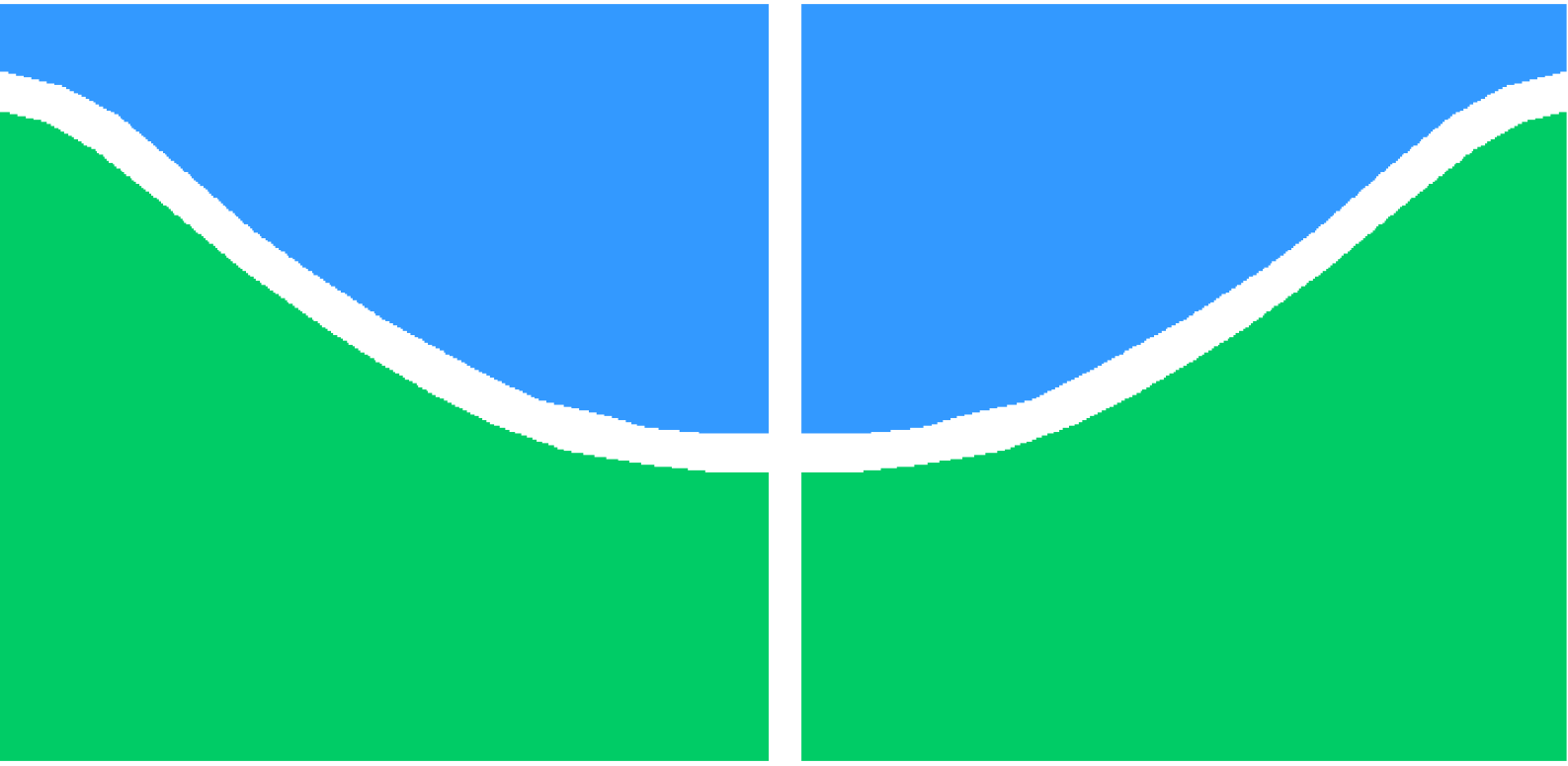

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Engenharia de Software

Serviço de Seleção de Característcas para Linguagem de Aprendizado de Máquina no Contexto de Redes de Computadores (?)

Autor: Vinícius Franco da Cunha Arantes

Orientador: Professor Dr. Fabricio Ataides Braz

Brasília, DF

2013

Vinícius Franco da Cunha Arantes

Serviço de Seleção de Característcas para Linguagem de Aprendizado de Máquina no Contexto de Redes de Computadores (?)

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Professor Dr. Fabricio Ataides Braz

Brasília, DF

2013

Vinícius Franco da Cunha Arantes

Serviço de Seleção de Característcas para Linguagem de Aprendizado de Máquina no Contexto de Redes de Computadores (?)/ Vinícius Franco da Cunha Arantes. – Brasília, DF, 2013-

45 p. : il. (algumas color.) ; 30 cm.

Orientador: Professor Dr. Fabricio Ataides Braz

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2013.

1. Palavra-chave01. 2. Palavra-chave02. I. Professor Dr. Fabricio Ataides Braz. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Serviço de Seleção de Característcas para Linguagem de Aprendizado de Máquina no Contexto de Redes de Computadores (?)

CDU 02:141:005.6

Vinícius Franco da Cunha Arantes

Serviço de Seleção de Característcas para Linguagem de Aprendizado de Máquina no Contexto de Redes de Computadores (?)

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 27 de novembro de 2015:

Professor Dr. Fabricio Ataides Braz
Orientador

Professor Dr. Luiz Augusto Laranjeira
Convidado 1

Professor Dr. Nilton Corrêa
Convidado 2

Brasília, DF
2013

A dedicatória é opcional. Caso não deseje uma, deixar todo este arquivo em branco.

