



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS SOBRAL

Curso de Engenharia Elétrica

Inteligência Computacional

Relatório 1

Aluno: Julio Cesar Ferreira Lima - 393849

Prof: Jarbas Joaci de Mesquita Sá Junior

Sobral, **Abril de 2018.**

1. Introdução

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 aprendido. 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:

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

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

Após alguns algebrismos para a determinação de “a”:

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

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

$$\begin{aligned}-2 \sum_{i=1}^n x_i (y_i - \bar{y} + b\bar{x} - bx_i) &= 0 \\ \sum_{i=1}^n [x_i (y_i - \bar{y}) + x_i b (\bar{x} - x_i)] &= 0 \\ \sum_{i=1}^n x_i (y_i - \bar{y}) + b \sum_{i=1}^n x_i (\bar{x} - x_i) &= 0 \\ b &= \frac{\sum_{i=1}^n x_i (y_i - \bar{y})}{\sum_{i=1}^n x_i (x_i - \bar{x})}\end{aligned}$$

Extrapolando a Regressão Linear Simples para a Regressão Múltipla, 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:

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

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.

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

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:

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

$$\frac{\partial S}{\partial b} = -2X'y + 2X'Xb = 0$$

$$X'Xb = X'y$$

$$b = (X'X)^{-1}X'y$$

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

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

Onde esta última equação é dada como o somatório da diferença da distância média entre a curva e a média desta ao quadrado visando apenas tomar posse de uma ordem desse valor de forma absoluta.

Assim como, faz-se necessário a tomada do somatório das diferenças entre a curva e o *fit*:

$$SQ_{\text{res}} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Por fim R^2 pode ser determinado por:

$$R^2 = \frac{SQ_{\text{exp}}}{SQ_{\text{tot}}} = 1 - \frac{SQ_{\text{res}}}{SQ_{\text{tot}}}$$

Por último, visando combater a negação do princípio da parcimônia por R^2 , que dá notoriedade a dados de com baixo índice de pertencimento a curva, aplica-se o método do R^2 ajustado:

$$\bar{R}^2 = 1 - \frac{n-1}{n-(k+1)} (1 - R^2)$$

Onde “n” é a quantidade de dados e “k” a ordem adotada.

2. Resultados

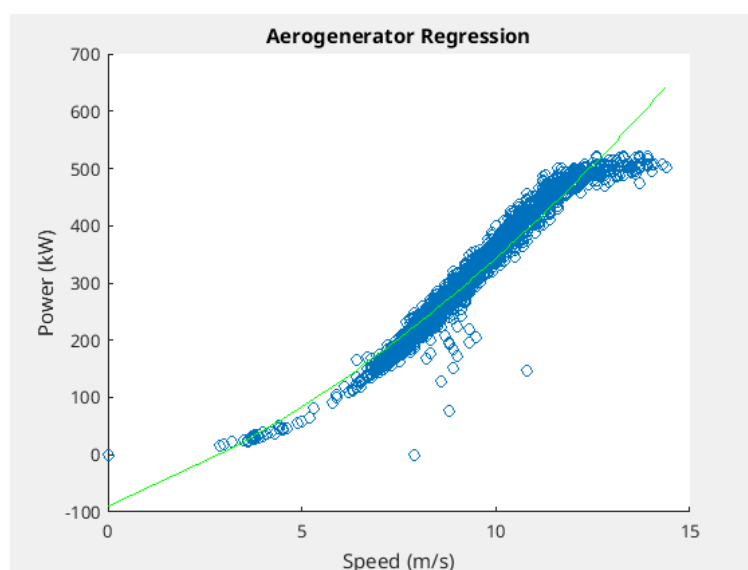
A base para todo o procedimento experimental desenvolvido nesse experimento/simulação está demonstrado na introdução. Portanto, de modo a não onerar este relatório, apenas será mostrado a saída do MATLAB seguido pela resposta em curva do gráfico para cada resposta do sistema.

OBS.: 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 torne-se útil a outrem.

Saída do MATLAB:

- Type the LAMBDA parameter, it has to in between $0 \leq \lambda \leq 1$, in real numbers: .5
- Type the ORDER of polynomial 2 a 5, in natural numbers: 1
- Type the ORDER of polynomial 2 a 5, in natural numbers: 2
- Betas parameters for polynomial regression:
- -88.8958
- 25.8804
- 1.7353
- Determination Coefficient:
- 0.9434
- Determination Coefficient Adjusted:
- 0.9434

Figura 1 – Aerogerador Grau 2.

