UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS SOBRAL

Curso de Engenharia Elétrica

Inteligência Computacional

Relatório 2

Aluno: Julio Cesar Ferreira Lima - 393849

Prof: Jarbas Joaci de Mesquita Sá Junior

Sobral, **Junho** de **2018**.

1. Introdução

1.1. Método do Mínimos Quadrados

Nesse relatório foi utilizada uma técnica utilizada na estatística de modo a obter a curva que mais se aproxima de um padrão de amostragem de dados. A priori tal estratégia pode não parecer uma boa estratégia relacionada à inteligência computacional, no entanto por descrever o comportamento da maioria da população de amostras, demonstra tendência o que pode ser considerado aprendizado. Defendido pelo Teorema Gauss-Markov que garante confiabilidade não enviesada de mínima variância linear e desenvolvido por Carl Friedrich Gauss, quando tinha apenas dezoito anos, aqui, sua qualidade de estimativa será medida pelo coeficiente de determinação, também chamado de R^2, que é uma medida de ajustamento de um modelo estatístico linear generalizado assim como um variação desse parâmetro ajustado, que considera a qualidade das amostras e não só totalmente sua quantidade, retirando assim a importância do ruído nas amostras. Todo esse trabalho será usada variâncias dessas teorias, variando apenas a dimensionalidade do sistema e a ordem da estimativa.

Ao se aproximar de uma curva, é natural que exista um erro associado, assim temos essa aproximação dada por:

$$y = \alpha + \beta x + \varepsilon,$$

Sendo “a” e “b” parâmetros de aproximação, “e” o erro associado “y” a saída da instância de “x”.

Discorrendo sobre a reta poderemos fazer o somatório dos erros dado um mesmo conjunto de parâmetros “a” e “b”. Tais erros podem ser somados e será mínimo quando o *fit* da curva for ótimo. Considerando que há uma curva onde a combinação dos parâmetros “a” e “b” são entradas e que é igual a soma dos erros totais, temos:

$$S(a,b) = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

Tomando a derivada parcial de cada um dos parâmetros e determinando a função para que essa variável seja zero, dado que nesse ponto será a maximização do *fit* e logo minimização do erro associado a aquele parâmetro, temos:

$$\frac{\partial S}{\partial a} = \frac{\partial S}{\partial x} * \frac{\partial x}{\partial a}$$

$$\frac{\partial S}{\partial x} = 2\sum_{i=1}^{n}(y_i - a - bx_i)$$
$$\frac{\partial x}{\partial a} = -1$$
$$\frac{\partial S}{\partial a} = -2\sum_{i=1}^{n}(y_i - a - bx_i) = 0$$
$$\frac{\partial S}{\partial b} = -2\sum_{i=1}^{n} x_i\,(y_i - a - bx_i) = 0$$

Após alguns algebrismos para a determinação de “a”:

$$\frac{-2\sum_{i=1}^{n} y_i}{2n} + \frac{2\sum_{i=1}^{n} a}{2n} + \frac{2\sum_{i=1}^{n} bx_i}{2n} = \frac{0}{2n}$$
$$\frac{-\sum_{i=1}^{n} y_i}{n} + \frac{\sum_{i=1}^{n} a}{n} + \frac{b\sum_{i=1}^{n} x_i}{n} = 0$$
$$-\bar{y} + a + b\bar{x} = 0$$
$$a = \bar{y} - b\bar{x}$$

Mais alguns algebrismos para a determinação de “b”, substituindo “a”.

$$-2\sum_{i=1}^{n} x_i\,(y_i - \bar{y} + b\bar{x} - bx_i) = 0$$
$$\sum_{i=1}^{n}\left[x_i\,(y_i - \bar{y}) + x_i b\,(\bar{x} - x_i)\right] = 0$$
$$\sum_{i=1}^{n} x_i\,(y_i - \bar{y}) + b\sum_{i=1}^{n} x_i\,(\bar{x} - x_i) = 0$$
$$b = \frac{\sum_{i=1}^{n} x_i\,(y_i - \bar{y})}{\sum_{i=1}^{n} x_i\,(x_i - \bar{x})}$$

Extrapolando a Regressão Linear Simples para a Regressão Multipla, temos então a possibilidade de multidimensionalidade e aumento de ordem do aproximador, dado que não há apenas mais um “x” e sim um “xi”, além de que há a soma e atribuições de pesos para variável aleatória “xi” e há progressão em graus, não limitando-se mais apenas ao primeiro.