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

Agradecimentos

A inclusão desta seção de agradecimentos é opcional, portanto, sua inclusão fica a critério do(s) autor(es), que caso deseje(em) fazê-lo deverá(ão) utilizar este espaço, seguindo a formatação de *espaço simples e fonte padrão do texto (arial ou times, tamanho 12 sem negritos, aspas ou itálico.*

Caso não deseje utilizar os agradecimentos, deixar toda este arquivo em branco.

Resumo

O resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (. . .) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto. O texto pode conter no mínimo 150 e no máximo 500 palavras, é aconselhável que sejam utilizadas 200 palavras. E não se separa o texto do resumo em parágrafos.

Palavras-chaves: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

Fig.	Area of the i^{th} component
456	Isto é um número
123	Isto é outro número
lauro cesar	este é o meu nome

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
$\in$	Pertence

Sumário

1 Introdução

1.1 Motivação

Na sociedade atual, cada vez mais milhares de dados são gerados diariamente e há uma dificuldade em se aproveitar ao máximo do conteúdo desses dados, uma vez que é impossível analisar esses dados individualmente para poder identificar o seu real valor. Segundo o Business Intelligence Observatory (2013), cerca de 90% dos dados de hoje foram gerados nos últimos dois anos, tendo como fontes documentos de textos, e-mails, streaming de vídeos e áudio, transações comerciais, telecomunicações e diversas outras. Estimas-se que valor investido para trabalhar esses dados tende a superar a faixa de 27 bilhões em 2015 Observatory (2013).

A fim de analisar esses dados, técnicas foram desenvolvidas e ferramentas foram criadas, tais meios visam facilitar a análise dessa grande massa de dados extraindo valor dela. Um questionário levantado pela Bloomberg Businessweek (2011) mostra que 97% das companhias com rendas acima de 100 milhões de dólares usam algum tipo de análise de negócio, e que essa análise se baseia em grandes volumes de dados e seu valor agregado à companhia.

Para realizar essas análises, várias empresas tem feito investimentos na área de Aprendizado de Máquina *(machine learning)*. A área de Aprendizado de Máquina é considerada um ramo da Inteligência Artificial, sendo umaárea especializada no estudo e construção de sistemas que sejam capazes de aprender de forma automatizada a partir de dados (BRINK; RICHARDS, 2014). Dentro da área de Aprendizado de Máquina existem vários passos para se construir um modelo, e um desses passos é o Processo de Seleção de Características. Uma característica é uma informação que é potencialmente útil para realizar predições (MITCHELL, 1997).

Problemas de Aprendizado de Máquina costumam gerar muitos dados que aparentam ser úteis, mas que nem sempre definem com exatidão o problema, ou que ajudem a alcançar o objetivo traçado na construção do modelo, de acordo com Guyon (2003), muitos problemas de Aprendizado de Máquina chegam a variar entre cinco mil a cinquenta mil características que, inicialmente, são consideradas importantes ou relevantes ao problema. Esse grande número de características que são levantadas inicialmente demandam muito tempo para serem processadas, levando a necessidade de reduzir esse número para auxiliar na sua análise. O elevado número ainda acarreta no possível aparecimento de ruídos durante a análise dos dados. A Seleção de Características visa auxiliar esse problema motivado pelas técnicas para lidar com esses dados. Com essa abordagem é possível

escolher os melhores subconjuntos de características que conseguem alcançar os objetivos propostos pelo modelo.

1.2 Objetivos

Esse trabalho tem como objetivo principal a implementação de um serviço que seja capaz de analisar um conjunto de características e extrair o melhor subconjunto possível analisando-o em meio a três modelos de seleção de características.

1.3 Objetivos Específicos

Para poder alcançar o objetivo geral, esse trabalho foi divido em etapas. Os objetivos específicos do trabalho são:

1. Especificar o processo de seleção de características
2. Pesquisar modelos de seleção de características a serem utilizados
3. Implementar os modelos de seleção de características
4. Realizar laboratório nos modelos com testes focados em uma base de dados classica (a ser determinada)
5. Específicar arquitetura do serviço
6. Construir protótipo do serviço
7. Realizar laboratório no protótipo do serviço
8. Coletar e relatar resultados

1.4 Justificativa

Esse trabalho se torna importante, do ponto de vista prático, uma vez que os recursos computacionais nem sempre conseguem processar as grandes massas de dados contidas em uma certa base de dados. Esse problema tende a ser contornado com o uso do processo de seleção de características, uma vez que ele reduz o número de caracterítiscas a serem utilizados no modelo de Aprendizado de Máquina reduzindo o consumo computacional e agilizando o treinamento e execução dos modelos. Será mostrado também que o uso do processo de seleção de características melhora o desempnho dos classificadores, já que um problema com muitas características tende a gerar muito ruído.

EM CONSTRUÇÃO.

1.5 Método

1.5.1 Etapa 1: Especificar o processo de seleção de características

A primeira etapa do trabalho consiste em elucidar e específicar como funciona o processo para seleção de características. Para tal feito será necessário fazer uma pesquisa bibliográfica visando demonstrar o fluxo do processo de seleção e suas principais etapas.

1.5.2 Etapa 2: Pesquisar modelos de seleção de características a serem utilizados

Essa etapa consiste em pesquisar os modelos existentes e como são compostos, levando em consideração o processo estudado na etapa anterior e se esses modelos estão consistentes ao processo de seleção de características em si. Então, a partir da pesquisa, escolher os modelos que serão utilizados no serviço a ser implementado.

1.5.3 Etapa 3: Implementar os modelos de seleção de características

Para cada um dos modelos selecionados, será necessário realizar a sua implementação, seja através de bibliotecas de terceiros ou realizar sua completa implementação. Esses modelos devem ser implementados de forma a receberem um mesmo tipo de dados para que seja possível realizar uma avaliação da sua performance.

