UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
DEPARTAMENTO DE CIÊNCIA DE COMPUTAÇÃO

André Luiz Jacuá Dias Tschaffon
Rafael de Araújo Martinho Pinheiro

UTILIZAÇÃO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA DETECÇÃO DE REGIÕES CANCERÍGENAS EM IMAGENS DE RAIOS-X

Niterói
2009

André Luiz Jacuá Dias Tschaffon

Rafael Martinho Pinheiro

UTILIZAÇÃO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA DETECÇÃO DE REGIÕES CANCERÍGENAS EM IMAGENS DE RAIOS-X

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Bianca Zadrozny

Niterói

2009

André Luiz Jacuá Dias Tschaffon

Rafael Martinho Pinheiro

UTILIZAÇÃO DE TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA DETECÇÃO DE REGIÕES CANCERÍGENAS EM IMAGENS DE RAIOS-X

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em Fevereiro de 2010

BANCA EXAMINADORA

_____

Bianca Zadrozny, Ph.D.
Orientadora
UFF

_____

Ana Cristina Bicharra Garcia, Ph.D.
UFF

_____

Flávia Cristina Bernardini, D.Sc.
UFF

Niterói

2009

RESUMO

O câncer de mama é o segundo tipo de câncer mais mortal entre as mulheres, e sua rápida e eficaz identificação deve ocorrer para que os riscos à paciente sejam reduzidos. A identificação de áreas cancerígenas em mamogramas é, sabidamente, de grande dificuldade para o olho humano, mesmo para um médico experiente. Portanto, a tentativa de utilização de técnicas de inteligência artificial, como a aprendizagem supervisionada, tem um grande potencial de auxílio, podendo minimizar os riscos de erro neste procedimento.

No ano de 2008, a ACM *Special Interest Group on Knowledge Discovery and Data Mining* organizou o KDD-CUP, campeonato anual de mineração de dados, tendo como tema o câncer de mama e tornou pública sua base de dados composta de diversas informações referentes a imagens de pacientes. Cada imagem contém regiões candidatas a possuírem tal doença. Uma característica importante dessa base é o desbalanceamento dos dados, isto é, o número de imagens com regiões cancerígenas é muito menor que o número de imagens sem regiões cancerígenas (numa proporção de x para y).

Neste trabalho, utilizou-se a base de dados disponibilizada para a KDD-CUP e o software livre WEKA para testar a eficácia da técnica de aprendizagem Costing em combinação com três conhecidos algoritmos de mineração de dados: árvore de decisão, rede bayesiana e *support vector machine* (SVM). Com a utilização da técnica Costing, é possível atribuir custos diferentes a cada tipo de erro (falsos positivos ou falsos negativos) e assim tentar reduzir os efeitos do desbalanceamento de dados em nossos classificadores.

Os resultados obtidos neste projeto comprovam que a utilização do Costing otimiza a acurácia das classificações geradas pelos algoritmos de aprendizagem automático. Através da medida AUC, verificou-se que as respostas fornecidas pelos algoritmos, em combinação com o Costing, tendem a ser mais corretas do que as

respostas fornecidas pelos mesmos algoritmos sem a utilização de informações de custo.

Para validar os resultados obtidos, tomou-se como base para comparação os resultados obtidos pela equipe vencedora [5] da competição KDD-CUP 2008. Dessa forma, pode-se verificar que com a utilização da técnica Costing é possível atingir resultados próximos aos encontrados pelos vencedores.

Palavras Chave:
Costing, Aprendizagem de Máquina, Câncer de Mama

ABSTRACT

Breast cancer is the second most deadly type of cancer among women, and its rapid and effective identification must occur so that the risks to the patient are reduced. The identification of cancerous areas on mammograms is known to be of great difficulty for the human eye, even for an experienced one. Therefore, the attempt to use artificial intelligence techniques such as supervised learning, has a great potential to help, which may minimize the risk of error in this procedure.

In 2008, the ACM Special Interest Group on Knowledge Discovery and Data Mining organized the KDD-CUP, the annual championship of data mining, on the theme of breast cancer and has made public its database composed of various information relating to images patients. Each image contains regions that are candidates for having such a disease. An important feature of this base is the unbalance of the data, *ie* the number of images with regions of cancer is much less than the number of images without regions of cancer (a ratio of x to y).

In this study, we used the database available for the KDD-CUP and a free software called WEKA to test the effectiveness of Costing technique in combination with three known data mining algorithms: decision tree, Bayesian network and *support vector machine* (SVM). With this Costing technique you can assign different costs for each type of error (false positives or false negatives) and thus try to reduce the effects of imbalance in the data mining classifiers.

The results of this project show that the use of the Costing optimizes the accuracy of the ratings generated by the automatic learning algorithms. By the AUC measure, it was found that the answers given by the algorithms in combination with the Costing tend to be more accurate than the answers provided by these algorithms without the use of cost information.

To validate the obtained results, was taken as a basis for comparing the results obtained by the team that won [5] the competition KDD-CUP 2008. Thus, we can verify that with this Costing technique is possible to achieve similar results to those found by the winners.

Keywords:
Costing, Machine Learning, Breast Cancer

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

CAPÍTULO 1 – INTRODUÇÃO

O câncer de mama é uma doença global e o número de pessoas que sofrem desta doença vem crescendo a cada dia. Caracteriza-se pela presença de um nódulo ou tumor no seio podendo ser acompanhado ou não de dor no local. A detecção precoce é essencial para evitar graves problemas à paciente, e as formas utilizadas para tentar detectá-la são o toque da mama e imagens de raios-X desta região. As imagens de raios-X apresentam tamanhos e resoluções que podem dificultar a tentativa de detecção feita pelo especialista e é nesta área em que deve-se realizar todo o esforço necessário para a detecção da doença o quanto antes.

Para auxiliar a prevenção do câncer de mama, sistemas CADs (Computer Aided Design) vem sendo desenvolvidos, muitas vezes utilizando técnicas de aprendizagem de máquina que tentam classificar uma determinada paciente como positivo (possui a doença) ou negativo (não possui a doença). A partir destas classificações, os médicos podem fazer diagnósticos mais precisos e, a partir daí, tomar as medidas necessárias para a cura da doença [1].

Diversas pesquisas vêm sendo realizadas com o objetivo de desenvolver um sistema CAD que apresente a classificação mais precisa possível. O desenvolvimento destes sistemas normalmente é realizado em três etapas, sendo que cada uma delas apresenta determinadas características peculiares.

A primeira etapa consiste no processamento de imagens. Esta etapa é muito importante, pois é a responsável por extrair todas as informações relevantes das imagens. Esta também é bastante complexa quando existem imagens com formatos e tamanhos diferentes, dificultando a extração das características relevantes. O resultado desta etapa é a criação de uma base de dados que serve como entrada para a próxima etapa.

Já a segunda etapa consiste na escolha ou desenvolvimento de uma técnica de aprendizagem de máquina e adaptações necessárias para encontrar uma classificação o mais precisa possível. Técnicas de classificação padrão como Naive Bayes e SVM são normalmente utilizadas, porém muitas vezes é necessária a aplicação de estratégias para de pré-processamento da base de dados de treinamento, antes de construir o modelo de classificação que será utilizado para classificar novas instâncias.

Nos casos em que existe um grande desbalanceamento da base de treinamento, os classificadores costumam encontrar dificuldades em classificar corretamente os exemplos da classe minoritária. Assim, é necessária a utilização de alguma estratégia que minimize os efeitos deste problema na aprendizagem do classificador.

Em bases desbalanceadas, os classificadores, quando combinados com alguma técnica de redução deste desbalanceamento, tendem a obter melhor acurácia. Uma destas técnicas é o Costing, onde são atribuídos custos a cada exemplo da base de treinamento, e com isto a base é reduzida, eliminando aleatoriamente elementos da mesma baseando-se em seu custo.

A terceira etapa é a avaliação dos resultados para verificar a eficiência do sistema CAD desenvolvido. Com os dados resultantes do classificador podemos avaliar o quanto a técnica utilizada na segunda etapa foi eficaz, através da utilização de medidas de avaliação como a acurácia e curvas FROC.

Neste trabalho, utilizamos a base de dados disponibilizada no site da KDD-CUP 2008. A partir desta base, foi realizada a divisão da base de dados, onde a primeira parte foi utilizada na geração do modelo do classificador e a segunda foi utilizada como entrada para este classificador para gerar as saídas.

Como a ampla maioria dos exemplos desta base é rotulada como não cancerígenos, utilizou-se a técnica Costing para reduzir os efeitos causados pelo desbalanceamento. Esta técnica foi utilizada em conjunto com três métodos de classificação: J-48, Naive Bayes e SVM. Os resultados obtidos foram avaliados e comparados com os resultados obtidos pelos vencedores [5] do KDD-CUP 2008 a fim

de verificar a eficácia do Costing.

Este trabalho está organizado da seguinte forma. O Capítulo 2 apresenta o problema de detecção do câncer de mama. O Capítulo 3 faz uma revisão sobre as técnicas de aprendizagem de máquina utilizadas no estudo. O Capítulo 4 descreve a metodologia empregada. O Capítulo 5 discute os resultados obtidos. Por fim, o capítulo 6 apresenta as conclusões.

CAPÍTULO 2 – TÉCNICAS DE DETECÇÃO DO CÂNCER DE MAMA

O câncer de mama, que atinge milhares de mulheres em todo o mundo, é geralmente detectado de duas formas. Uma primeira maneira de detecção seria usando o método de palpação da mama, sendo esta realizada pela própria paciente ou pelo médico. A palpação consiste em palpar toda a região da mama, a região da axila e a parte superior do tronco em busca de algum nódulo ou alteração da pele, como retração ou endurecimento, ou de alguma alteração no mamilo. A segunda maneira é mais eficiente em relação à primeira e é realizada fazendo um mamograma**,** o qual consiste em um raio-X das mamas e das porções das axilas mais próximas das mesmas. Nesse exame**,** o radiologista procura imagens sugestivas de alterações do tecido mamário e dos gânglios da axila. A ecografia das mamas pode auxiliar o radiologista a definir que tipos de alterações são estas [4].

Estes dois métodos de detecção de câncer de mama citados acima têm suas vantagens e desvantagens. Porém**,** os dois têm um problema principal que é a imprecisão na detecção, a qual depende muito do fator humano para que seja feito o diagnóstico e um erro ou até mesmo um atraso pode causar sérios danos à paciente, pois quanto antes for detectado, maiores são as chances de cura, evitando assim que uma possível anomalia aumente de tamanho, agravando o quadro clínico da paciente em observação.