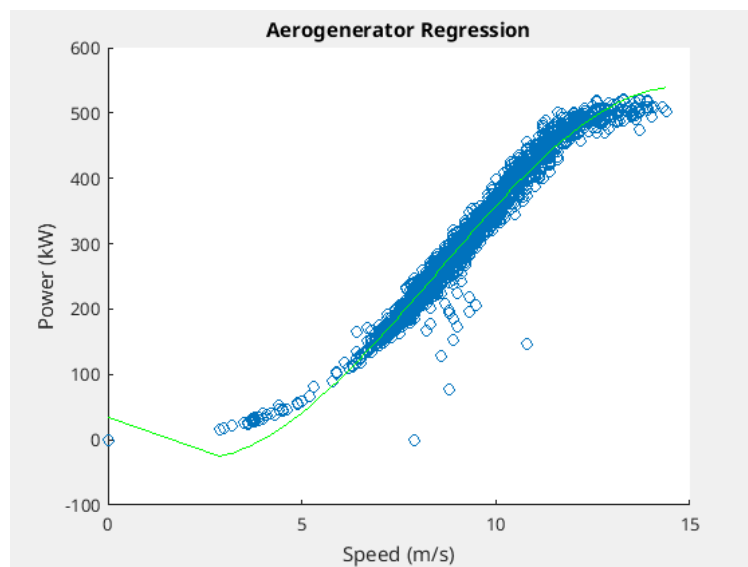


Fonte: Autor

Figura 2 – Aerogerador, Grau 3.

Saída do MATLAB:

- Type the LAMBDA parameter, it has to in between $0 \leq \lambda \leq 1$, in real numbers: .5
- Type the ORDER of polynomial 2 a 5, in natural numbers: 3
- Betas parameters for polynomial regression:
 - 33.8084
 - -59.1113
 - 15.0857
 - -0.5935
- Determination Coefficient:
 - 0.9690
- Determination Coefficient Adjusted:
 - 0.9690

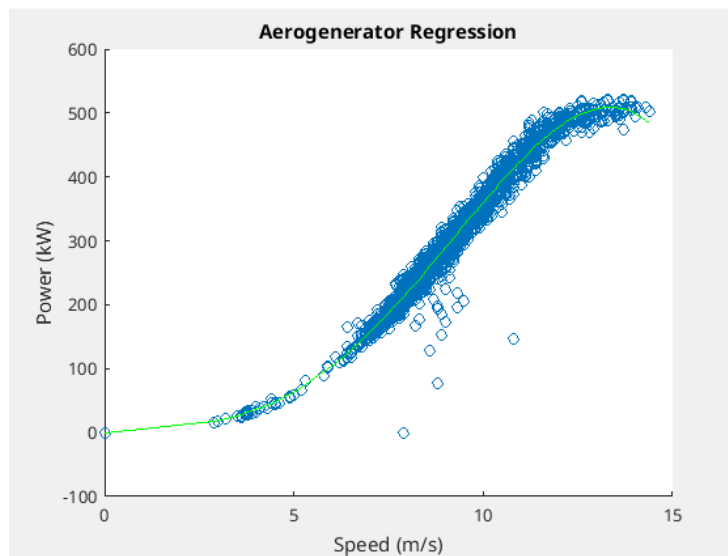


Fonte: Autor

Figura 3 – Aerogerador, Grau 4.

Saída do MATLAB:

- Type the LAMBDA parameter, it has to be in between $0 \leq \lambda \leq 1$, in real numbers: .5
- Type the ORDER of polynomial 2 a 5, in natural numbers: 4
- Betas parameters for polynomial regression:
 - -1.1040
 - 10.4536
 - -4.9820
 - 1.4301
 - -0.0675
- Determination Coefficient:
 - 0.9737
- Determination Coefficient Adjusted:
 - 0.9737

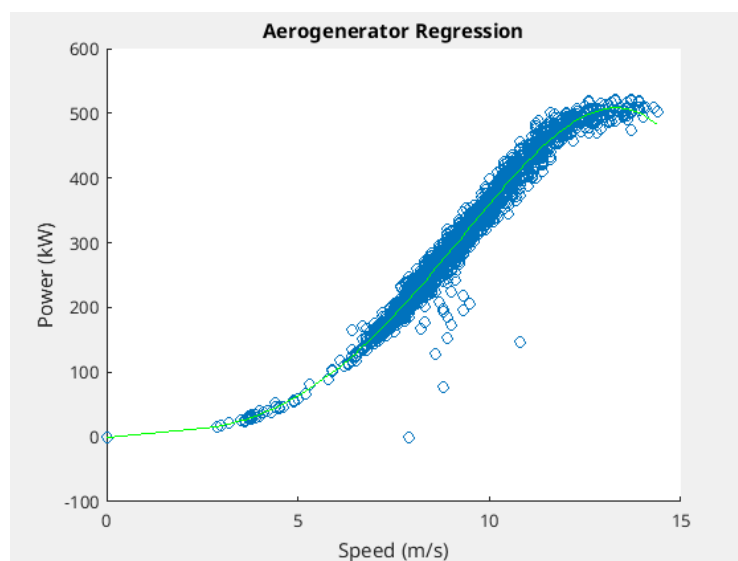


Fonte: Autor

Saída do MATLAB:

- Type the LAMBDA parameter, it has to be in between $0 \leq \lambda \leq 1$, in real numbers: .5
- Type the ORDER of polynomial 2 a 5, in natural numbers: 5
- Betas parameters for polynomial regression:
 - -0.7260
 - 7.3221
 - -3.4674
 - 1.1653
 - -0.0478
 - -0.0005
- Determination Coefficient:
 - 0.9737
- Determination Coefficient Adjusted:
 - 0.9737

Figura 4 – Aerogerador, Grau 5.

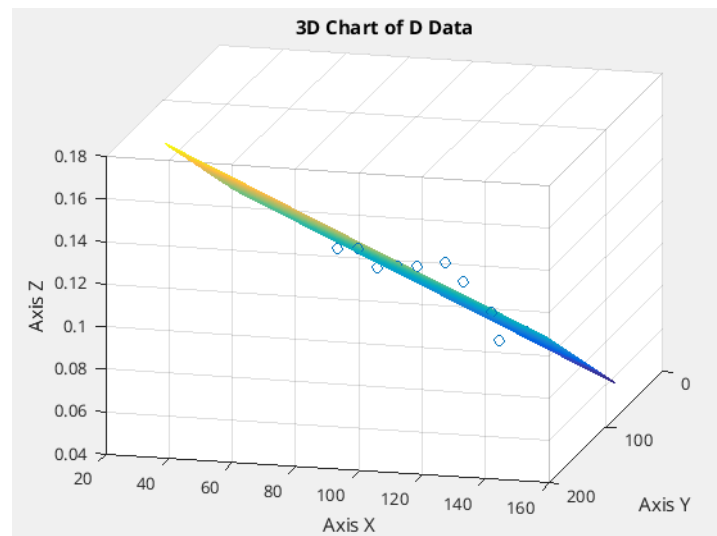


Fonte: Autor

Saída do MATLAB:

- Betas parameters for polynomial regression:
- 0.1274
- -0.0007
- 0.0004
- Determination Coefficient:
- 0.7239
- Determination Coefficient Adjusted:
- 0.6894

Figura 5 – Base de dados D, Grau 1, Dimensão 3.



Fonte: Autor

3. Conclusão

A teoria desenvolvida na determinação de aproximação de curvas e previsão de tendências baseadas em prévios aferidas com excelência nesse no experimento de simulação a que se dedicou esse relatório. A importância desse tipo de estudo pode ser tomada como exemplos bastante difundidos, como na determinação aproximada do limiar de temperatura de um determinado material a em temperaturas custosas, de modo que não fosse possível a determinação por experimento, mas pode ser tomada com boa precisão dado o comportamento do material em cenários plausíveis de reprodução. Assim como a construção de uma tendência de mercado após a análise histórica de uma determinada commodity. Similarmente ao que foi obtido com a curva de geração do aerogerador, utilizado nesse experimento.

Também foi possível demonstrar o possível *fit* de uma curva para uma variável de dimensão superior a 2D, por meio do segundo experimento, onde foi novamente possível constatar a acurácia da aproximação.

No primeiro experimento foi adotado em todos os casos um ajuste feito pela regularização de Tikhonov, no entanto após algumas variações nesse parâmetro, atingindo ambos limiares pouco ou nada conseguiu se extrair de qualidade em relação a essa regularização, para todos os casos então fora atribuído o valor médio do intervalo igual a 0,5. Essa regularização não fora utilizada para o segundo experimento.

4. Bibliografia

- [1] WIKIPEDEA. **Método dos mínimos quadrados**. Disponível em:
<https://pt.wikipedia.org/wiki/Método_dos_mínimos_quadrados>. Acesso em abril de 2018.
- [2] WIKIPEDEA. **R²**. Disponível em: <<https://pt.wikipedia.org/wiki/R%C2%B2>>. Acesso em abril de 2018.
- [3] JUNIOR, Jarbas. **Regressão Linear Múltipla** . Acesso em abril de 2018.
- [4] JUNIOR, Jarbas. **Regressão Simples** . Acesso em abril de 2018.

5. Repositório

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