Classificação Supervisionada de Câncer de Mama Utilizando o Algoritmo SVM

Khalil Carsten do Nascimento
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
khalilcarsten@gmail.com

Renato Avelar Nobre
Departamento de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
rekanobre@gmail.com

***Index Terms*—SVM, câncer de mama, classificação supervisionada, kernel linear, kernel rbf**

I. Introdução

O câncer de mama é o principal tipo de câncer que afeta o publico feminino, o mesmo devido à multiplicação de células anormais no tecido mamário, formando um tumor maligno, inicialmente imperceptível, que pode aumentar e atingir outros locais do corpo.

A mamografia pode levantar a suspeita do câncer de mama, entretanto, a confirmação é feita após consulta com o mastologista, que irá fazer uma avaliação mais detalhada do nódulo e do exame e, se necessário, solicitar exames que podem ser mais específicos, como ultrassom, ressonância magnética ou, se a suspeita persistir, uma biópsia do nódulo mamário.

Nota-se que a detecção do câncer de mama não é um processo simples, sendo necessário assim um método eficiente de gerar um diagnóstico. Portanto, nosso trabalho propõe uma abordagem utilizando o algoritmo de aprendizagem de máquina SVM para tentar classificar suspeitas de câncer de mama como dados de exames FNA, classificando o tumor como maligno ou benigno.

II. Especificações do Sistema

A estrutura do sistema foi desenvolvida para evitar acoplamento entre os algoritmos de aprendizagem, os algoritmos de tratamento de imagens, e os algoritmos de suporte. O sistema adaptador é responsável por conciliar o ambiente para respeitar as interfaces requeridas pelo usuário ao realizar entradas pela linha de comando.

Todo o sistema foi construído utilizando a linguagem de programação *Python 3* que é a linguagem usada em milhares de aplicações de negócio no mundo, incluindo diversos sistemas de larga escala [1]. Para o auxilio no desenvolvimento das funções de classificação, foi utilizada uma biblioteca de aprendizagem de máquina denominada *scikit-learn*. A biblioteca, acessível para todos, é um conjunto de métodos simples e eficientes para mineração e analise de dados [2]. Também foi utilizada a biblioteca *matplotlib* para criar gráficos e a matriz de confusão [3]. E a biblioteca *pandas* para fazer a leitura dos dados [4].

III. Dados e Características

O banco foi criado pelo Dr. William H. Wolberg, W. Nick Street e Olvi L. Mangasarian, todos docentes da universidade de Wisconsin. Os dados possuem 569 linhas e 32 colunas. Cada linha descreve um número de características com respeito aos núcleos de células digitalizados em imagem atráves o procedimento FNA nas massas encontradas em mamas. Das 32 colunas a primeira diz respeito ao ID da imagem e a segunda à classificação, benigna ou maligna, para a massa encontrada. As outras 30 colunas são um conjunto de três grupos de dez características. Cada grupo das 10 características corresponde ao valor médio, o desvio padrão e piores ou maiores valores encontrados daquelas características extraídas das células das imagens, respectivamente.

- Raio (média das distâncias do centro até os pontos de perímetro)
- Textura (desvio padrão das escalas de cinza)
- Perímetro
- Área
- Suavidade (Variação local dos Raios)
- Densisdade $(perimetro^2/area - 1.0)$
- Concavidade (severidade das partes côncavas do contorno)
- Ponto concavos (número de pontos côncavos)
- Simetria
- Dimensão do Fractal $(\text{“}coastline \quad approximation\text{”} - 1)$

Tendo essa estruturas usaremos 30 características, alocadas em um vetor para cada imagem, e 2 classificações possíveis no problema, maligno e benigno, onde vieram dividas em 357 benignas e 212 malignas.

IV. Proposta e Desenvolvimento

Cada característica citada anteriormente se tornará um elemento de um vetor v onde cada serão no total 569 vetores com 30 elementos cada. Esse será a principal entrada do

[1] Disponível em: https://www.python.org
[2] Disponível em: http://scikit-learn.org
[3] Disponível em: https://matplotlib.org
[4] Disponível em: http://pandas.pydata.org

classificador SVM. Os dados foram divididos da forma 70/30, onde 70% das amostras serão para treinamento e 30% para executar os testes. Foram feitas três separações diferentes de maneira aleatória para diversificar os testes abrangendo melhor os resultados da classificação da SVM para a mesma coletânea de dados.