O primeiro método de detecção tem sua eficácia conhecidamente inferior, pois um pequeno nódulo pode passar despercebido e**,** caso seja um princípio de câncer, o mesmo não será detectado precocemente, sendo somente detectado em um estágio mais avançado, onde os riscos aumentam consideravelmente.

Já o método que utiliza imagens de mamogramas vem recebendo atenção especial em pesquisas, com a intenção de auxiliar no diagnóstico precoce da existência de um possível nódulo. O problema principal desta detecção é que pequeno nódulos podem passar despercebidos aos olhos do médico, mesmo sendo este um especialista e tendo um olhar muito bem treinado, podendo os nódulos serem confundidos com pequenas partes de músculos. Assim, a paciente pode realmente ter um nódulo mamário, mas este ser diagnosticado de forma equivocada, sendo assim uma vítima de uma falha humana, provocada pela limitação do médico em diferenciar regiões da imagem. Para minimizar as chances de erro, é comum que o diagnóstico seja realizado por mais de um especialista, porém com isto aumenta-se o custo e também o tempo necessário para a conclusão do exame.

Tendo em vista os problemas apresentados acima, fica claro que o uso de técnicas computacionais poderia ser de grande ajuda aos especialistas na realização de um diagnóstico com maior grau de precisão e consideravelmente mais rápido. As pesquisas nesta área consistem na automatização da detecção, usando como objeto de estudo as imagens de mamogramas. Em todos os estudos realizados, a preocupação maior é auxiliar o médico com um diagnóstico, ficando claro que este em nenhuma hipótese deve ser substituído. Para isso**,** são utilizadas várias técnicas de aprendizagem automática baseadas em arquivos contendo diversas informações relativas às imagens das mamas, para assim chegar a uma conclusão sobre a condição da paciente.

Para a utilização de aprendizagem automática, é necessária a geração de uma base de dados de treinamento contendo informações referentes às imagens de cada paciente. Cada imagem é usualmente dividida em pequenas áreas denominadas regiões de interesse (ROIs). Diversas técnicas [15] de segmentação são utilizadas nesta etapa para encontrar as ROIs, que tem a maior probabilidade de serem reconhecidas pelos classificadores quando forem utilizadas em seu treinamento.

Cada uma destas regiões recebe a avaliação de um especialista que indica se esta é ou não cancerígena. No fim, o resultado é uma base de dados que possui as imagens de raios-X divididas em regiões, sendo que cada uma destas regiões tem características extraídas de sua imagem, como tamanho, posição x,y na imagem e formato de um possível nódulo.

Após a criação de uma base de dados com um grande número de imagens contendo tanto informações de regiões cancerígenas quanto sãs, este é dividido e a escolha de um algoritmo de classificação, este é executado e predições são geradas. Os rótulos gerados em cada predição são comparados com os rótulos reais de cada uma das regiões para averiguar o desempenho do classificador utilizado.

Após as predições do classificador, normalmente ocorre a análise dos seguintes atributos: acurácia, sensibilidade, especificidade, os valores positivos e negativos preditos [3]. Utilizando todos estes atributos, podemos avaliar se o método de classificação utilizado está atingindo todos os seus objetivos previamente estabelecidos.

CAPÍTULO 3 – TÉCNICAS DE APRENDIZAGEM DE MÁQUINA

A aprendizagem de máquina [6], ou aprendizagem automática, é uma área da Inteligência Artificial onde são estudados algoritmos que extraem padrões de exemplos de uma base de treinamento para aprender automaticamente a partir destes exemplos. Estes algoritmos são normalmente utilizados quando se tem muitas informações e é muito difícil programar um computador que seja capaz de se comportar como esperado. Como servem para extrair informações implícitas nos exemplos disponíveis, são considerados algoritmos de mineração de dados.

Existem diversos tipos de tarefas de aprendizado de máquina. Na classificação binária, o modelo gerado pelo algoritmo deve aprender a separar duas classes, dadas amostras de exemplos de cada uma das classes. Já na classificação multi-classe, o classificador deve aprender a separar três ou mais classes. Na regressão, o objetivo é prever o valor de uma função, dados exemplos de entrada e saída da função. Temos também a clusterização, onde se agrupa os exemplos com características semelhantes em aglomerados (*clusters*). Por fim, na aprendizagem por reforço, o objetivo é aprender uma sequência de ações ótimas, dados exemplos de seqüências de ações e reforços referentes a cada ação.

O foco deste trabalho é a utilização de algoritmos de classificação binária para o problema de detecção do câncer de mama. Na classificação binária, dados exemplos:

$$(x,y) \rightarrow \begin{cases} \boldsymbol{x}: vetor\ de\ atributos \\ \boldsymbol{y}: classe\ (valor\ discreto \end{cases}$$

derivados de uma distribuição ***D*** com domínio *X* x *Y,* o objetivo é encontrar um classificador

$$\boldsymbol{h: X \rightarrow Y}$$

que minimize a taxa de erro esperada:

$$E_{x,y\sim D}[I(h(x) \neq y)]$$

onde x ∈ X, e y ∈ Y.

Assim temos:

Exemplos (x,y) ⟶ *Algoritmo de Aprendizado* ⟶ $\boldsymbol{h: X \rightarrow Y}$

Para realizar os experimentos computacionais, foram escolhidos três dentre os diferentes tipos de algoritmos de aprendizagem automática implementados no aplicativo WEKA. Esta seção apresenta uma breve descrição de cada algoritmo utilizado.

3.1 ÁRVORE DE DECISÃO

Uma árvore de decisão [7] é uma ferramenta de suporte a tomadas de decisão que utiliza um modelo de decisão e suas possíveis conseqüências. É comumente usado para selecionar a melhor estratégia para se atingir um objetivo e para calcular probabilidades condicionais. Utiliza a técnica de divisão e conquista para aprendizagem automática a partir de um conjunto de dados.

A seguir, na Figura 1, temos um exemplo de uma árvore de decisão simples:

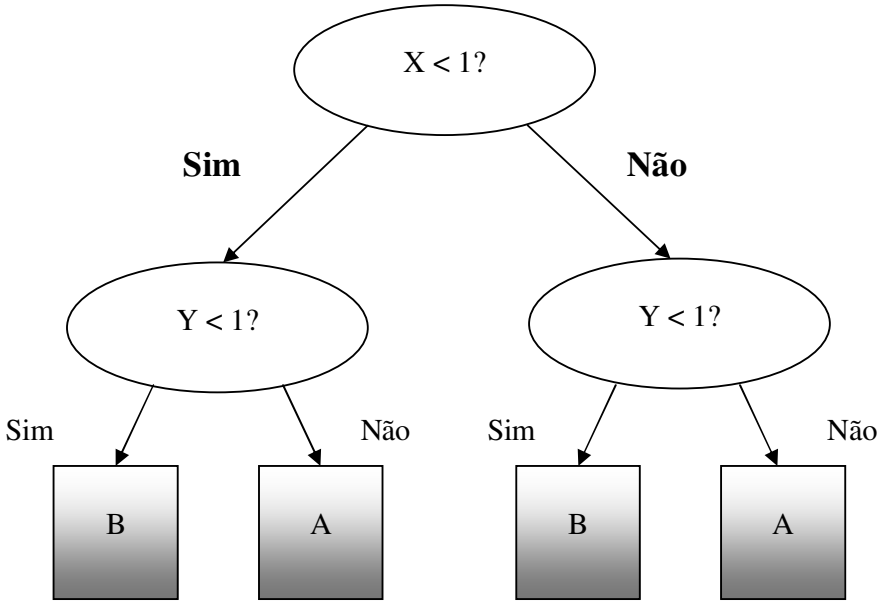

Figura 1: Árvore de Decisão

Cada nó da árvore testa um atributo que este é usualmente comparado com um número constante, embora possam haver implementações que utilizem funções com um ou mais atributos ou mesmo façam comparações entre eles. Caso este atributo seja numérico, verifica-se se o valor do mesmo é maior ou menor que a determinada constante e dependendo do resultado escolhe-se um entre os dos possíveis caminhos a serem seguidos. Já as folhas das árvores indicam a melhor classificação, um conjunto de classificações ou ainda a distribuição probabilística de todas as possíveis classificações para as instâncias que a atingirem.

Se um atributo a ser testado em um nó é nominal, então normalmente o número de nós descendentes deste nó é equivalente ao número de possíveis valores para este atributo e o mesmo não é mais testado nos ramos subseqüentes.

Assim, para fazer a predição de uma instância de classificação desconhecida, percorre-se a árvore de decisão verificando em cada nó atingido o valor do atributo relacionado a este. Por fim, a instância será classificada de acordo com a classe relacionada à folha atingida.

Este algoritmo costuma ser utilizado em problemas onde os conceitos e as decisões dos problemas são descritos através de exemplos, sendo estes representados por pares atributo-valor. É muito utilizado com dados de treinos ruidosos[7].

Em termos de implementação, a árvore de decisão segue a filosofia de "dividir para conquistar" [8]. Os nós de decisão internos são univariados, pois usam um único atributo x_i . Se x_i é numérico, então é comparado com uma constante e se é discreto, cada um de seus possíveis valores se torna uma ramificação na árvore.

O *loop* principal da implementação consiste em: A é o melhor atributo de decisão para o próximo nó. Assim, associa-se A como o atributo de decisão do nó. Para cada valor de A, cria-se uma nova ramificação. A seguir, ordena-se os dados de treino para as folhas. Se todos os exemplos já estão classificados, o algoritmo para. Senão, itera sobre as novas folhas.

A métrica da impureza mais comum para problemas de classificação binária é a entropia. Tendo a base de treinamento S, e p_+ e p_- como a proporção de elementos positivos e negativos, respectivamente, em S, a entropia mede a impureza de S da seguinte maneira:

$$\text{Entropia}(S) \equiv - p_+ \log 2\, p_+ - p_- \log 2\, p_-$$

3.2 NAIVE BAYES

Redes Bayesianas [9] podem eficientemente representar distribuições probabilísticas complexas, e tem recebido muita atenção nos últimos anos. Na prática, os métodos para inferência exata em redes bayesianas são normalmente muito custosos, e costuma-se recorrer a métodos aproximados como cadeias de Markov. No entanto, a aplicação deste último algoritmo é de dificuldade mais elevada, pois a convergência é de difícil diagnóstico e em alguns casos sequer ocorre, o tempo de inferência é imprevisível, podendo não ser viável, e algumas vezes ocorrem resultados incorretos como saída. Como resultado, e apesar de muita investigação, inferência em redes

Bayesianas permanece uma grande incógnita e isso limita significativamente a sua aplicabilidade.