1.5.4 Etapa 4: Realizar laboratório nos modelos com testes focados em uma base de dados classica

Os modelos implementados deverão ser testados utilizando uma base de dados clássica para que seja possível ter um maior controle sobre os resultados a serem avaliados. O uso da base classica se dá justamente pelo fato de já terem sido utilizadas em outros projetos, trazendo assim uma maior confiabilidade na hora de comparar os resultados.

1.5.5 Etapa 5: Específicar arquitetura do serviço

Nessa etapa será feito a especificação de como o serviço proposto por esse trabalho funcionará, descrevendo cada um dos seus módulos e como eles se comunicarão.

1.5.6 Etapa 6: Construir protótipo do serviço

Após feito a especificação da arquitetura, será feito um protótipo de como o serviço funcionará, com todas as funcionalidades levantadas nas etapas anteriores. Esse protótipo deverá ser capaz de receber um conjunto de características e poder fazer o processo de seleção utilizando os modelos selecionados.

1.5.7 Etapa 7: Realizar laboratório no protótipo do serviço

Com a implementação do protótipo do serviço, será feito um laboratório para garantir que as funcionalidades implementadas estão funcionando de acordo com o projetado, sendo assim, essa etapa consiste em testar o serviço e sua integridade.

1.5.8 Etapa 8: Coletar e relatar resultados

Essa será a etapa final deste trabalho, consistindo em coletar, analisar e relatar os resultados obtidos pelo serviço, sendo essa fase de suma importância para o trabalho.

2 Seleção de Características

2.1 Aprendizado de Máquina (*Machine Learning*)

A área de Aprendizado de Máquina é considerada um ramo da área de Inteligência Artificial, sendo uma área especializada no estudo e construção de sistemas que sejam capazes de aprender de forma automatizada a partir de dados (BRINK; RICHARDS, 2014). Desde que os computadores foram inventados, pesquisadores tentam procurar uma maneira de fazer com que eles possam aprender e melhorar sua performance através da experiência. Se fosse possível fazer com que os computadores pudessem aprender, diagnósticos médicos poderiam ser feitos, seria possível escolher as matérias de jornais que seriam entregues às pessoas de acordo com seus gostos, seria possível criar uma nova maneira de lidar com problemas e situações (MITCHELL, 1997).

Por definição, Mitchell (1997) diz que um dado programa de computador é capaz de aprender com experiência E, com respeito a um grupo de tarefas T e índice de performance P, se a performance medida por P em T aumenta com E. Ou seja, se conforme o tempo passa, a performance aumenta, ele estará aprendendo.

2.2 Processo de Seleção

Seleção de Características é um processo que seleciona um subconjunto de um conjunto original de características, a optimicidade do subconjunto é medida por um critéro de avaliação. (YU, 2005). As características são fundamentais para a solução de um dado problema de Aprendizado de Máquina, e quão melhor for essa seleção, melhor será o resultado obtido assim como o tempo necessário para resolve-lo, em termos de acertividade e consumo de recurso computacional.

As características podem ser de dois tipos: numéricas ou categóricas. As numéricas são representadas por valores, como o próprio nome surgere, já a categórica visa dividir as características em diferentes categorias, não importando a sua ordem.

O processo de seleção de características geralmente segue quatro passos conforme ilustrado pela Figura 1, sendo eles, geração de subconjuntos (*subset generation*), avaliação do subconjunto (*subset evaluation*), critério de parada (*stopping criterion*), e validação de resultados (*result validation*). De acordo com Molina (2002) um subconjunto ideal é aquele que segue uma das seguintes abordagens: a) um subconjuto que possua um tamanho específico que otimize a forma de avaliação. b) um subconjunto de tamanho menor que satisfaça alguma restrição na forma de avaliação. c) o subconjunto com o

melhor desempenho levando em consideração sua dimensão e o valor de sua medida de avaliação.

De acordo com Yu (2005), a forma de avaliar os subconjuntos gerados pelos algoritmos de seleção de características geralmente caem em três categorias: o modelo de filtro (*filter*), o modelo de envelopamento (*wrapper*) e o modelo híbrido (*hybrid*). O modelo de filtro, de acordo com Yu (2005), se baseia nas características gerais dos dados para avaliar os subconjutos de características e não utiliza nenhum tipo de algorítimo de mineração de dados para poder realizar essa avaliação. Já o modelo de envelopamento necessita de pelo menos um algoritimo pré determinado, para poder realizar a avaliação dos subconjuntos de características. O modelo de envelopamento geralmente tende a consumir mais em questões computacionais. E o modelo hibrido se aproveita de ambas técnicas para poder avaliar os subconjuntos

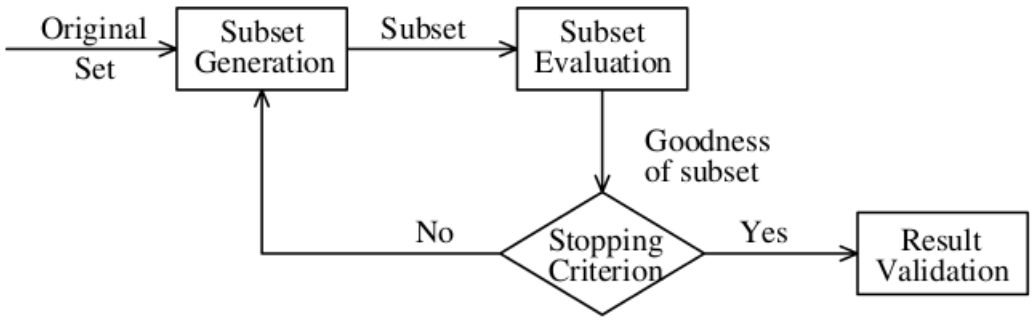

Figura 1 – Fluxograma de seleção de características. (YU, 2005)

2.3 Geração de Subconjuntos

O processo de seleção de características pode ser visto como um problema de busca em que cada um dos estados da busca representa um dos subconjuntos de características (LIU; MOTODA., 1998). Imagine um conjunto com três características. Se forem colocadas em um vetor, se posicionariam da seguinte maneira (A1, A2, A3). Tendo em vista que o valor 1 representa a presença da características no subconjunto a ser gerado, temos que em (1, 0, 0) apenas a característica A1 está presente nesse subconjunto. Variando todas as possibilidades partindo de um conjunto com todas as características e chegando a um conjunto com nenhuma. Tal processo está ilustrado na figura 2.