Tendo em vista a estrutura dos dados explicados anteriormente faremos teste experimentais utilizando o algoritmo SVM com kernel Linear, RBF e gaussiano. Para os testes com kernel linear executaremos com os valores de C, parâmetro de penalidade do termo de erro, para gaussiano, o σ e por último γ para o RBF. Foram feitos para 100 valores de C, σ e γ numa escala logarítmica nos intervalos de 2^{-5} a 2^{15}.

Os teste experimentais tem o intuito de explorar os dados entendendo sua distribuição e encontrar valores que levem ao menor taxa de erro através dos parâmetros $C, \sigma e \gamma$.

V. Resultados Experimentais

Retirando a média de todos os vetores advindos dos dados e mostrando em um gráfico de distribuição notamos que não seguem uma distribuição normal em parte dos dados. Devido a isso a divisão aleatória dos dados citada anteriormente pode acarretar em diferenças significativas no resultado final das repetições do testes. Como mostrado na V se durante a escolha aleatória os dados ficarem muito próximos da parte com 30% pode gerar uma taxa de erro maior pelo fato de não estar entre os dados mais dentro da média geral.

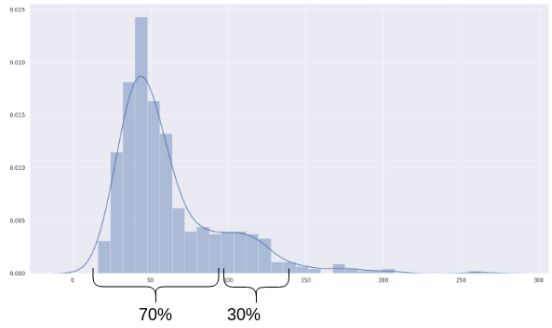

Figura 1. Distribuição da média dos dados

Abaixo vemos três gráficos com respeito a taxa de erro em relação aos valores de C com um Kernel Linear. Em todos os três testes com uma divisão aleatória dos dados o valor de C não converge para um valor comum de menor erro, isso devido a aleatoriedade em dados pouco normalizados. Todos os três testes atingiram valores altos de precisão chegando a uma média de 97%.

Avaliando agora os resultados do kernel Gaussiano notamos que houve uma certa convergência do valor de sigma entre os intervalos de 2^6 a 2^9 o que indica que pelo tamanho de σ os dados estão com uma alta variâncias sendo melhor separados quando o valor resultante de 1 é menor, ou seja, a variância σ maior. A precisão do Gaussiano somente chega aos 90% no intervalo citado anteriormente.

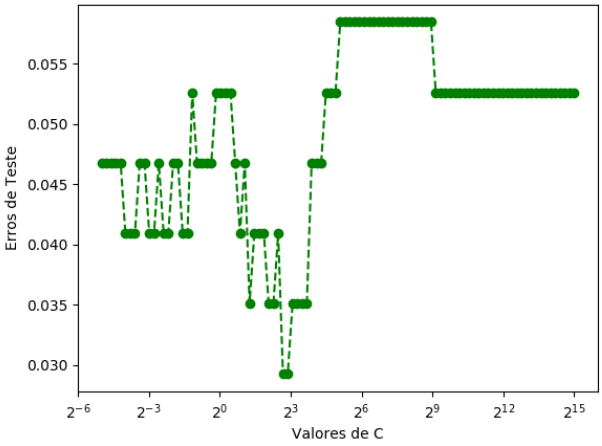

Figura 2. 2 - Gráfico erro de Teste x Valores de C relativos ao kernel Linear

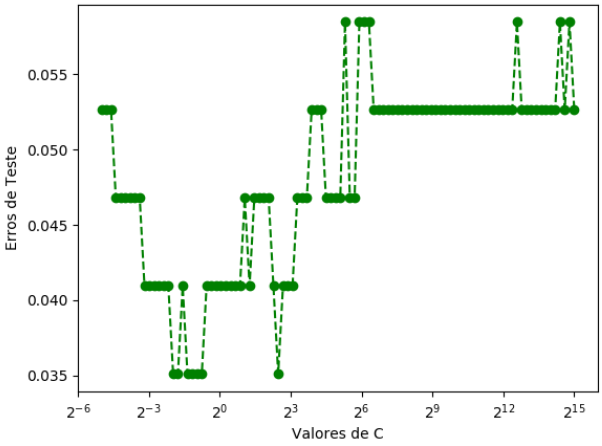

Figura 3. 1 - Gráfico erro de Teste x Valores de C relativos ao kernel Linear

$$K(x, x') = exp\left(- \frac{||x - x'||^2}{2\sigma^2} \right) \quad (1)$$

Agora olhando para o kernel RBF temos que não houve alteração nos resultados com o decorrer da mudança de gama. Isso pode ser explicado por 2 onde $\gamma = 1/2\sigma^2$, então é basicamente uma multiplicação de um escalar na diferença, logo acontece o inverso do mostrado no kernel anterior.