Naive Bayes (ou rede bayesiana "inocente") é uma variação especial de rede bayesiana que é amplamente utilizada para classificação e *clustering*, mas o seu potencial para modelagem probabilística permanece largamente inexplorado [9]. Naive Bayes representa uma distribuição como uma mistura de componentes, onde dentro de cada um destes, todas as variáveis são consideradas independentes entre si. Com um número de componentes suficiente, ele pode aproximar uma distribuição arbitrária com um resultado muito próximo do correto. O tempo de inferência é linear no número de componentes e o número de variáveis de consulta.

Uma rede Bayesiana é uma implementação computacional [10,11] de uma distribuição de probabilidade de um conjunto de n variáveis, (X_1,. . . , X_n), como um grafo acíclico direcionado e um conjunto de distribuições de probabilidade condicional (CPDs ou *conditional probability distributions*). Cada nó corresponde a uma variável, e as CPDs associadas a esta variável dão a probabilidade de cada estado da variável dadas todas as combinações possíveis de estados de seus pais. O conjunto de pais de X_i, denotado π_i, é o conjunto de nós com um arco chegando em X_i no gráfico.

A estrutura da rede garante que cada nó é condicionalmente independente dos nós que não são seus descendentes dados seus pais. A distribuição conjunta das variáveis é dada por P (X_1,..., X_n) = $\prod_{i=1}^{n}$ P (X_i | π_i). Para domínios discretos, a forma mais simples de CPD é uma tabela de probabilidade condicional, mas isso requer espaço exponencial no número de pais da variável.

Modelos Naive Bayes são assim chamados por sua "ingênua" suposição que todas as variáveis X_i são independentes entre si, dada uma variável C. Logo, a distribuição conjunta é dada de forma compacta por

$$P(C, X_1, \ldots, X_n) = P(C) \prod_{i=1}^{n} P(Xi \mid C)$$

As distribuições condicionais P (X_i | C) podem assumir qualquer forma, por exemplo, multinomial para as variáveis discretas e Gaussianas para as contínuas.

Quando a variável C é observada nos dados de treinamento, Naive Bayes pode ser usado para a classificação, atribuindo exemplo de teste (X_1,..., X_n) para a classe C que possua P (C | X_1,..., X_n) máximo. Quando C não é observado, os exemplos de dados (X_1,..., X_n) podem ser agrupados mediante a aplicação do algoritmo EM (*Expectation-Maximization*) com C como a informação em falta. Cada valor de C corresponde a um *cluster* diferente, e P (C | X_1,..., X_n) é a probabilidade de adesão do exemplo ao cluster.

Classificadores bayesianos são conhecidos por serem classificadores ótimos, pois minimizam o risco de erros de classificação. No entanto, eles precisam definir P (X | C), ou seja, a probabilidade conjunta dos atributos dado a classe. Estimar esta distribuição de probabilidade em um conjunto de treino é um problema relativamente difícil, pois pode exigir uma base de dados muito grande, mesmo para um número reduzido de atributos, para que possa explorar de forma significativa todas as combinações possíveis.

Por outro lado, no Naive Bayes, os atributos são assumidos como independentes uns dos outros, dada a classe. Assim:

$$P (C \mid X) = \prod_{i=1}^{n} = (P (X_i \mid C\ p)\ C) / P (X)$$

O classificador Naive Bayes é, portanto, totalmente definido pelas probabilidades condicionais de cada determinado atributo dada a classe. A propriedade de independência condicional simplifica o processo de aprendizagem do modelo de dados. Na presença de dados discretos ou gaussianos este processo é simples e direto. Além disso, o classificador é conhecido por ser um método robusto, mostrando um bom desempenho em termos de acurácia. Devido à sua indução rápida, é muitas vezes considerado como um método de referência em estudos de classificação.

3.3 SVM

O SVM [12] é um algoritmo de aprendizado supervisionado que infere, a partir de um conjunto de exemplos, rotulados uma função que recebe novos exemplos como entrada, e produz um conjunto de predições como saída. A saída do algoritmo é uma

função matemática definida no espaço no qual os exemplos são tomados, e que separa todos os pontos desse espaço em dois valores, fazendo uma divisão correspondente aos dois rótulos de classe que são considerados na classificação binária (positivo e negativo).

Tal função de decisão pode ser expressa por uma função matemática que leva um vetor de entrada x ao valor previsto para x (+1 ou -1). O classificador linear pode ser escrito como

$$g(\mathbf{x}) = \text{sinal}(f(\mathbf{x})), \text{ onde } f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

Desta forma, temos uma função parametrizada pelo vetor de pesos w e o escalar b. A notação $\langle \mathbf{w}, \mathbf{x} \rangle$ corresponde ao produto escalar de w e x, definido por

$$\langle \mathbf{w}, \mathbf{x} \rangle = \sum_{i=1}^{d} w_i x_i$$

onde d é o número de atributos, e w_i é o *i-ésimo* elemento de **w,** onde **w** é da forma (w_1, w_2, … w_d). Este algoritmo utiliza métodos de otimização quadrática para encontrar os parâmetros **w** e **b** do classificador, como podemos ver em [13].

Resumidamente, temos o SVM como: dado um conjunto de vetores de treinamento $\mathbf{x}_1$, $\mathbf{x}_2$, … $\mathbf{x}_n$, com rótulos correspondentes $\mathbf{y_1}$, $\mathbf{y_2}$, … $\mathbf{y_n}$ que assumem valores +1 ou -1, escolher os parâmetros **w** e b da função de decisão linear que generalize bem, ou seja, faça predições razoáveis para exemplos desconhecidos.

Este algoritmo é de fácil aplicação e geralmente gera classificadores com baixas taxas de erro. No entanto na prática, normalmente tem problemas em lidar com um grande conjunto de treinamento, tendo um limite de aproximadamente 100.000 elementos.

3.4 COSTING

Conforme citamos anteriormente, uma base de dados desbalanceada influencia negativamente a aprendizagem de um classificador. Dessa forma, é importante realizar

um balanceamento dos dados na fase de pré-processamento. Porém, é necessário saber como executar este balanceamento de forma a otimizar os resultados. Uma forma de realizar esse balanceamento é atribuir custos diferentes para erros do tipo falso-positivo e falso-negativo e utilizar técnicas de classificação sensível a custos.

Em situações onde o custo de um erro na classificação de um determinado tipo de exemplo é extremamente alto como quando se trata da identificação de uma região cancerígena (onde o custo de um falso-negativo é muito maior do que de um falso-positivo), é muito importante que este custo seja levado em consideração pelo classificador. Caso contrário este último simplesmente rotula todos os exemplos como sendo da classe com maior número de elementos na base, e assim, perde-se completamente o sentido de sua utilização.

Técnicas de classificação que levam custos em consideração são conhecidas como técnicas de classificação sensíveis ao custo (do inglês, *cost-sensitive*). Atualmente, essas técnicas podem ser divididas em três vertentes: novos classificadores sensíveis ao custo; utilização da teoria do risco bayesiana para atribuir cada exemplo à classe de menor custo; técnicas para converter algoritmos arbitrários de aprendizagem em algoritmos sensíveis ao custo, também conhecidas como reduções.

O trabalho [14] segue esta última vertente e propõe a utilização do método Costing para classificação binária, baseando-se na proporção de elementos positivos e negativos do conjunto de treinamento. A partir do custo atribuído a cada exemplo, este método realiza uma redução da amostragem de dados, retirando elementos da classe majoritária. Com este *undersampling*, o custo computacional é extremamente reduzido, com a diminuição da quantidade de memória utilizada e do tempo de processamento das informações.

Apesar de nem sempre atingir resultados tão bons quanto os encontrados com a técnica MetaCost [14], trabalho pioneiro neste mesma vertente, este método possui algumas características marcantes como sua maior simplicidade, garantias teóricas de melhor performance dos algoritmos que o utilizam, além de não utilizar estimativa de densidade de probabilidade. Devido a utilização deste último, o MetaCost [14] é um método que pertence à segunda vertente supracitada.

O Costing é um exemplo de redução, sendo esta técnica utilizada na junção de uma classificação sensível a custos e um algoritmo de aprendizado já existente, como classificação e regressão [6,14]. Na figura 2 pode-se ver um esquema de redução.

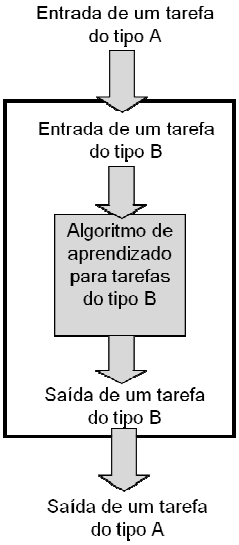

Figura 2: Esquema da redução

Uma redução de uma tarefa A para uma tarefa B é um algoritmo que resolve a tarefa A, supondo que exista outro algoritmo que já resolveu a tarefa B, e o primeiro tem acesso a estas informações. A transferência destas informações acarreta em uma economia de tempo pesquisando uma possível resolução da tarefa B. Uma melhoria neste algoritmo que já existe, também garante uma melhoria na sua redução, sem que o algoritmo que o reduz tenha que ser alterado.

Na classificação sensível a custos, dado um conjunto S de exemplos:

$$S = (x,y,c)^{*} \begin{cases} \boldsymbol{x}: vetor\ de\ atributos \\ \boldsymbol{y}: classe\ (valor\ discreto) \\ \boldsymbol{c}: importância\ do\ exemplo \geq 0 \end{cases}$$

derivados de uma distribuição $\boldsymbol{D}$ com domínio $\boldsymbol{X}$ x $\boldsymbol{Y}$ x $\boldsymbol{C}$, o objetivo é encontrar o classificador

$$\boldsymbol{h}: \boldsymbol{X} \rightarrow \boldsymbol{Y}$$

que maximize o valor esperado da importância dos exemplos que são classificados corretamente:

$$E_{x,y,c\sim D}[\, c\, I(h(x) = y)]$$

A redução na qual é necessária a adaptação de um algoritmo já conhecido para executar a classificação utilizando custos é chamada de *caixa transparente ou branca* (Figura 3). Uma alternativa mais simples, chamada de *caixa preta* (Figura 4), está em fazer uma amostragem de acordo com cada custo e utilizá-la como entrada para o algoritmo de classificação, sem modificar este último. O Costing é um exemplo desta segunda alternativa.