Extrapolando em relação a dimensionalidade:

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + \cdots + x_k\beta_k + \varepsilon$$

Extrapolando em relação ao número de graus, temos que valor de uma mesma coluna em “xi” correspondem ao mesmo grau e ele crescer da esquerda para a direita.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \cdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{21} & \cdots & x_{k1} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ 1 & x_{13} & x_{23} & \cdots & x_{k3} \\ 1 & x_{14} & x_{24} & \cdots & x_{k4} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{pmatrix} \times \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \cdots \\ b_k \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ \cdots \\ e_n \end{pmatrix}$$

Simplificando tudo em matrizes, voltamos simplicidade da tomada inicialmente:

$$y = Xb + e$$

Tomando posse de alguns algebrismos para a simplificação da extração de beta, temos:

$$\begin{aligned} S(b) &= (y - Xb)'(y - Xb) \\ &= y'y - y'Xb - b'X'y + b'X'Xb \end{aligned}$$

$$\begin{aligned} \frac{\partial S}{\partial b} &= -2X'y + 2X'Xb = 0 \\ X'Xb &= X'y \\ b &= (X'X)^{-1}X'y \end{aligned}$$

Objetivando agora a quantização do *fit* curva, faz-se uso do do coeficiente de determinação:

$$SQ_{\text{tot}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

1.2. *Leave One Out*

Quando utilizando regressão, há geralmente duas maneiras de se mensurar o erro. A primeira delas é por meio do erro obtido com o treinamento, já a segunda por meio do erro provocado pela generalização. A o primeiro parâmetro diz respeito da capacidade de assimilação dos dados a uma

tendência ou convergência, enquanto que a segunda se se preocupa com a capacidade de aprendizado se daquele teste se submetido a novos testes, ou seja, sua capacidade de predição.

Cross-validation prove uma estimação para o erro de generalização, onde o foco consiste na aplicação do método *leave-one-out*. O método propõe a captura de uma amostra, deixando-a de fora da construção de um modelo a partir do restante dos dados. Ao fim a amostra capturada será testada e algoritmos estatísticos podem ser utilizados para a determinação do grau daquela amostra ao modelo gerado.

O inconveniente desse método é a geração de um modelo para cada amostra testada o que requer um poder computacional muito grande. No entanto se mostra eficiente para pequenas amostras ou *data-sets*.

1.3. *K-Nearest Neighbourhood (KNN)*

Trata-se de um padrão de reconhecimento, bastante utilizado com o método de regressão, onde as entradas consistem nos k mais próximos dados de treino de determinada amostra e a saída corresponde à média, mediana ou moda das distâncias.

A classificação k-NN é considerada um tipo de *lazy-learning*, aprendizagem preguiçosa uma vez que a função é apenas aproximada localmente e toda a computação é adiada até a classificação.

É comumente utilizada como medida a distância Euclidiana. No entanto para variáveis discretas e sistemas de classificação textual, outros métodos de medida podem ser utilizados como a distância de Hamming.

Frequentemente, a precisão de classificação de k-NN pode ser melhorada significativamente se a métrica de distância for aprendida com algoritmos especializados, *Large Margin Nearest Neighbor* ou *Neighbourhood Components Analysis*.

1.4. *Holdout*

Holdout é outro exemplo de *cross-validation*. A implementação deste consiste na separação entre os dados de teste e os dados de treino. Então uma função de aproximação é utilizada para predizer os dados.