Existem quatro formas de realizar a busca: *sequential forward generation (SFG)*, *sequential backward generation (SBG)*, *bi-drectional generation*, e *random generation*. A primeira consiste em começar com um conjunto vazio e ir adicionando características a ele. Já a *backward* consiste em usar um conjunto completo e ir removendo características

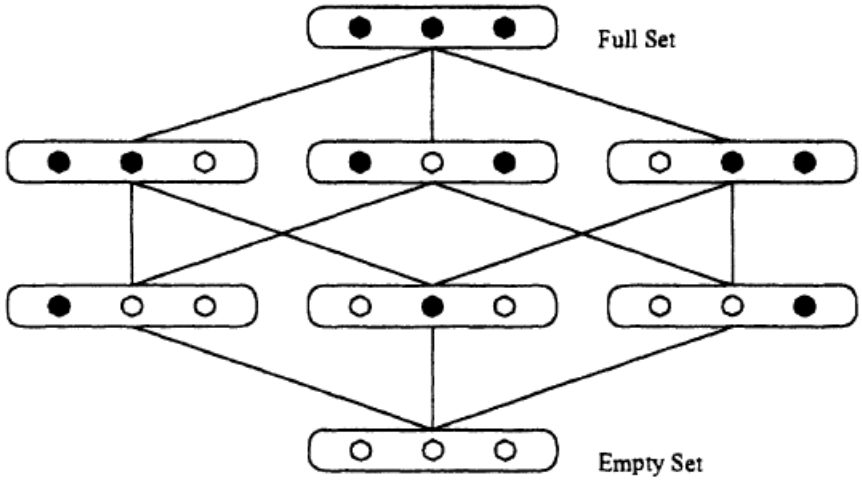

Figura 2 – Busca por remoção de características (SBG). (LIU; MOTODA., 1998)

sucessivamente. A terceira forma consiste em adicionar e remover simultaneamente, ela é composta por duas buscas simultaneas, em que o objetivo é encontrar o melhor subconjunto. Os críterios de parada são: quando o melhor subconjunto é encontrado, ou quando ambas chegam ao meio do subconjunto. A ultima forma consiste em adicionar e remover características de maneira arbitrária, essa forma busca evitar *local optima* (YU, 2005).

Um dos problemas da busca é a sua estratégia de busca. Comparar todas as características entre si e todos os conjuntos gerados seria um problema da ordem de $O(2^N)$, mesmo com um N pequeno esse problema seria muito exaustivo para processá-lo. Para contornar esse problema existem três maneiras de realizar a busca: busca completa, busca sequencial e busca randomica.

Busca completa - se baseia em analisar todos os subconjuntos possíveis, essa busca pode ser exaustiva ou não-exaustiva. A exaustiva buscará analisar todas as combinações, sendo assim, ela se torna um problema da ordem $O(2^N)$. A não-exaustiva busca analisar o máximo de estados possiveis dentro de um determinado limite. (YU, 2005).

Busca sequencial - se baseia em remover ou adicionar características conforme é executado. Esse tipo de busca pode acarretar em perda de alguns subconjuntos ótimos, porém é mais rapido, sendo um problema da ordem de $O(N^2)$. (DASH, 1997)

Busca randomica - ela segue duas frentes. Uma é utilizar busca sequencial e adicionar as características de forma randomica. A outra frente é gerar o próximo subconjunto de maneira completamente randomica, ou seja, gerar todo o subconjunto de características aleatórias. Tende a ser um problema da ordem de $O(N^2)$ (YU, 2005).

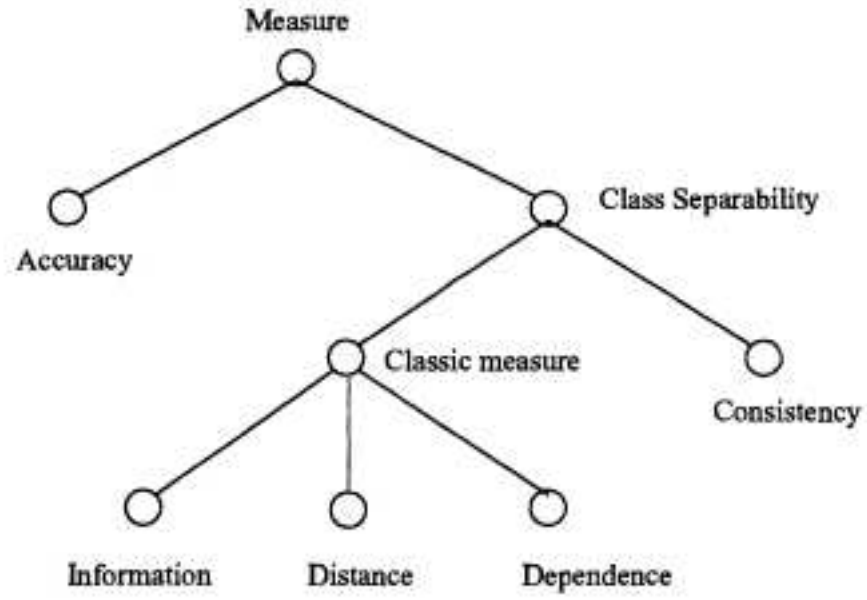

Figura 3 – Hierarquia de Medidas. (LIU; MOTODA., 1998)

2.4 Avaliação do Subconjunto

Para poder dizer quão bom é um subconjunto é preciso atentar-se a algum critério. Esses critérios são descritos de acordo com a hierarquia aprsentada na figura 3. As medidas para avaliar uma caracaterística são: medidas de informação (*information measures*), medidas de distancia (*distance measures*), medidas de dependencia (*dependancy measures*), medidas de consistência (*consistency measures*) e medidas de acurácia (acuracy measures). (YU, 2005; LIU; MOTODA., 1998) As medidas podem ser divididas em três grupos distintos: a) grupo de medidas classicas, que englobam as medidas de informação, medidas de distancia e medidas de dependencia b) grupo de medidas de consistência, que engloba a própria medida em si c) grupo de medidas de acurácia, que também engloba apenas a medida em si (LIU; MOTODA., 1998).

