Resoluções da Primeira Lista de Atividades da Disciplina de Aprendizado de Máquina

Natália Freitas Araújo[1]

[1]Programa de Pós-Graduação em Computação Aplicada (PPCA) – Núcleo de Desenvolvimento Amazônico em Engenharia (NDAE) - Universidade Federal do Pará (UFPA)
Tucuruí – PA – Brasil

taiaraujo20@gmail.com

Resumo. *O presente artigo apresenta resoluções e discursões referente a assuntos abordados, na primeira semana de aula, dentro da disciplina* Aprendizado de Máquina *(Machine Learning). Cada seção corresponde as atividades e as subseções aos itens da lista de exercício, em anexo a esse documento será enviado um arquivo compactado contendo os códigos desenvolvidos em Python.*

1. Atividade 01 – *Polynomial Curve Fitting*

Os primeiros passos na disciplina foram dados com as discussões sobre problemas de ajuste de curva (*Curve Fitting*) que têm como finalidade a construção de uma curva ou função matemática que melhor se ajusta a um determinado conjunto de dados de entrada, variáveis independentes (*x*), e saída, variáveis dependentes de *x*, nos quais a função geradora*, f(x),* entre eles é desconhecida. Com a obtenção da curva ajustada é possível auxiliar na visualização dos dados, assim como, inferir valores não descritos na variável de entrada, e por conseguinte, ajuda na extração de informações que os dados descrevem.

Existem diversos exemplos de *Curve Fitting*, contudo, o modelo que será discutido nesta seção será o de ajuste de curva utilizando regressão polinomial (*Polynomial Curve Fitting),* este ajuste é modelado através de um polinômio de enésimo grau em x, descrito pela equação 1. Com o intuito de estudar o comportamento do ajuste de curva polinomial, além dos dados de entrada e saída, será também conhecida a função geradora do problema, equação 2.

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{j=0}^{M} \boldsymbol{w}_j \boldsymbol{x}^j \tag{1}$$

$$\boldsymbol{t} = \sin(2\boldsymbol{\pi x}) + \text{ruído} \tag{2}$$

1.1. Dados de Treinamento

Os dados de entrada que serão utilizados neste trabalho são dados pré-determinados (apresentados abaixo em forma de vetor). Considerando estes dados e a equação 2 sem o ruído, têm-se a figura 1 para ilustrar a curva que originou os dados e como estes dados estão dispostos.

$\mathbf{x} \leftarrow [0.1387, 0.2691, 0.3077, 0.3625, 0.4756, 0.5039, 0.5607, 0.6468, 0.7490, 0.7881]$
$\mathbf{t} \leftarrow [0.8260, 1.0469, 0.7904, 0.6638, 0.1731, -0.0592, -0.2433, -0.6630, -1.0581, -0.8839]$

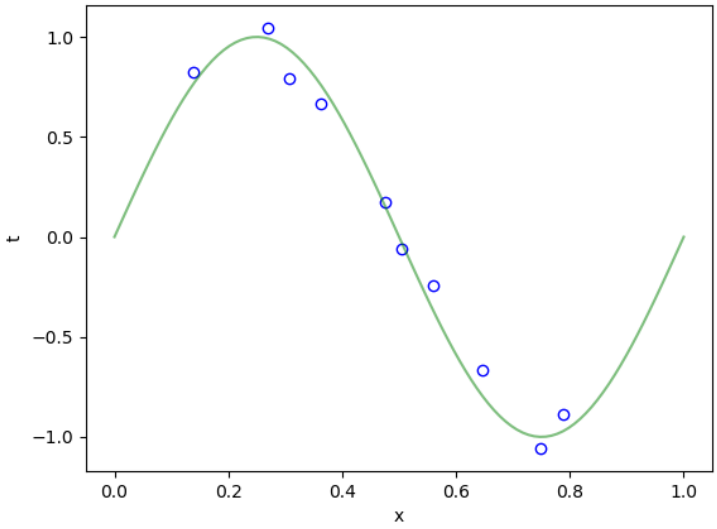

Figura 1. Gráfico com os pontos de entrada do vetor x em relação ao vetor de saída t em azul e a função 2 sem a presença do ruído em verde.

