essenciais do Data Mining*. In INFOIMAGEM, CENADEM, São Paulo. 2002.

OGURI, P. *Aprendizado de Máquina para o Problema de Sentiment Classification.* Dissertação (Mestrado em Informática) – PUCRio - Pontifica Universidade Católica do Rio de Janeiro, Rio de Janeiro, RJ, Outubro 2006.

PERLOVSKY, L. I. *Conundrum of combinatorial complexity*. IEEE Trans. On Pattern Analysis and Machine Intelligence, 20(6) : 666-670, 1998.

SANTORO, D. M. *Sobre o Processo de Seleção de Atributos – As Abordagens Filtro e Wrapper*. São Carlos, SP: Universidade Federal de São Carlos – UFSCar, Abril, 2005.

SIEDLECKI, W. W.; SKLANSKY, J. *On automatic feature selection*. International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), v.2, n.2, p. 197-220, Junho, 1988.

SMITH, L. I.; *A tutorial on Principal Components Analysis*, Fevereiro, 2002.

THOMÉ, A. C. G., *Redes Neurais, uma ferramenta para KDD e Data Mining*. Apostila. 2008. Disponível em: http://equipe.nce.ufrj.br/thome/grad/nn/mat_didatico/apostila_kdd_mbi.pdf Data de último acesso: 18/05/2011.

WEKA. *Waikato Environment for Knowledge Analysis*. Disponível em: http://www.cs.waikato.ac.nz/~ml/weka/. Consultado na Internet em: 01/03/2012.

WITTEN, I. H.; FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques*. 2. ed. San Francisco, CA, USA: Morgan Kaufmann, 2005. ISBN 0-12-088407-0.

ZANOTTA, D. C. *Detecção de queimadas a partir de técnicas automáticas e operadores morfológicos de erosão/dilatação usando imagens de sensoriamento remoto*. In: 2o Simpósio de Geotecnologias no Pantanal. Corumbá, MS: Unesp/Rio Claro, 2009. p. 666–673.

ZHENG, Z. *A Benchmark for Classifier Learning*. N.S.W Australia, 2006.

ZONGKER, D.; JAIN, A. K. *Algorithms for feature selection: An evaluation*. In: PROCEEDINGS OF THE 13TH INTERNATIONAL CONFERENCE ON PATHERN RECOGNITION (ICPR´96) – VOLUME 2, 1996. Proceeding… Los Alamitos, CA, USA: IEEE Computer society, 1996. V.2, p.18-22.