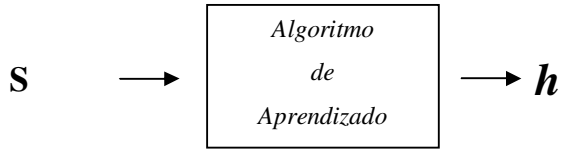

Figura 3: Redução caixa transparente

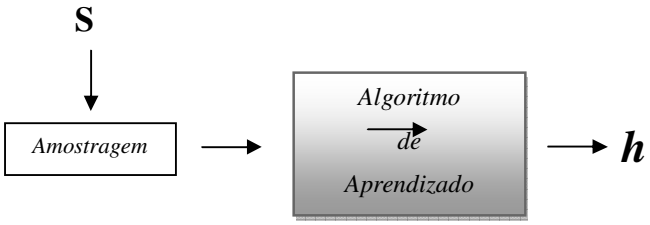

Figura 4: Redução caixa preta

O esquema de amostragem, representado na Figura 4, utilizado por esta técnica para modificar a distribuição dos exemplos é a amostragem por rejeição, decidindo aleatoriamente, para cada exemplo da base de treino, se ele continua ou não na amostra de dados, baseando-se em seu custo, produzindo assim uma amostra de menor tamanho. Outra possibilidade seria uma amostragem com reposição, fazendo um *oversampling,* mas contra este procedimento pesa o fato da base de dados aumentar de tamanho e a possibilidade de passarem a existir dados duplicados na base, sendo assim necessários maiores recursos computacionais para o treinamento do classificador.

Esta amostragem por redução é repetida diversas vezes para formar um número *t* de conjuntos de treinamento distintos. Sendo assim a base de treinamento inicial T gera conjuntos de treinamento T_1, T_2, ... ,T_t . Como estes conjuntos de treinamento são muito pequenos comparados com o tamanho da base T inicial, o tempo requerido para a aprendizagem de um classificador, normalmente, é muito menor. Um classificador *h* é treinado para cada uma destas amostras e a classificação final é feita através de uma votação de todos os classificadores h_1, h_2, ..., h_t.

No algoritmo a seguir, são descritos os passos que devem ser executados pela técnica Costing:

Costing *(Algoritmo de classificação* ***A****, Base de Dados* ***T****, número de amostras* ***t****)*

Para ***i=1*** *até* ***t*** *faça*

$\boldsymbol{T_i}$*= amostra por rejeição de* ***T***

Seja $\boldsymbol{h_i = A(T_i)}$

Saída*:* $h(x) = sinal\left(\sum_{i=1}^{t} h_i(x)\right)$

Mesmo utilizando esta abordagem de múltiplas amostragens da base de dados de treinamento e gerando um classificador para cada base gerada, o tempo total de classificação é, em geral, menor do que se um único classificador fosse utilizado para a base completa, com ou sem custos.

Para avaliar empiricamente o método Costing, [14] utilizou-se de duas bases de dados públicas: as base de dados da KDD-98 e da DMEF-2.

A base de dados KDD-98 contém informações sobre pessoas que fizeram doações para uma determinada instituição de caridade. Os valores doados variam entre $1 e $200. O objetivo do classificador é escolher quais as pessoas receberão uma carta pedindo uma nova doação, sendo que o envio desta carta possui um custo estipulado. A acurácia é medida pelo valor total arrecadado. Em termos de desbalanceamento, apenas 5% das pessoas desta base são doadoras. Temos então:

→ ***x***: **atributos do cliente** *(renda, informações sobre doações anteriores, etc)*

→ ***y:*** **sim/não** *(se houve ou não doação na campanha atual)*

→ ***c:*** ***Se y = sim, então c = valor da doação – custo do envio da carta***
Senão, c = custo do envio da carta

Já a base da DMEF-2 contém informações sobre clientes que costumam comprar produtos em um determinado catálogo. O classificador deve selecionar os consumidores que devem receber um novo catálogo, sendo que apenas 2,5% destes respondem fazem uma nova compra. Assim temos:

→ ***x***: **atributos do cliente** *(informações sobre compras anteriores).*

→ ***y:*** **sim/não** *(se houve compra no catálogo atual).*

→ ***c:*** ***Se y = sim, então c = valor da compra – custo do envio do catálogo***
Senão, c = custo do envio do catálogo.

As tabelas 1 e 2 mostram uma comparação entre os desempenhos alcançados por [14] utilizando o método Costing com os seguintes algoritmos: Naive Bayes (NB), Boosted Naive Bayes (BNB), C4.5 e SVMLight (SVM). O valor entre parênteses é o erro padrão encontrado.

Tabela 1: Referente à base de dados KDD-98

	Sem Custos	t = 1	t = 100	t = 200
NB	0,24	11667(192)	13111 (102)	13162 (68)
BNB	-1,36	11377 (263)	14829 (92)	14714 (62)
C4.5	0	9628 (511)	14935 (102)	15016 (61)
SVM	0	10041 (393)	13075 (41)	13152 (56)

Tabela 2: Referente à base de dados DMEF-2

	Sem Custos	t = 1	t = 100	t = 200
NB	16462	26287 (3444)	37627 (335)	37629 (139)
BNB	121	24402 (2839)	37376 (393)	37891 (364)
C4.5	0	27089 (3425)	36992 (374)	37500 (307)
SVM	0	21712 (3487)	33584 (1215)	35290 (849)

Analisando as duas tabelas, podemos ver claramente a melhora de desempenho de cada classificador, com um exorbitante ganho na utilização de custos. Podemos notar também que os resultados obtidos por C4.5 e SVMLight foram zero, provavelmente porque o algoritmo classifica todos os exemplos como sendo da classe majoritária devido ao desbalanceamento dos dados. Há um significativo ganho quando o número de iterações *t* é aumentado de 1 para 100, sendo este muito maior de que o ganho na variação da iteração *t* de 100 para 200. Isto ocorre porque esta acurácia tende a convergir para um número próximo ao considerado o limite superior quando aumentamos o número *t*. Com isso, a partir de um determinado valor, a variação do número de iterações se torna irrelevante. Também podemos observar que quanto maior este número, menor é o erro padrão encontrado.

A técnica Costing foi integrada ao WEKA e pode ser utilizada em combinação com qualquer método de classificação binária. Para utilizá-lo, devemos criar um arquivo de custos de extensão *.ecost* onde são introduzidos os custos para cada exemplo da base de treino. Este arquivo é dividido em duas colunas, sendo que a primeira indica o custo do exemplo ser classificado como sendo da classe 0 (negativa), e a segunda o custo do mesmo ser classificado como sendo da classe 1 (positiva). Ressaltamos que esta técnica só é utilizada para classificação binária.

Também é necessário que se escolha o número de iterações *t* utilizado para dividir a base aleatoriamente, como descrevemos anteriormente. Como a aleatoriedade não é suportada pelo computador, escolhemos um *seed* (semente) que é utilizado como um fator para a escolha deste número arbitrário. Utilizaremos posteriormente a variação deste *seed* para encontrar a média e o desvio padrão dos resultados obtidos por cada classificador, como descrevemos a seguir no capítulo 4.

CAPÍTULO 4 – METODOLOGIA

4.1 HISTÓRICO

A primeira etapa deste projeto, tendo sido definida a idéia geral de utilização de técnicas de mineração de dados aplicadas à detecção de câncer de mama, consistiu em escolher uma base de dados com informações suficientes para se obter resultados satisfatórios com a aplicação de técnicas de classificação.

Inicialmente, se utilizaria uma base de dados a qual conteria informações retiradas de imagens térmicas, nas quais as temperaturas seriam indicadas por variações de cores. Como as imagens frontais sempre eram formadas pelas duas mamas, a idéia de utilizar informações de simetria surgiu como uma boa solução, assim como a verificação do formato das regiões com temperaturas mais baixas, pois especialistas as consideram regiões potencialmente cancerígenas. Esta é uma área onde a mineração de dados ainda é pouco explorada, então seria muito interessante uma pesquisa neste sentido.

Mas apesar das boas perspectivas de utilização de imagens térmicas, possivelmente nesse caso o foco deste trabalho se aproximaria de processamento de imagens, diferentemente da ideia prevista. Além disso, o principal fator para a desistência da utilização de imagens térmicas foi a dificuldade de acesso a uma base de dados significativamente grande para a utilização de técnicas de mineração de dados. Com isto, nos vimos obrigados a alterar o rumo de nosso trabalho.

Surgiu então a oportunidade de trabalharmos com a base de dados da KDD-CUP. O tema do desafio no ano de 2008 sobre detecção de regiões cancerígenas em mamogramas era ideal e foi disponibilizada no site uma base de dados suficientemente

grande para o nosso trabalho. Com isso definimos o nosso projeto e descartamos definitivamente a idéia inicial de utilização de imagens térmicas.

4.2 BASE DE DADOS

Uma imagem de câncer de mama usualmente é formada por quatro imagens de raios-X, sendo duas imagens de cada seio em diferentes direções. Cada uma destas imagens é constituída de diversas regiões que podem ser cancerígenas, para as quais foi dado o nome de "candidatas". Muitas das candidatas são relacionadas a mesmas lesões em imagens diferentes da mesma mama. Para cada uma destas regiões candidatas foram disponibilizadas informações como ID da imagem, ID da paciente, posição (x,y) na imagem da qual foi extraída, além de diversas outras características não especificadas devido a regras de privacidade da empresa que forneceu os dados. Ao todo, a base contém 117 características diferentes, exclusivamente referentes às imagens. Cada uma destas regiões é classificada como maligna (positiva) ou benigna (negativa). Esta classificação pode ter sido dada através de uma biópsia ou de uma interpretação de um radiologista especializado neste tipo de diagnóstico. Vale ressaltar que o número de regiões relativas a cada paciente é variável.

A base de treino disponibilizada no site da competição consiste em dados de 118 pacientes com regiões diagnosticadas como malignas e 1.594 pacientes consideradas sãs, ou seja, sem suspeita de câncer. Assim, temos um total de 102.294 regiões candidatas e, por ser uma base de treino, cada uma com sua respectiva classificação. Entretanto, apenas 623 candidatos deste total são considerados positivos. Este desbalanceamento de dados se tornou um grande desafio em nossa pesquisa. Ressaltamos que as informações de casa candidato foram divididas em dois arquivos pela organização da competição, denominados arquivo de características (ou *features*) e arquivo de informações (*info*), sendo que este último contém os respectivos rótulos de cada candidato, além de outras informações.

A organização da KDD Cup também disponibiliza uma função para o programa Matlab que cria uma curva definida como *Free Response Receiver Operating Curves*

(FROC), onde a área da região abaixo desta curva define a acurácia da predição do classificador. O gráfico gerado tem em seu eixo x o número de candidatos falsos positivos (*false alarms*) e em seu eixo y o número de pacientes positivos corretamente classificados (*sensitivity*). Além deste gráfico, é dada a AUC (*Area Under the Curve*) que representa a área abaixo da curva FROC. Quanto maior este coeficiente, melhor é a acurácia do classificador utilizado. É necessária apenas uma única região classificada como positiva para a paciente conseqüentemente ser considerada positiva.