1.2. Discursão sobre Linearidade

O modelo polinomial dado pela equação 1 é dependente de duas variáveis, **w** e **x**, em relação a **w** é uma função linear e em relação a x é uma função não linear. A justificativa deste cenário se justifica pelo fato que a regressão polinomial se encaixa em uma relação não linear entre o valor de x e a média condicional correspondente de y.

Embora a regressão polinomial ajuste um modelo não linear aos dados, como um problema de estimativa estatística é linear, no sentido de que a função de regressão é linear nos parâmetros desconhecidos estimados a partir dos dados. Outro ponto de destaque é o parametro $\mathbf{w}_0$ permite qualquer deslocamento fixo nos dados, todas essas caracteristicas não empõe limitações significativas ao modelo resultante do ajuste.

1.3. Minimização da Função Erro

Considerando as ordens M = [0, 1, 3, 9] para a equação 1 é possível determinar os parâmetros livres, e o vetor **w,** que minimizam a função erro descrita na equação 3. Desta forma obtém-se quatro gráficos com os quatros modelos em relação as ordens M apresentados na figura 2.

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_{\mathrm{n}},\boldsymbol{w}) - \boldsymbol{t}_n\}^2 \tag{3}$$

Observa-se que quando M = 0 têm-se um traço constante que passa no meio dos pontos **x** e M = 1 têm-se uma reta linear decrescente, ambas tentam descrever o comportamento dos dados, porém fornecem ajustes insuficientes para a representação da função. O polinômio ordem M = 3 consegue ajustar um modelo muito próximo da função de origem, o melhor ajuste dos gráficos apresentados, esse feito está ligado diretamente ao grau do polinômio que permite um comportamento similar a equação 2 sem a presença do ruído. Todavia, o polinômio de ordem M = 9 apresenta um ajuste extremo dos pontos de **x,** e desta forma, ao invés de representar a função de origem o modelo busca descrever o comporta de cada ponto, torna-se assim uma modelagem ruim dos dados, esse fenômeno é conhecido como *over-fitting*.