2.4.1 Medidas de informação

A medida de informação determina a informação obtida por uma dada característica (YU, 2005). Isso implica em dizer que uma característica A1 oferece mais informações do que uma característica A2, se dado uma função de incerteza F(x), A1 reduza o grau de incerteza mais do que A2. (colocar uma função de incerteza para ilustrar?)

2.4.2 Medidas de distancia

Também conhecido como medida de separabilidade, medida de divergencia ou medida discriminativa, essa medida é feita através do calculo da distancia entre duas

classes, ou seja, calcular o quanto uma característica consegue separar duas classes de maneira distinta. Sendo F(X) o calculo da distancia, uma característica A1 separar melhor duas classes do que A2 se F(A1) > F(A2). (LIU; MOTODA., 1998)

2.4.3 Medidas de dependencia

Conhecida também como medida de correlação ou de similaridade, essa medida visa medir a habilidade de poder predizer o valor de um variável X sabendo o valor de uma variável Y, sendo assim, dado uma função para medir a dependencia F(X, Y), sendo A1 e A2 características distintas, e C a classe, a característica A1 é preferida à característica A2, se F(A1, C) > F(A2, C), ou seja, a característica A1 está melhor associada com a classe C do que A2. (LIU; MOTODA., 1998; YU, 2005)

2.4.4 Medidas de consistência

As medidas anteriores tem o propósito de achar as melhores características que consigam destinguir uma classe de outra. Porém elas não conseguem distinguir características que sejam igualmente boas, ou seja, que apontem para um mesmo resultado e gere redundancia nos subconjuntos. A medida de consistência procura achar um menor número de características em um dado subconjunto que seja tão consistente em seprar as classes quanto ao conjunto inicial. Ou seja, dado uma função de consistência F(X, Y) em relação a classe C, o subconjunto A1 será consistente se F(A1, C) = F(S, C), sendo S o conjunto inicial (YU, 2005)

2.4.5 Medidas de acurácia

Essa medida já é utilizada dentro do classificador, ou seja, não é aplicada ao modelo de filtro, apenas no modelo de envelopamento (*wrapper*) e no híbrido. Para um dado subconjunto de características, quanto mais alta for a acurácia desse conjunto em relação a sua predição, melhor será esse subconjunto (LIU; MOTODA., 1998).

2.5 Critério de Parada

O critério de parada serve para que, dado um conjunto de características e um algoritmo de seleção de características, ao alcançar um determinado estado de seleção ele diga se o subconjunto gerado está bom ou se ainda precisa ser refinado (DASH, 1997). Critérios de parada comumente usados são: a) a busca é finalizada; b) algum valor pré-determinado é alcançado, como por exemplo um número minimo ou máximo de características; c) a adição ou remoçao de características não gera um subconjunto melhor; d) um subconjunto satisfatóriamente bom é encontrado.

Os critérios de parada são comumente utilizados para que não seja necessário realizar toda a busca, uma vez que o trabalho de exaustão da busca demanda muito tempo e recurso.

2.6 Validação de resultados

Para se avaliar os resultado, primeiramente deve-se ter algum conhecimento sobre os dados que foram submetidos para a seleção de características. Existem vários cálculos que podem ser feitos para que seja posível validar que o processo de seleção realmente resultou em um subconjunto com características que resolva o problema de uma maneira melhor do que o conjunto de características iniciais. Podemos utilizar três dimensões para poder verificar se um método de seleção realmente foi efetivo, são eles: acurácia de predição, complexidade das representações e o número de características selecionadas (LIU; MOTODA., 1998). Além dessas dimensões, devemos levar em consideração alguns fatos dos próprios algorítimos em si, como a velocidade de um método de seleção e a generalidade das características selecionadas. Essas dimensões e considerações serão melhor explicadas na seção Validar Resultados.

3 Modelos de seleção de características

3.1 Escolha de modelos

Como visto, existem várias maneiras e técnicas para se montar um modelo para seleção de características, podendo-se combinar os modelos de algorítimos (*filter*, *wrapper* e *hybrid*), as formas de busca e as várias maneiras de avaliar os subconjuntos gerados para chegar a um modelo que seja adequado ao problema, ou seja, encontrando um subconjunto que sejá otimizado em relação ao conjunto inicial.

Para poder melhor entender o funcionamento dos modelos, deve-se primeiro entender como os algorítimos funcionam.

3.1.1 Modelo de Filtro (*filter*)

Os algoritmos de filtro não necessitam de um algoritmo de *machine learning* para poder fazer a seleção de características, geralmente necessitando de um menor poder computacional para poder serem realizados. A imagem 5 ilustra como trabalha um algoritmo de filtro.