Comparando-se falsos positivos, regiões classificadas como malignas indevidamente, com falsos negativos, classificadas como benignas erroneamente, é notório que este último resultará em perdas de proporções muito maiores que o primeiro. Como o objetivo é reduzir o trabalho do especialista em analisar as imagens de raio-X, uma paciente classificada como sã estaria propensa a ter sua doença não diagnosticada, podendo levá-la inclusive ao óbito. Por outro lado, se uma paciente é diagnosticada como tendo uma região maligna, as suas imagens passam por um radiologista ou o mesmo passa por uma biópsia. Assim a doença é descartada pelo especialista e o dano à paciente é nulo, se a mesma não foi impactada emocionalmente por saber que era uma potencial candidata a ter um câncer.

4.3 METODOLOGIA EXPERIMENTAL

Com a base de dados definida, nosso objetivo foi encontrar uma ferramenta que disponibilizasse, de forma simples e gratuita, um conjunto de algoritmos de aprendizagem automática para utilizarmos em nossa pesquisa. O programa que se encaixou perfeitamente nestas especificações foi o WEKA [15], ferramenta amplamente utilizada em mineração de dados, e que contém diversos algoritmos de classificação, regressão e clusterização implementados na linguagem Java. Esta ferramenta possui código aberto, portanto pode-se efetuar qualquer alteração em seu código. Para este trabalho, entretanto, isto não foi necessário. Com isso foram definidos os 3 diferentes tipos de classificadores a serem utilizados: Árvore de decisão (J48), Rede bayesiana (*Naive Bayes)* e *Support Vector Machine* (SMO).

Nossos experimentos iniciaram-se com a divisão da base de dados em 50% para treinos e ou outros 50% para testes de nossos classificadores. Para isso, fizemos um algoritmo randômico que divide a base de dados. Este algoritmo, ao mesmo tempo, executava a formatação dos arquivos com o intuito de que pudessem ser lidos pelo sistema WEKA, o qual espera arquivos no formato *.arff*, agregando informações do arquivo de informações com as do arquivo de características. Com esta base dividida, carregam-se os dois arquivos no WEKA e o mesmo executa o treinamento do classificador selecionado e posteriormente utiliza a base de testes também no WEKA para encontrar a acurácia do algoritmo, comparando-se as predições realizadas no arquivo de testes com sua verdadeira classificação.

Os passos básicos seguidos para todos os experimentos são iniciados, na etapa de pré-processamento do WEKA, com o carregamento do arquivo de treino, seguido da seleção de todos os atributos na mesma janela. A próxima etapa é a escolha do algoritmo de classificação e de seus respectivos parâmetros e o carregamento da base de testes, para podermos iniciar o processo de indução do algoritmo. Antes, porém, é necessário que seja selecionada a opção de exibição de predição de cada exemplo da base de testes, fundamental para a geração da curva FROC. Com estes resultados gerados pelo WEKA, salvamos todas as predições e a matriz de confusão relativa a estas informações. Em seguida, carregamos em um programa feito a parte que recebe como entrada os arquivos de informações e características, e arquivo salvo com as predições, Este programa gera três arquivos, onde cada um contém uma coluna de informações de predição, rótulo verdadeiro, e ID da paciente relacionado a cada candidato. Estes arquivos são carregados no sistema MATLAB para a geração da curva FROC e da respectiva acurácia do classificador executado.

A curva FROC é um gráfico onde podemos extrair a informação sobre a AUC alcançada por um classificador. Essa informação é obtida através da área abaixo da curva do gráfico e este tem como eixo Y a sensibilidade para cada paciente e o eixo X corresponde a taxa de candidatos falsos positivos. Ressaltamos que esta curva faz uma correlação entre a paciente e as áreas candidatas, que são partes de imagens deste paciente. Este tipo de curva é utilizado para avaliar o desempenho dos sistemas CADs e, junto com a AUC gerada, podemos dizer se os resultados gerados em cima da base de teste estão convergindo para um bom valor.

Conforme salientamos anteriormente, o conjunto de dados que utilizamos possui um relevante desbalanceamento de dados, com a esmagadora maioria de seus dados tendo rotulação negativa. Portanto, fez-se necessário um método de minimização deste problema, com a finalidade de melhorarmos a acurácia de nossos classificadores. Com isso, utilizamos o algoritmo de custos descrito em [14]. Este reduz drasticamente o tamanho de elementos da classe majoritária, escolhendo exemplos a serem excluídos randomicamente de acordo com o *seed* (semente) escolhido.

Uma das estratégias utilizadas neste trabalho foi a verificação do melhor custo para cada um dos classificadores utilizados. É importante frisar que cada algoritmo de classificação comporta-se de uma forma diferente, podendo ser este mais ou menos sensível a uma base de dados desbalanceada. Desta maneira, classificadores com maior sensibilidade ao desbalanceamento tendem a ter melhor acurácia com valores de custos maiores do que os utilizados para classificadores menos sensíveis, desde que este valor não seja grande a tal ponto que o desbalanceamento da base se torne o inverso, ou seja, com a classe majoritária se tornando minoritária.

Para comprovarmos a eficácia de cada um dos algoritmos escolhidos, era preciso que os nossos classificadores não fossem dependentes de apenas um número randômico para serem executados, pois exemplos importantes da base de dados poderiam ser excluídos ou não, ficando assim nossa pesquisa dependendo da sorte, algo obviamente não desejado. Assim, para minimizar este grave problema, os experimentos com os classificadores *J48* e *Naive Bayes* e que utilizaram o algoritmo de custos foram feitos com variação de *seed* de 1 a 10. Posteriormente, foi feita a média e encontrado o desvio padrão destes resultados, conforme veremos no capítulo seguinte deste trabalho. Por causa da considerável lentidão do SVM, cerca de 10 vezes maior do que encontrada nos outros algoritmos utilizados, não pudemos executar este passo para o mesmo.

Após acharmos o melhor custo para cada método, verificamos como o número de iterações utilizado influencia diretamente na discrepância entre os resultados obtidos quando variamos o *seed*. Assim, executamos todos os algoritmos variando o número de iterações de cada método com valores entre 1 e 100. No próximo capítulo encontram-se os resultados obtidos nestes experimentos. Mais uma vez, por limitação de nosso

hardware, onde o limite de memória de 1,5 GB a ser utilizado pelo SVM foi insuficiente, ficamos impossibilitados de executar este classificador com número de iterações superiores a 10.

A seguir explicitaremos os resultados obtidos, comparando os resultados encontrados para cada diferente tipo de classificador.

CAPÍTULO 5 – RESULTADOS

Para realizarmos todas as tarefas, utilizamos toda a metodologia descrita no capítulo anterior, e como etapa final do nosso projeto, analisamos os resultados obtidos neste trabalho.

De acordo com a proposta inicial, o foco de nossos estudos baseia-se na utilização do Costing como diferenciador para a obtenção de melhores resultados em comparação com os algoritmos que não são treinados, utilizando informações de custos para cada exemplo da base de treinamento. Um bom resultado consiste em ser alcançado um valor de AUC relativamente alto pelo classificador, o que indica que o modelo gerado pelo algoritmo de treinamento consegue classificar de forma correta os dados utilizados como teste, com uma baixa taxa de erros. Quando maior a AUC, menor o erro do modelo em classificar os exemplos.

Assim, definimos três etapas para avaliação dos resultados obtidos em nossos experimentos. A primeira etapa consiste na análise dos valores de AUC alcançados pelos classificadores gerados pelos três algoritmos de aprendizagem, com e sem a utilização do Costing, além de mostrarmos como é importante a escolha do número de iterações *t* deste método. A segunda consiste em compararmos os valores de AUC gerados entre todos os algoritmos que utilizam esta técnica de redução de desbalanceamento. Por fim, na última etapa faremos um comparativo entre os valores de AUC obtidos com o valor alcançado por [5]. Todas essas comparações foram feitas tendo como base os algoritmos descritos no capítulo 3 e os resultados encontrados nos experimentos.

Inicialmente, rodamos cada um dos algoritmos usados no projeto com um arquivo de custo e usando o número de iterações t = 50. A escolha desse número de iterações se deu pelo fato de que em torno de 50 iterações ocorre a convergência dos

resultados. Porém para o SVM este número não foi possível devido ao estouro de memória, logo tivemos que executar com o t = 10.

Tendo como base esse número de iterações, utilizando-se 10 seeds diferentes para o J-48 e para o Naive-Bayes, e de 1 a 5 para o SVM devido a sua maior lentidão, e variamos os valores de custo. Com isso, geramos os resultados necessários para chegarmos a conclusão de qual custo seria melhor para cada algoritmo. As Tabelas 3, 4 e 5 mostram os resultados que foram gerados.

Tabela 3: AUC obtidas com a variação dos custos para o algoritmo Naive Bayes.

Custo\Seed	1	2	3	4	5	6	7	8	9	10	Média	Desvio Padrão
10	0,0483	0,0493	0,0496	0,0481	0,0483	0,0477	0,0478	0,0483	0,0477	0,0473	0,04824	0,000719877
50	0,0564	0,0536	0,0534	0,0541	0,0542	0,0521	0,0518	0,0515	0,0532	0,0522	0,05325	0,001463823
100	0,0574	0,0551	0,0599	0,0622	0,0569	0,0595	0,056	0,0561	0,056	0,0576	0,05767	0,002215125
300	0,0705	0,0691	**0,0716**	0,0727	0,0683	0,0705	0,0711	**0,0696**	0,0678	0,0637	0,06949	0,002524304
500	0,0717	**0,0719**	0,0714	**0,0728**	**0,0714**	0,0708	**0,0714**	0,0696	0,0708	**0,0668**	**0,07086**	0,001650051
1000	0,0717	0,0701	0,0697	0,068	0,07	**0,0714**	0,0648	0,069	**0,0719**	0,0678	0,06944	0,002168051
2000	**0,0719**	0,064	0,0645	0,065	0,0604	0,062	0,0672	0,0655	0,0619	0,0643	0,06467	0,003216641

Tabela 4: AUC obtidas com a variação dos custos para o algoritmo SVM.

Custo\Seed	1	2	3	4	5	Média	Desvio Padrão
10	0,0587	0,0578	0,0641	0,0571	0,0628	0,0601	0,003144042
50	0,0817	0,0831	**0,0818**	**0,0829**	**0,0831**	**0,08252**	0,00070852
100	**0,0818**	**0,0831**	0,0799	0,0828	0,08	0,08152	0,001512283
300	0,0512	0,0507	0,0502	0,053	0,0538	0,05178	0,001546609
500	0,0512	0,0409	0,0372	0,0367	0,0395	0,0688	0,005898729
1000	0,0254	0,0284	0,026	0,0245	0,0268	0,02622	0,00148054
2000	0,0203	0,0188	0,0187	0,0191	0,0189	0,01916	0,000654217

Tabela 5: AUC obtidas com a variação dos custos para o algoritmo J-48.