$$K(x, x') = exp(-\gamma||x - x'||) \quad (2)$$

VI. Conclusão

Com os resultados mostrados podemos ter uma clara visão de que o problema tende para uma resolução linear. Podemos constatar isso pela precisão média de 97% no kernel Linear em relação aos kernels de base radial, RBF e Gaussiano. Tendo isso em mente podemos perceber a importância da

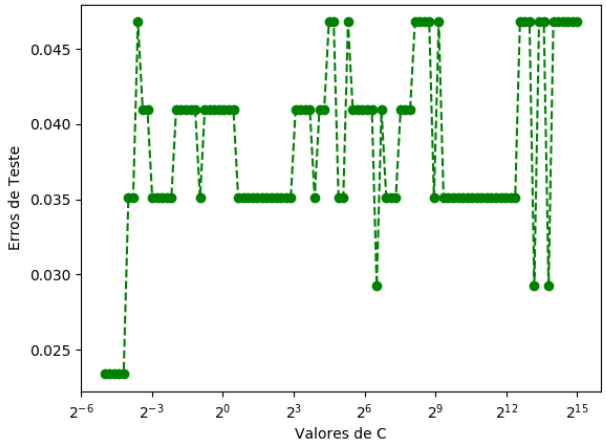

Figura 4. 2 - Gráfico erro de Teste x Valores de C relativos ao kernel Linear

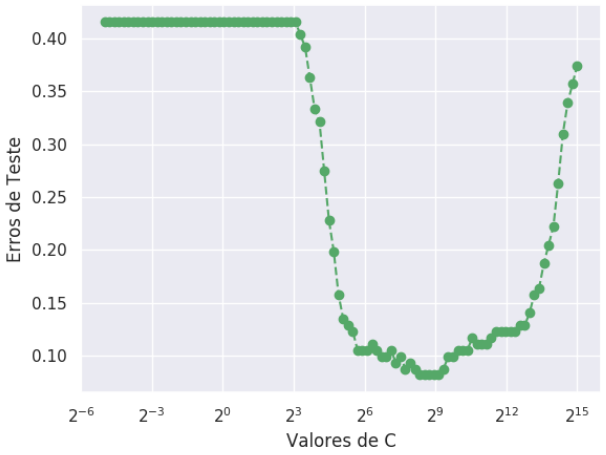

Figura 5. Gráfico erro de Teste x Valores de σ relativos ao kernel Gaussiano

decisão da função kernel para a resolução de um problema de classificação. Foi entendido também como cada fórmula de Kernel usada funciona e como seus parâmetros podem afetar significativamente na divisão de classes.

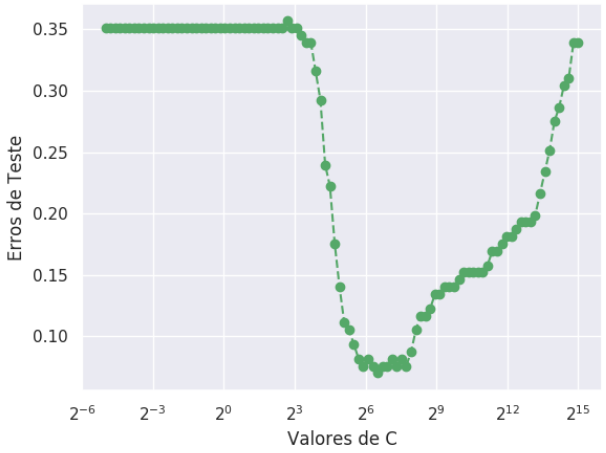

Figura 6. Gráfico erro de Teste x Valores de σ relativos ao kernel Gaussiano

Figura 7. Gráfico erro de Teste x Valores de γ relativos ao kernel RBF