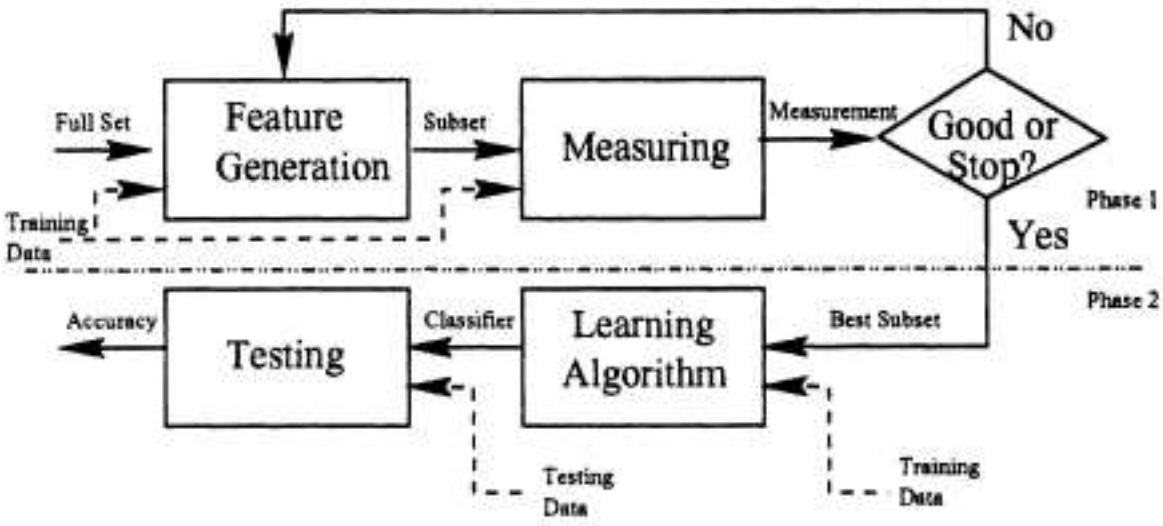

Figura 4 – Fluxograma do modelo de filtro (*filter*). (LIU; MOTODA., 1998)

O modelo é composto por duas fases, a primeira fase consiste em seleção de características utilizando as medidas descritas no capítulo anterior, a segunda fase consiste em utilizar o subconjunto gerado em um classificador. Nessa fase também é colocado os dados de treino no classificador para avaliar se o resultado foi satisfatório. Os algoritmos

de filtro não dependem da avaliação do classificador, e sim das informações e medidas obtidas diretamente dos dados. (LIU; MOTODA., 1998)

Pode-se generalizar o modelo de filtro de acordo com a imagem 6, onde para um dado conjunto D de características, é escolhido um subconjunto $S0$ através de alguma das formas de busca. Cada conjunto gerado é avaliado e comparado com o seu anterior, sempre sendo escolhido aquele que tenha um melhor valor em seu critério de avaliação (γ). Esse processo será repetido até que um critério de parada (α) seja alcançado ou que sejam testados todos os conjuntos possíveis. (YU, 2005)

Filter Algorithm
input: $D(F_0, F_1, ..., F_{n-1})$ // a training data set with N features
S_0 // a subset from which to start the search
δ // a stopping criterion
output: S_{best} // an optimal subset

```
01 begin
02     initialize: S_best = S_0;
03     γ_best = eval(S_0, D, M); // evaluate S_0 by an independent measure M
04     do begin
05         S = generate(D); // generate a subset for evaluation
06         γ = eval(S, D, M); // evaluate the current subset S by M
07         if (γ is better than γ_best)
08             γ_best = γ;
09             S_best = S;
10     end until (δ is reached);
11     return S_best;
12 end;
```

Figura 5 – Generalização do algoritmo de filtro (*filter*). (YU, 2005)

3.1.2 Modelo de Envelopamento (*wrapper*)

Os algoritmos que se baseiam no modelo de envelopamento necessitam de um algoritmo de *machine learning* pré determinado para que seja possível utiliza-lo. Isso se deve ao fato de que o modelo usa o próprio classificador para avaliar o quão bom é o subconjunto de dados gerado. A imagem 7 ilustra como funciona um algoritmo de envelopamento.

O modelo é composto por duas fases, a primeira fase consiste em escolher um subconjunto utilizando um classificador para poder avaliar o quão bom é esse modelo. Já a segunda fase é igual a do modelo de filtro, onde o subconjunto é utilizado no classificador e então são inseridos os dados de treino para verificar se o resultado foi satisfatório.

Pode-se generalizar o modelo de envelopamento de acordo com a imagem 8, onde para um dado conjunto D de características, é escolhido um subconjunto $S0$ através de

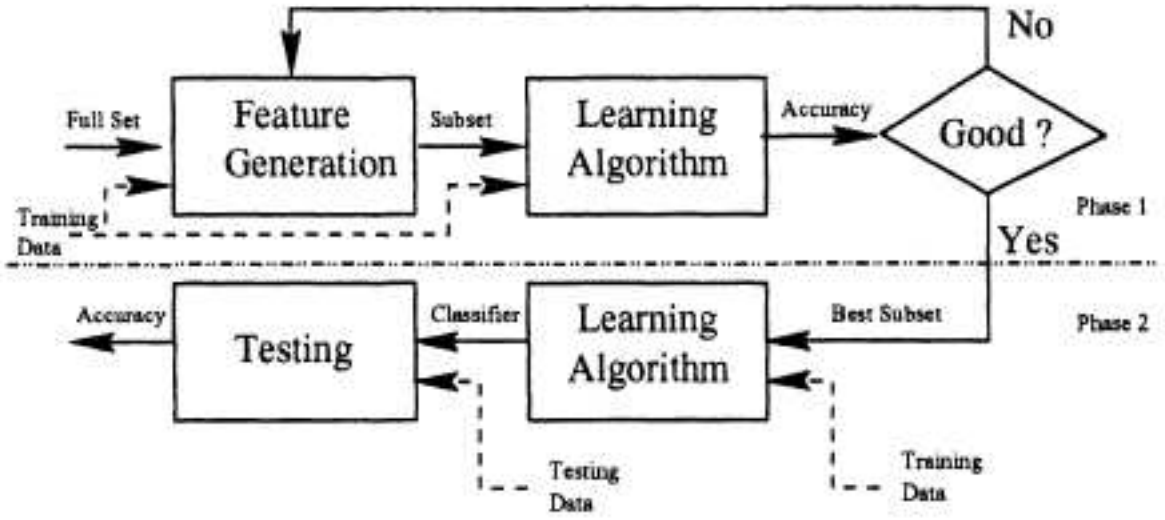

Figura 6 – Fluxograma do modelo de envelopamento (*wrapper*). (LIU; MOTODA., 1998)