Custo\Seed	1	2	3	4	5	6	7	8	9	10	Média	Desvio Padrão
10	0,0854	**0,0862**	0,0859	**0,0864**	**0,0863**	**0,0866**	**0,0864**	**0,0865**	**0,0863**	**0,0861**	**0,08621**	0,000347851
50	0,0856	0,0845	0,084	0,0848	0,0841	0,0847	0,0845	0,0831	0,0854	0,0857	0,08464	0,000800278
100	**0,0878**	0,085	**0,0861**	0,0858	0,0858	0,0852	0,0834	0,0841	0,0816	0,0859	0,08507	0,001701666
300	0,0825	0,0819	0,0824	0,0823	0,0806	0,0816	0,0828	0,0834	0,0841	0,0807	0,08223	0,001095496
500	0,0811	0,0844	0,0843	0,0823	0,0809	0,081	0,082	0,082	0,0803	0,0803	0,08186	0,001481141
1000	0,0813	0,0815	0,0826	0,082	0,0789	0,0809	0,0796	0,0793	0,0801	0,0773	0,08035	0,001614001
2000	0,0795	0,0765	0,0803	0,0806	0,0783	0,0784	0,0725	0,0775	0,0774	0,0756	0,07766	0,002409795

Durante a execução da primeira etapa de análise dos resultados, cruzamos os valores que se encontram na tabela 1 com o valor conseguido pela execução do Naive Bayes sem o uso do Costing, obtendo o valor de AUC igual a 0,0437. Também foram

comparados os valores da AUC do SVM usando o Costing, que se encontram na tabela 2, com o valor gerado pela execução do SVM sem o Costing onde foi encontrado o seguinte valor de AUC igual a 0,0095. Finalmente, cruzamos os valores do AUC do J-48 usando o Costing, que se encontram na tabela 3, com o valor encontrado usando o J-48 sem esta técnica onde o seu valor de AUC encontrado foi 0,0667. Como podemos analisar, o tratamento do desbalanceamento da base de dados de treino utilizando-se a técnica Costing resultou em um ganho considerável em todos os testes realizados.

O gráfico da figura 5 mostra visualmente a comparação entre os melhores valores da AUC encontrados usando o Costing com os valores de cada algoritmo executado sem o a utilização do mesmo.

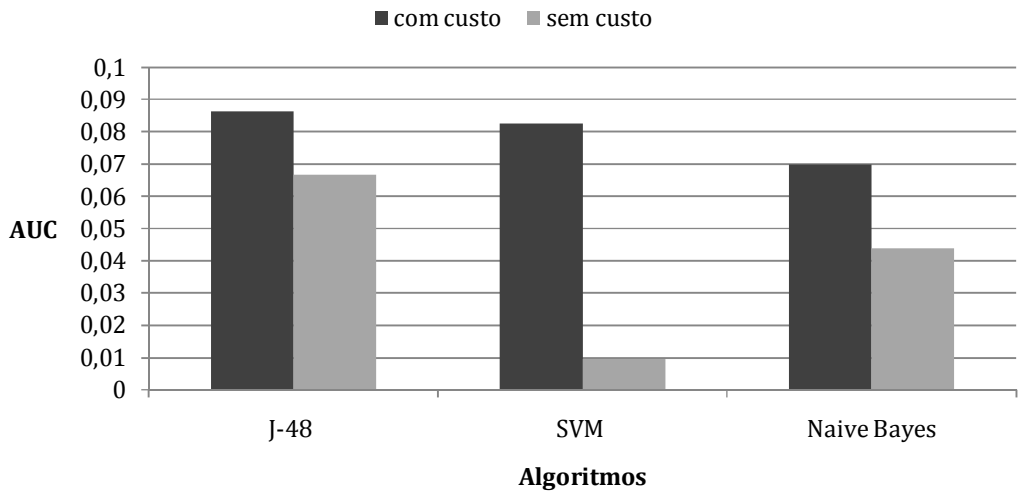

Figura 5: Comparação entre a AUC dos algoritmos com custos e sem custos

Com base no melhor custo de cada método, executamos cada algoritmo novamente, porém desta vez variamos o número de iterações, obtendo os resultados explicitados nas tabelas 6 e 7. Para o algoritmo SVM não foi possível a execução deste procedimento devido a quantidade de memória requerida ser superior à 1,5GB quando o número de iterações utilizado é superior a t = 10.

Tabela 6: Resultados gerados pelo uso do Costing associado ao Naive Bayes.

Iterações\Seed	1	2	3	4	5	6	7	8	9	10	Média	Desvio Padrão
1	0,0389	0,0408	0,0344	0,0381	0,0354	0,0367	0,037	0,0348	0,0278	0,0387	0,03626	0,003583357
10	0,0742	0,0575	0,0461	0,0661	0,0713	0,0633	0,0695	0,0659	0,065	0,0599	0,06388	0,008002888
20	0,0726	0,0661	0,0652	0,0639	0,0693	0,071	0,0728	0,0728	0,0657	0,063	0,06824	0,003885929
100	0,0717	0,0678	0,07	0,0695	0,0717	0,0715	0,0635	0,0695	0,0719	0,0717	0,06988	0,002624161

Tabela 7: Resultados gerados pelo uso do Costing associado ao J-48.

Iterações\Seed	1	2	3	4	5	6	7	8	9	10	Média	Desvio Padrão
1	0,0744	0,0744	0,0783	0,0737	0,0708	0,0767	0,0722	0,0716	0,0745	0,0742	0,07408	0,002252307
10	0,0831	0,0833	0,0849	0,0865	0,0831	0,0839	0,084	0,0809	0,0837	0,0849	0,08383	0,001468219
20	0,0848	0,0824	0,0851	0,0862	0,0845	0,0843	0,0862	0,0845	0,0868	0,0875	0,08523	0,001477272
100	0,0858	0,0854	0,0861	0,0869	0,0869	0,0863	0,0862	0,0865	0,0856	0,0868	0,08625	0,000535931

A seguir, nos gráficos das figuras 6 e 7, temos as médias e os desvios-padrões das tabelas 4 e 5 representados graficamente:

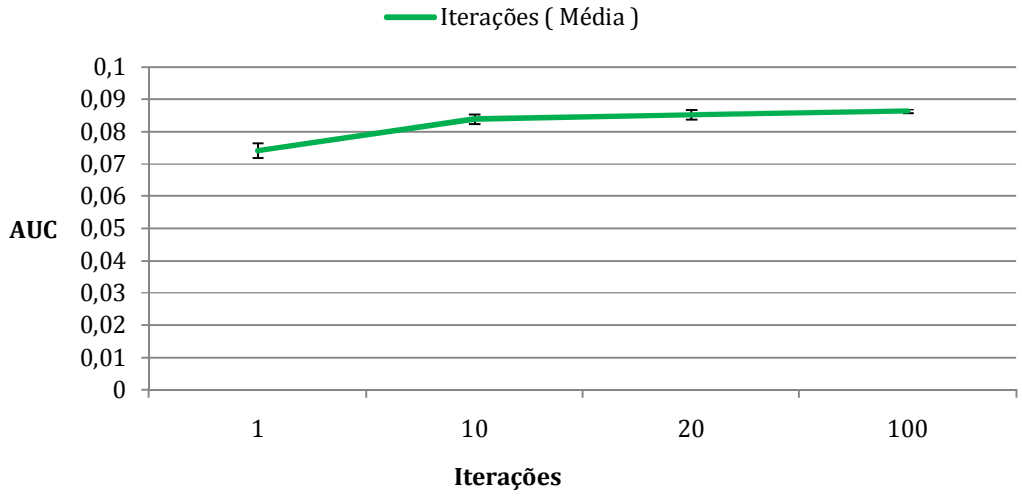

Figura 6: AUC x Iterações do J-48

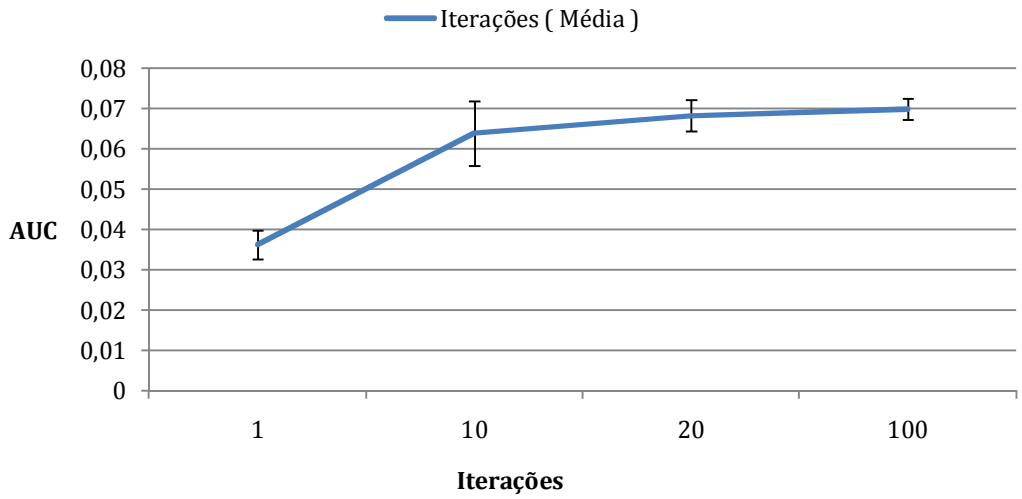

Figura 7: AUC x Iterações do Naive Bayes

Na execução da segunda etapa, foram analisados os valores de AUC de todos os algoritmos usados no projeto e foi verificado o quanto os dados convergiam de forma

adequada para descobrirmos qual algoritmo teve um ganho maior usando a abordagem de custos. A seguir, na Tabela 4, está o comparativo entre os resultados de cada iteração de acordo com o melhor custo para cada algoritmo. A média e o desvio padrão também foram importantes para chegar a conclusão de que o algoritmo J-48 teve o melhor desempenho em relação ao demais.

Tabela 8: Comparação entre os melhores valores de AUC de cada algoritmo.

Algortimo	AUC (média)
J-48	0,08625
SVM	0,08252
Naive bayes	0,06988

Com o gráfico 2, podemos observar os possíveis valores de AUC conforme mudamos o custo dentro dos valores mostrados no gráfico. Podemos observar que na média, o J-48 teve melhores resultados que os demais algoritmos utilizados. Os gráficos das figuras 9, 10 e 11 mostram cada curva do gráfico da figura 8 com a média e o desvio padrão para cada ponto.