A vantagem deste método em relação ao *leave-one-out* é o seu ganho computacional uma vez que é preciso a geração de apenas um modelo. No entanto, poderá haver variação muito grande se os dados não estiverem dispostos de maneira aleatória, desse modo os dados retirados para teste devem representar de forma genérica o todo, de modo a não tornar significantemente diference a variação.

2. Metodologia

A base para todo o procedimento experimental desenvolvido nesse experimento/simulação está demonstrada na introdução. O passo a passo executado pode ser observado com a leitura do código, onde está documentado as tomadas de decisões que fazem parte dos algoritmos utilizados, portanto, de modo a não onerar este relatório, apenas será mostrado a saída do MATLAB.

Visto que a língua padrão de toda a comunidade científica é o inglês e que os códigos produzidos neste trabalho serão disponibilizados na plataforma GitHub, todos os comentários do

código, incluindo tomadas de entrada e saída foram escritos em inglês, tornando-o não capaz, no entanto, mais possível, que esse trabalho se torne útil a outrem.

3. Resultados

O primeiro teste trata-se do *leave-one-out*, no qual o *data-set* foi percorrido e sempre deixando uma amostra de fora e gerando um modelo por meio do classificador Mínimos Múltiplos Quadrados, após isso foi testado a amostra reservada. Cada amostra do data-set passou por esse processo gerando um modelo para cada amostra deixada de fora, e gerando um erro devido ao taxa de pertencimento da amostra ao modelo gerado. Ao fim a acurácia do modelo foi de 77,42%, como pode ser verificado abaixo.

Figura 1 – Entrada: Leave One Out.

```
1 - Least Squares Multiple Classificator:
2 - KNN Classificator:
Type the question: 1
```

Fonte: Autor

Figura 2 – Saida: Leave One Out.

```
Tax classification awards, Least Squares Multiple with Leave One Out: 77.4264%
...
Do you wanna try one more test? (y\n):
```

Fonte: Autor

O segundo teste trata-se do KNN, no qual o *data-set* foi percorrido e inicialmente com 70% para o treinamento e 30% destinados ao teste, tais amostras foram captadas de maneira aleatória. O k inicial foi utilizado 1 e as épocas correspondem a quantidade de vezes que o algoritmo repetiu os mesmos passos, ao fim o resultado corresponde a média dos resultados das amostras. A acurácia pode ser demonstrada em tempo real conforme há a atualização das épocas. Coo pode ser verificado com as figuras abaixo.

Figura 3 – Entrada 1: KNN com Holdout.

```
1 - Least Squares Multiple Classificator:
2 - KNN Classificator:
Type the question: 2
```

Fonte: Autor

Figura 4 – Entrada 2: KNN com Holdout t, K=1 Treinamento=70%, Épocas=10.

```
What K-Nearest Neighbourhood you will gonna use?: 1
What percent you will gonna use for training?: 70
How many time do you wanna repeat this?: 10
```

Fonte: Autor

Com essas configurações fora obtido 62% de acurácia, como pode ser observado com a figura abaixo. No entanto, o algoritmo fora implementado de maneira que pudesse ser variado tanto k, de modo a obter a moda da classe dos k vizinhos mais próximos, assim como a variação da porcentagem de treinamento e a quantidade de épocas.

Figura 5 – Saída: KNN com Holdout, K=1 Treinamento=70%, Épocas=10.

```
Tax classification awards, KNN with Hold Out: 62%
...
Do you wanna try one more test? (y\n):
```

Fonte: Autor

Desse modo, fora reduzido a quantidade de amostras para treinamento, o provocaria uma redução na acurácia, e diminuição do *over-fit*, no entanto os demais parâmetros foram aumentados de forma a buscar uma maior acurácia mesmo com uma menor quantidade de dados, 10% a menos.

Então mantido k igual a 1 e tomando um número de épocas igual a 20 a acurácia manteve-se a mesma com o 70% das amostras destinadas a treinamento.

Figura 7 – Saída: KNN com Holdout, K=1 Treinamento=60%, Épocas=20.

```
Tax classification awards, KNN with Hold Out: 62%
...
Do you wanna try one more test? (y\n):
```

O melhore resultado obtido, com apenas alguns testes foi obtido com k igual a 3 e tomando um número de épocas igual a 20.