alguma das formas de busca. Cada conjunto gerado é avaliado e comparado com o seu anterior utilizando um algoritmo de *machine learning*, sempre sendo escolhido aquele que tenha um melhor valor em seu critério de avaliação (γ). Esse processo será repetido até que um critério de parada (α) seja alcançado ou que sejam testados todos os conjuntos possíveis. (YU, 2005)

Wrapper Algorithm

input: $D(F_0, F_1, ..., F_{n-1})$ // a training data set with N features
S_0 // a subset from which to start the search
δ // a stopping criterion

output: S_{best} // an optimal subset

```
01 begin
02     initialize: S_best = S_0;
03     γ_best = eval(S_0, D, A); // evaluate S_0 by a mining algorithm A
04     do begin
05         S = generate(D); // generate a subset for evaluation
06         γ = eval(S, D, A); // evaluate the current subset S by A
07         if (γ is better than γ_best)
08             γ_best = γ;
09             S_best = S;
10     end until (δ is reached);
11     return S_best;
12 end;
```

Figura 7 – Generalização do algoritmo de envelopamento (*wrapper*). (YU, 2005)

3.1.3 Modelo Híbrido (*hybrid*)

Os algoritmos que se baseiam no modelo híbrido são compostos por técnicas do modelo de filtro e de envelopamento. O modelo híbrido busca utilizar da melhor forma possível os dois modelos para poder alcançar um melhor subconjunto. Geralmente utiliza-se o modelo híbrido para problemas como muitos dados (YU, 2005). Na figura 9 veremos como seu algorítimo funciona.

Hybrid Algorithm
input: $D(F_0, F_1, ..., F_{n-1})$ // a training data set with N features
S_0 // a subset from which to start the search
output: S_{best} // an optimal subset

```
01 begin
02     initialize: S_best = S_0;
03     c_0 = card(S_0); // calculate the cardinality of S_0
04     γ_best = eval(S_0, D, M); // evaluate S_0 by an independent measure M
05     θ_best = eval(S_0, D, A); // evaluate S_0 by a mining algorithm A
06     for c = c_0 + 1 to N begin
07         for i = 0 to N − c begin
08             S = S_best ∪ {F_j}; // generate a subset with cardinality c for evaluation
09             γ = eval(S, D, M); // evaluate the current subset S by M
10             if (γ is better than γ_best)
11                 γ_best = γ;
12                 S'_best = S;
13         end;
14         θ = eval(S'_best, D, A); // evaluate S'_best by A
15         if (θ is better than θ_best);
16             S_best = S'_best;
17             θ_best = θ;
18         else;
19             break and return S_best;
20     end;
21     return S_best;
22 end;
```

Figura 8 – Generalização do algoritmo híbrido (*hybrid*). (YU, 2005)

Para um dado conjunto D de características, é escolhido um subconjunto *S0* através de alguma das formas de busca, geralmente utiliza-se um conjunto vazio e é adicionado características a cada iteração. A cada iteração se adiciona uma característica e é comparada subconjunto a subconjunto dessa iteração utilizando medidas independentes (γ), ou seja, das características em si, que terá c + 1 características, até que seja possível encontrar o melhor subconjunto. Repete-se o processo até que se ache o melhor subconjunto de cada uma das iterações, e então esses subconjunto serão então comparados utilizando algum classificador (θ), para que seja obtido o melhor subconjunto possível. (YU, 2005)

3.2 Algoritmos Escolhidos

3.2.1 *Relief-F* - Método de Filtro

O algoritimo *Relief-F* utiliza métodos estatísticos para selecionar as características relevantes, é um método que se baseia nos pesos atribuídos às caracterísicas (DASH, 1997). O algorítmo realiza a busca de maneira sequêncial e utiliza medidas de distância para poder realizar a avaliação entre as características. Primeiramente o algorítmo inicializa os pesos (W) com 0, e escolhe um valor de limite (ϕ), então para uma característica n ele calcula os valores de *Near Hit* e *Near Miss*, ambos calculados através da distância euclidiana. *Near Hit* é a menor distância entre instancias de mesma classe, já o *Near Miss* é o menor valor entre instancias de classes diferentes. Depois de percorrer todas as características ele retorna aquelas que têm o valor do peso maior do que o limite estabelecido. O algoritmo original, *Relief* funciona apenas para classes binárias, o *Relief-F* contorna esse problema.

3.2.2 Método da Árvore de Decisão - *Decision Tree Method(DTM)* - Método de Filtro

O algorítmo de Árvore de Decisão (DTM) utiliza do cáculo de medidas de informação utilizando um algoritmo, nesse caso em específico será utilizado o C4.5. O algoritmo C4.5 é aplicado no conjunto de treinamento e as características que aparecerem na árvore "podada"serão selecionadas.

3.2.3 *Linear Forward Selection (LFS)* - Método de Envelopamento

O LFS começa com um conjunto vazio e a cada iteração ele adiciona uma característica que é selecionada através de alguma medida de avaliação e então gera um novo subconjunto. É então avaliado o desempenho do classificador naquele dado instante, e isso se repete até que o melhor subconjunto seja encontrado.