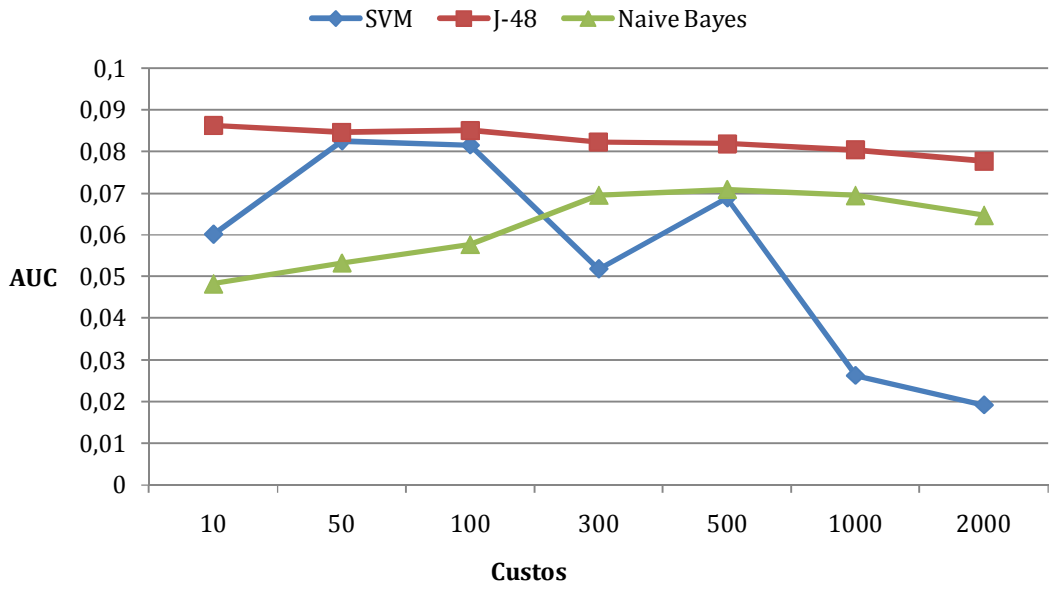

Figura 8: AUC x Custo - Comparação para cada algoritmo

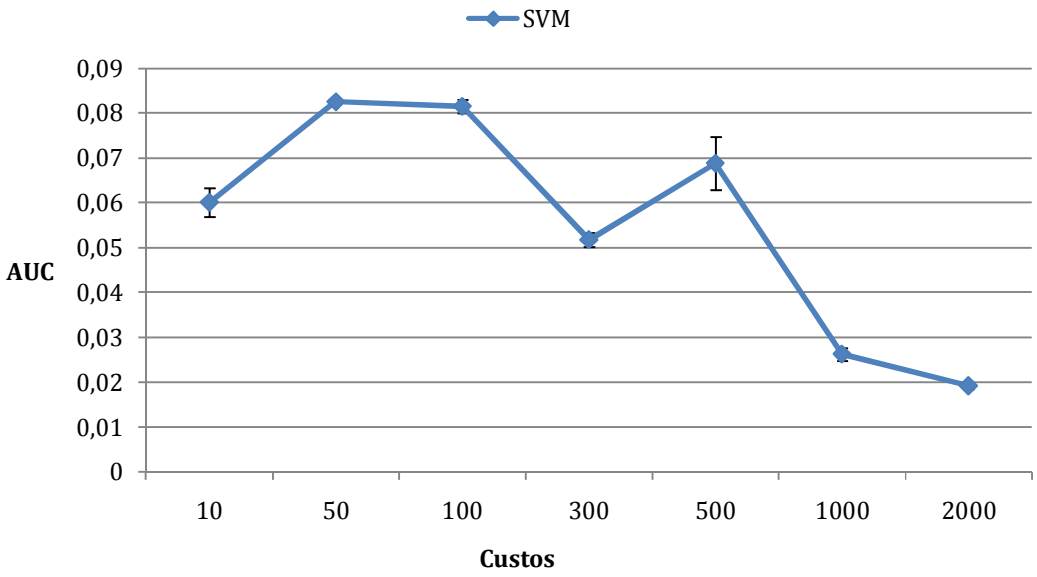

Figura 9: Média e desvio padrão dos resultados do SVM

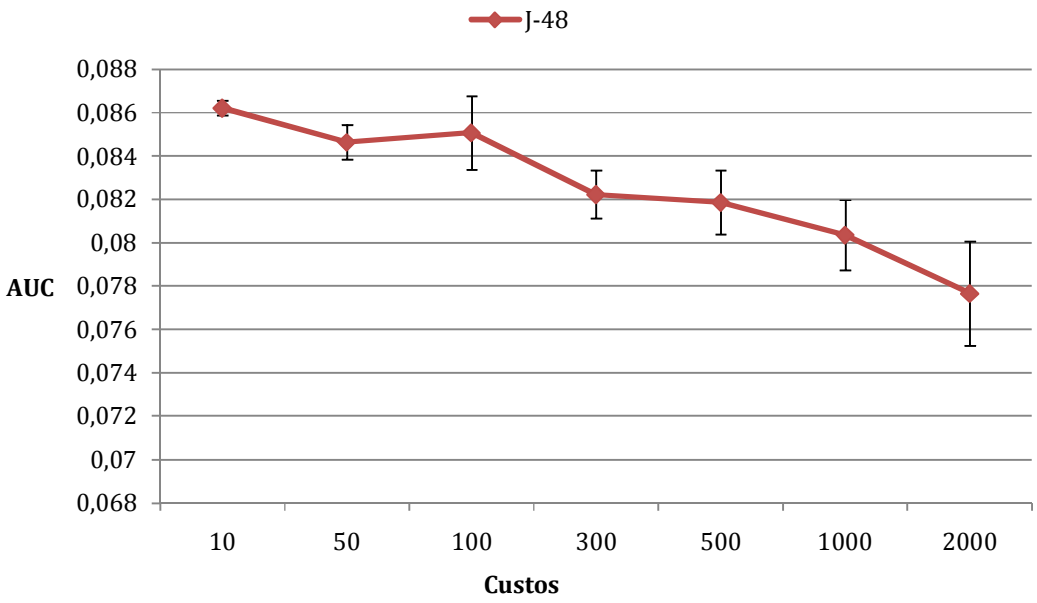

Figura 10: Média e desvio padrão dos resultados do J-48

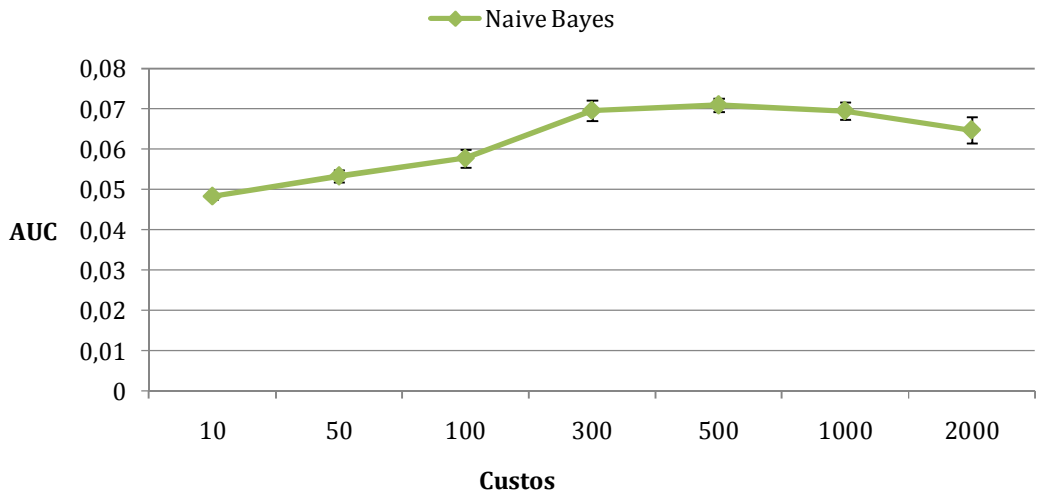

Figura 11: Média e desvio padrão dos resultados do Naive Bayes

Para ilustramos melhor os resultados dos três algoritmos utilizados no projeto, foram gerados três gráficos mostrando a melhor curva FROC obtida com cada algoritmo. Nos gráficos das figuras 12, 13 e 14 podemos observar esse detalhe.