Fonte: Autor

Figura 6 – Saída: KNN com Holdout, K=3, Treinamento=60%, Épocas=20.

```
Tax classification awards, KNN with Hold Out: 66%
...
Do you wanna try one more test? (y\n):
```

Fonte: Autor

Assim, justifica é justificada a análise da variação dos parâmetros na utilização do método dado uma significativa melhora na taxa de acerto.

4. Conclusão

A classificação em classes das mais variadas entradas é uma das características em que o ser humano se destaca com excelência, não só pela capacidade de classificação, mas pela variabilidade adaptativa que o classificador humano possui. Juntamente com outras zonas do conhecimento torna-o excelente em diagnósticos, previsões, aprendizado e até mesmo decisões simples como qual é a melhor fruta a se comprar. No entanto as maquinas possuem excelente capacidade de lembrar-se de fatos e de processamento em força bruta, que poderia ser muito útil em diagnósticos, essa linha de raciocínio demonstra a desse tipo de estudo. Assim, o estudo de classificadores pode ser exemplificado como importante dada a capacidade de criação de um banco de doenças mundiais que estão relacionadas a características que podem vir a serem sintomas ou variáveis de ambiente e por meio de uma matriz de transformação, com base nas características de cada paciente poderá ser retornada uma saída que corresponde com certa acurácia quais doenças o paciente pertence.

O estudo baseado nesse relatório torna possível a criação do referido banco de dados de doenças, bastando apenas uma consulta com especialista em doenças que indique uma centena de casos de doenças de pacientes e sintomas relacionados. A aplicação desta teoria usa como base o método dos múltiplos quadrados. A maior facilidade encontrada na utilização desse diagnosticador é a implementação que é de extrema simplicidade.

Se cada problema de classificação pudesse ser disposto em uma matriz de k-dimensão a análise do pertencimento de uma determinada amostra poderia ser considerada a distância mínima entre a amostra e classe mais próxima. Assim se destina o método *KNN*, que é de simples entendimento, no entanto sua junção com o método *Hold Out* torna sua implementação muito complexa. Na utilização desses métodos, primeiramente primeiro deve-se gerar índices aleatórios para as bases de teste e treino, depois testar a distância de cada ponto da base de teste com todos os pontos da base de treino de modo a encontrar k-próximos vizinhos, nesse momento deve ser bloqueado a modificação de cada um dos parâmetros caso seja encontrada um vizinho com menor distância, esse só deverá modificar uma posição. A seguir deverá testar se a amostra de teste possui a mesma classe da moda dos k-próximos vizinhos. Nesse relatório fora utilizada moda pela facilidade de implementação e acurácia na obtenção de resultados devido ao problema utilizado, para demais problemas poderá ser utilizado parâmetros como média, mediana etc.

5. Bibliografia

[1] JUNIOR, Jarbas. **Regressão Linear Múltipla** . Notas de aulas. Acesso em Maio de 2018.

[2] JUNIOR, Jarbas. **Classificadores Elementares II**. Notas de aulas. Acesso em Maio de 2018.

[3] WIKIPEDEA. **Método dos mínimos quadrados**. Disponível em: <https://pt.wikipedia.org/wiki/Método_dos_mínimos_quadrados>. Acesso em abril de 2018.

[4] LANDY, Jonathan. **Leave-one-out cross-validation**. Disponível em: <http://efavdb.com/leave-one-out-cross-validation/ >. Acesso em Junho de 2018.

[5] WIKIPEDEA. **K-Nearest Neighbors Algorithm**. Disponível em: <https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm>. Acesso em Junho de 2018.

[5] SCHNEIDER, Jeff. **Cross Validation**. Disponível em: <http://efavdb.com/leave-one-out-cross-validation/ html>. Acesso em Junho de 2018.

6. Repositório

[1] LIMA, Julio. **Computacional Inteligence**. Disponível em: <https://github.com/juloko/computacional-inteligence>. Acesso em abril de 2018.