3.3 Validação dos Modelos

Os modelos selecionados devem contribuir para um melhor desempenho de um dado modelo de Aprendizado de Máquina, sendo assim, serão testados utilizando uma base de dados e levando em consideração a acurácia do classificador. A base de dados a ser utilizada é dos sobreviventes do titanic. Cada um dos modelos serão testados utilizando essa base seguindo os seguintes critérios:

1. Será utilizado o classificador KNN

Tabela 1 – Características da base Titanic

Característica	**Tipo**
survived	Nominal {0, 1}
pClass	Numérico
sex	Nominal {male, female}
age	Numérico
sibsp	Numérico
parch	Numérico
fare	Numérico
embarked	Nominal {Q,S,C}

Tabela 2 – Informações sobre o classificador e instancias da base Titanic

Classificador	**KNN (Weka.IBk)**
N. de características inicial	8
Acurácia inicial	72.8977%
Instancias para treino	606
Instancias para teste	303

Tabela 3 – Resultados dos modelos - base Titanic

Modelo	**Características Mantidas**	**Acurácia Classificador**
Relief-F	sex, pClass, age	80.5281%
DTM	sex, age, sibsp, pclass, fare	78.5479%
LFS	sex, fare	79.538%

2. Será realizado um pré-processamento na base de dados para:

   Remover características que sejam do tipo String

   Substituir os campos vazios pela mediana daquele campo

3. Da base de treino, será utilizado 2/3 para realizar o treinamento e 1/3 para realizar os testes

As características da base de dados do titanic, após ser realizado o pré processamento, se encontra da seguinte maneira descrito na Tabela 1.

O classificador a ser utilizado, as instancias e a acurácia inicial do classificador se encontram descritos na Tabela 2.

Após cada um dos métodos de seleção ser aplicado à base de dados, alguns resultados foram obtidos e mostrados na Tabela 3:

Os resultados acima são para uma base com poucas características mas ainda sim foi possível ver uma melhora no desempenho do classificador. A base do Titanic é executada em poucos segundos, sendo impossível verificar o fato de que a diminuição de características leva a um melhor desempenho em quesito computacional. Para poder melhor exemplificar esse atributo será utilizado uma base maior. MADELON é uma base

Tabela 4 – Informações sobre o classificador e instancias da base MADELON

Classificador	**KNN (Weka.IBk)**
Base de dados	MADELON
N. de características inicial	500
Acurácia inicial	52.0194%
Tempo para treinamento	12 segundos
Instancias para treino	1227
Instancias para teste	613

Tabela 5 – Resultados dos modelos - base MADELON

Modelo	**Características Mantidas**	**Acurácia Classificador**	**Tempo de treinamento**	**Tempo para Selecionar características**
Relief-F	a_28, a_48, a_64, a_105, a_128, a_153, a_241, a_281, a_318, a_336, a_338, a_378, a_433, a_442, a_451, a_453, a_455, a_472, a_475, a_493.	88.6914%	<1 segundo	37 segundos
DTM	a_105, a_442, a_338, a_475, a_451, a_128, a_28, a_336, a_119, a_493, a_48, a_251, a_292, a_318, a_433	80.937%	<1 segundo	2 segundos
LFS	a_105, a_128, a_241, a_336, a_338, a_442	91.2601%	<1 segundo.	332 segundos

de dados artificial que foi parte de um desafio chamado NIPS, realizado em 2003. É um problema de duas classes que tem como característica ser altamente não linear. A base de dados consta com 1820 instancias e 500 características, alem da classe. Mantendo o planejamento inicial, que é de utilizar 1/3 da base para realizar os testes, temos o status inicial do classificador da seguinte maneira:

Após executado cada um dos modelos de seleção em cima da base MADELON, foi possível alcançar os resultados mostrados na Tabela 5. Como é possível observar, o número de características selecionado foi menor do que 10% do número inicial e o desempenho do classificador foi aumentado em até 40%. O tempo gasto para realizar seu treinamento foi diminuído drásticamente, porém o tempo gasto para realizar o processo de seleção foi maior do que o tempo economizado, porém isso compensa se levar em consideração o desempenho ganho com o tempo gasto realizando a seleção de caraterísticas.

4 Validação dos Resultados

5 Considerações Finais

Referências

BRINK, H.; RICHARDS, J. *Real World Machine Learning*. [S.l.]: Manning Publications C.O, 2014. Citado 2 vezes nas páginas 23 e 27.

BUSINESSWEEK, B. *The Current State of Business Analytics: Where Do We Go from Here?* 2011. Disponível em: <http://www.sas.com/resources/asset/busanalyticsstudy_wp_08232011.pdf>. Citado na página 23.

DASH, H. L. M. Feature selection for classification. p. 131–156, 1997. Citado 3 vezes nas páginas 29, 31 e 37.

GUYON, I. An introduction to variable and feature selection. p. 1157–1182, 2003. Citado na página 23.

LIU, H.; MOTODA., H. *Feature selection for knowledge discovery and data mining*. [S.l.]: SPRINGER SCIENCE+BUSINESS MEDIA, LLC, 1998. ISBN 978-1-4615-5689-3. Citado 9 vezes nas páginas 13, 28, 29, 30, 31, 32, 33, 34 e 35.

MITCHELL, T. *Machine Learning*. New York, NY: McGraw-Hill, 1997. ISBN 0-07-042807-7. Citado 2 vezes nas páginas 23 e 27.

MOLINA, L. C. Feature selection algorithms: A survey and experimental evaluation. v. 17, p. 306–313, 2002. Citado na página 27.

OBSERVATORY, B. I. Big data: Artificial intelligence. Netherlands, p. 1 – 15, 2013. Citado na página 23.

YU, H. L. e L. Toward integrating feature selection algorithms for classification and clustering. v. 17, p. 491–502, 2005. ISSN 1041-4347. Citado 9 vezes nas páginas 13, 27, 28, 29, 30, 31, 34, 35 e 36.