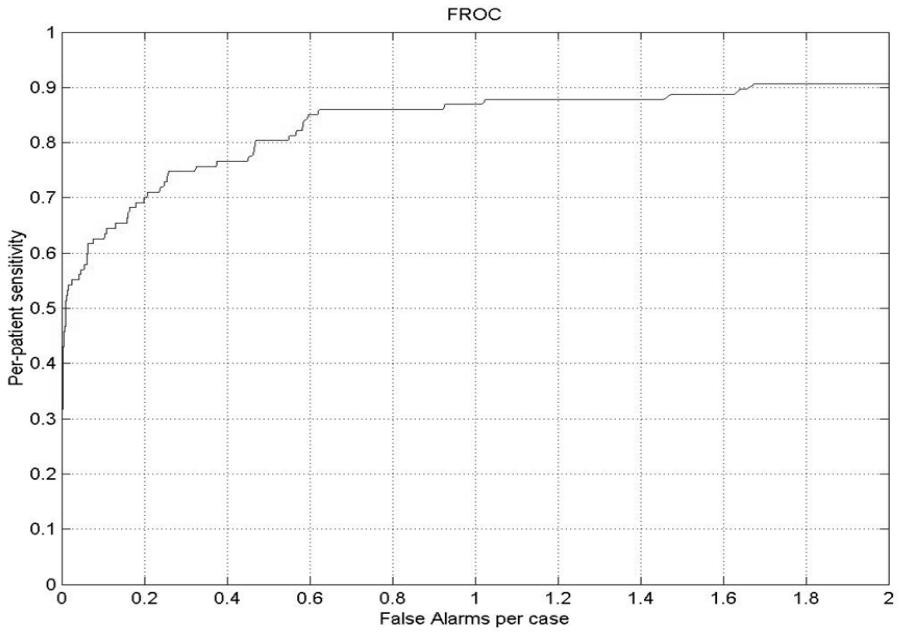

Figura 12: Curva FROC para o melhor valor de AUC encontrado por J-48 com Costing

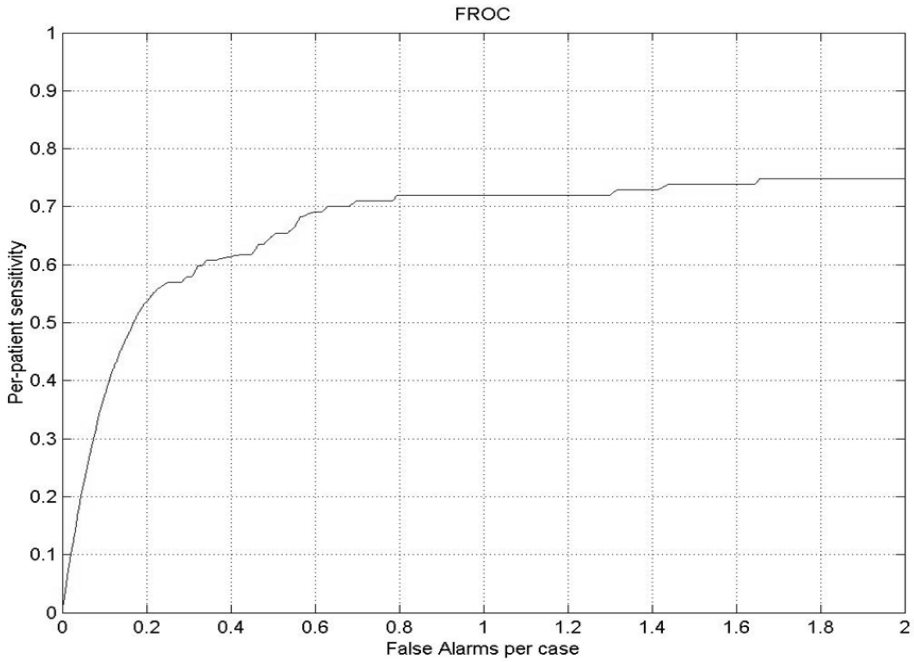

Figura 13: Curva FROC para o melhor valor de AUC encontrado por Naive Bayes com Costing

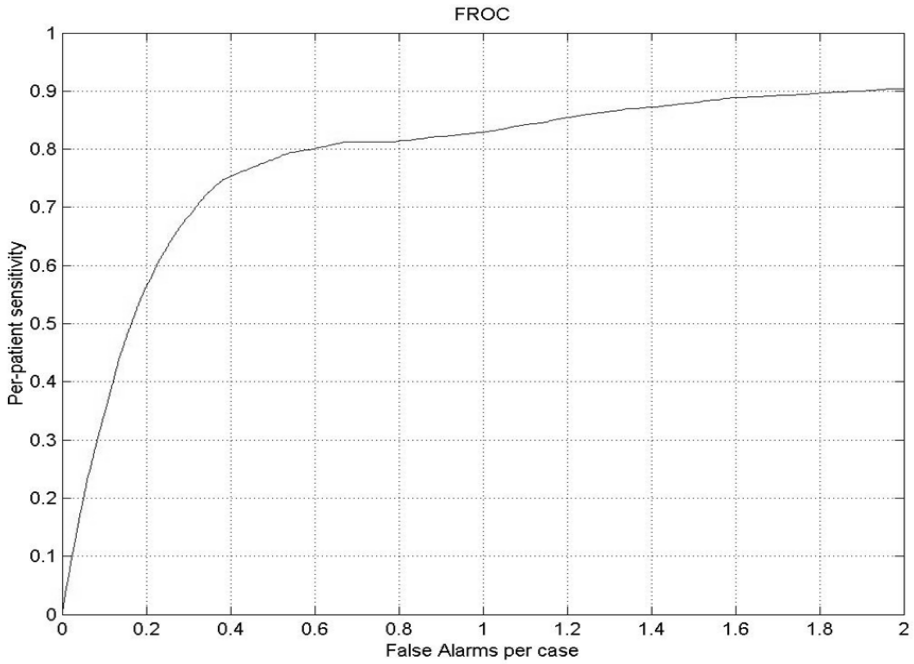

Figura 14: Curva FROC para o melhor valor de AUC encontrado por SVM com Costing

Já na terceira etapa, podemos analisar os resultados obtidos utilizando-se o mesmo algoritmo em comparação com os encontrados pela equipe vencedora [5] do KDD-CUP 2008. Com as duas equipes utilizando o SVM (mas a nossa utilizando o Costing), obtivemos o valor de AUC igual a 0,08252 e a equipe vencedora do KDD-CUP em [5] conseguiu o valor 0,0930. E se por acaso comparamos com o nosso melhor resultado entre todos os algoritmos podemos constatar que chegamos um pouco mais perto do valor encontrado pela equipe vencedora.

A tabela 9 mostra um comparativo entre os valores de AUC analisados. O detalhe para equipe vencedora conseguir atingir um valor de AUC maior foi a utilização do ID dos pacientes que estavam na base. Eles curiosamente carregavam informações importantes para a geração do modelo durante o treinamento do algoritmo.

Este fato chama-se *leakage*, e ocorreu porque a base foi montada com dados de diferentes períodos. Para comprovarmos a ocorrência deste fato, no início de nossas pesquisas, executamos o algoritmo J-48 sem custos e sem a divisão da base de dados, utilizada tanto para treino como para testes, e a diferença entre as AUCs encontradas foi considerável: J48 com informações de IDs dos pacientes conseguiu o resultado de AUC igual a 0,0956, contra 0,0861 do mesmo algoritmo sem esta informação.

Tabela 9: AUC do projeto e da equipe vencedora da KDD-CUP 2008.

Algoritmo	AUC
Equipe do KDD–CUP - SVM	0,0930
SVM	0,08252
J-48	0,08625

Mesmo sabendo desta informação, e que isto traria um ganho significativo nos resultados finais, optamos por não utilizar o ID da paciente como atributo das nossas bases de dados, pois em termos realísticos, com a utilização de sistemas computacionais para ajuda na detecção de câncer de mama na vida real, isto não faria sentido e seria desprezível.

Como podemos concluir, o uso do Costing melhora significativamente os valores de AUC diante dos algoritmos sem o uso do mesmo. E com a escolha de um

valor de custo e de um número de iterações apropriado, podemos fazer com que os resultados convirjam de forma a atingir valores altos de AUC, o que é importante para a resposta de um algoritmo de aprendizagem automático.

CAPÍTULO 6 – CONCLUSÃO

Neste trabalho, propôs-se a utilização da técnica Costing integrada aos algoritmos de aprendizagem automática (Árvore de Decisão, Naive Bayes e SVM) para detecção automática de câncer de mama em imagens.

Os resultados experimentais mostraram que a técnica Costing leva a uma efetiva melhora na acurácia dos classificadores gerados pelos algoritmos supracitados. Também foram comprovados empiricamente menores tempo e memória necessários para o treinamento destes modelos, embora estes resultados não tenham sido relatados explicitamente nesse trabalho.

De acordo com os nossos experimentos, a utilização de técnicas recentes de inteligência artificial aplicadas na detecção de regiões cancerígenas pode realmente ser de grande utilidade, auxiliando assim o trabalho do especialista.

Como trabalho futuro, pretendemos utilizar uma base de dados de imagens térmicas para os experimentos, pois podemos ter com estas um resultado ainda melhor em comparação com os obtidos neste projeto.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] BI, J. *et al. Computer Aided Detection via Asymmetric Cascade of Sparse Hyperplane Classifiers*. Disponível em <http://www.cs.rpi.edu/~bij2/doc/KDD06lung.pdf >. Acesso em 19/10/2009.

[2] PAPADOPOULOS , A. *et al. An automatic microcalcification detection system based on a hybrid neural network classifier*. Disponível em < http://www.cs.uoi.gr/~arly/papers/hybridMCC.pdf>. Acesso em 19/10/2009.

[3] YANGA, L. *et al. Learning Distance Metrics for Interactive Search-Assisted Diagnosis of Mammograms*. Disponível em < http://diamond.cs.cmu.edu/papers/mi2007.pdf >. Acesso em 19/10/2009.

[4] *ABC da Saúde*. Disponível em <http://www.abcdasaude.com.br/artigo.php?611>. Acesso em 22/10/2009.

[5] PERLICH, C. *et al. Breast Cancer Identification: KDD CUP Winner's Report*. SIGKDD Explorations, Vol. 10, Issue 2, 39-42, 2008. Disponível em < http://sites.google.com/site/premmelville/breast-cancer-sigkdd08.pdf?attredirects=0 >. Acesso em 27/06/2009.

[6] ZADROZNY, B. *Estudo e Implementação de Reduções entre Diferentes Tipos de Problemas de Aprendizado de Máquina*, 2006.

[7] MITCHEL, T. ; HILL, M. , *Decision Tree Learning*. Disponível em < http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/mlbook/ch3.pdf> Acesso em 01/11/2009.

[8] ZADROZNY, B. *Introduction to Machine Learning – Árvore de Decisão.* Disponíve em < http://www.ic.uff.br/~bianca/aa/aulas/AA-Aula8.pdf l >. Acesso em 02/11/2009

[9] MARQUES, R. ; DUTRA, I. *Redes Bayesianas: o que são, para que servem, algoritmos e exemplos de aplicações.* Disponível em < http://www.cos.ufrj.br/~ines/courses/cos740/leila/cos740/Bayesianas.pdf >. Acesso em 02/11/2009.

[10] LOWD, D. ; DOMINGOS, P. *Naive Bayes Models for Probability Estimation.* Disponível em < http://www.cs.washington.edu/ai/nbe/nbe_icml.pdf >. Acesso em 03/11/2009.

[11] DEMICHELIS, F. et al. *A hierarchical Naïve Bayes Model for handling sample heterogeneity in classification problems: an application to tissue microarrays.* Disponível em < http://www.biomedcentral.com/1471-2105/7/514 >. Acesso em 03/11/2009.

[12] LOVELL, B. ; WALDER, C. *Support Vector Machines for Business Applications*. Disponível em <http://www.nicta.com.au/__data/assets/pdf_file/0020/14960/Support_Vector_Machines_for_Business_Applications.pdf >. Acesso em 03/11/2009.

[13] SCHOLKOPF, B. *et al. Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classiers*. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.4712&rep=rep1&type=pfdf>. Acesso em 03/11/2009.

[14] ZADROZNY, B. *et al. Cost-Sensitive Learning by Cost-Proportionate ExampleWeighting*. Disponivel em < http://www.hunch.net/~jl/projects/reductions/costing/finalICDM2003.pdf >. Acesso em 4/10/2009.

[14] PAPADOPOULOS, A. et al. *An automatic microcalcification detection system based on a hybrid neural network classifie.* Disponível em < http://www.cs.uoi.gr/~arly/papers/hybridMCC.pdf >. Acesso em 4/10/2009.

[15] The Universe of Waikato, WEKA Machine Learning Project, Waikato Environment for knowledge Analysis. Disponível em <http://www.cs.waikato.ac.nz/ml/weka/>. Acesso em: 10/06/ 2009.