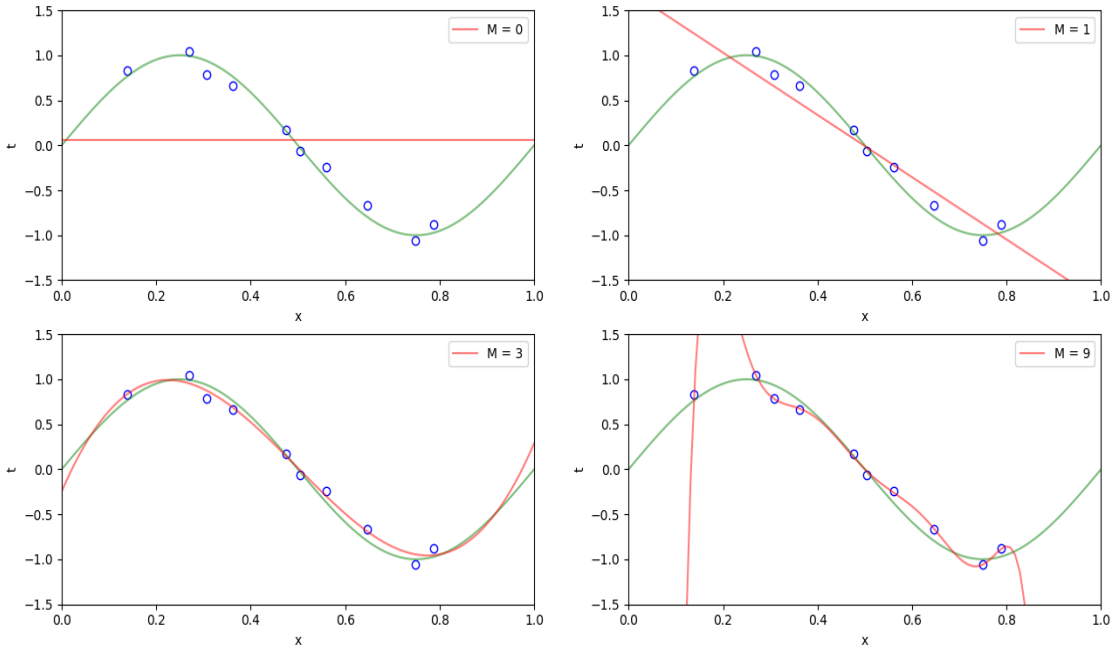

Figura 2. Ajustes de Curva da Equação 1 em relação a ordem M

1.4. Função Erro (E_{RMS})

O erro da raiz quadrada média, equação 4, permite comparar conjuntos de diferentes tamanhos de dados em uma mesma proporção, por esse motivo utiliza-se desta equação para analisar o desempenho do ajuste de curva para novos dados como apresentado na figura 3. A reta azul apresenta o cálculo da equação 4 para os dados de treinamento citados na seção 1.1 deste artigo e a reta vermelha apresenta o mesmo cálculo para os dados de testes descritos abaixo.

x_test ← [10 valores sorteados aleatoriamente pelo comando numpy random]
t_test ← [$\sin(2\pi \boldsymbol{x\_test}) + \text{ruído}$]

$$E_{RMS} = \sqrt{2E(w^*) / N} \tag{4}$$

O cálculo de erro para os dados de treinamento mostra que a cada acréscimo da ordem M a erro da raiz quadrada média apresenta um valor menor, chegando a $E_{RMS} = 0$ para M = 9. Porém, esse resultando não significa que é o melhor ajuste para este cenário, uma vez que quando o polinômio atinge grau nove o ajuste de curva apresenta *over-fitting,* como discutido na seção 1.3. O conjunto de teste entre o intervalo de M = 2 a M = 7 a função E_{RMS} de teve um decréscimo, apesar de não se aproximar de Zero o modelo tenta representar a função, já em M = 8 e M = 9 é perceptível que o erro tem um grande crescimento, isso se deve ao fato do polinômio nesses graus tentarem traçar os pontos de entrada

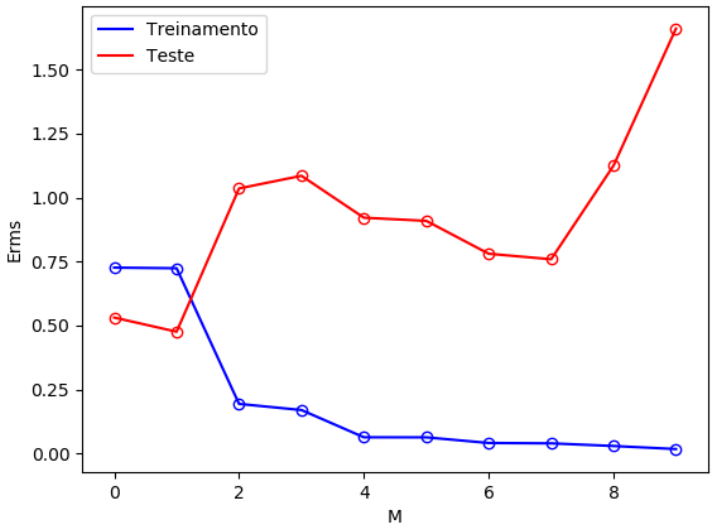

Figura 3. Cálculo E_{RMS} no conjunto de treinamento em azul de no conjunto de teste em vermelho

1.5. Tabela de Coeficientes de w

Como apresentado nas seções anteriores para cada grau do polinomio a função terá uma quantidade maior e equivalente de valores de **w.** Esse coeficiente é responsável por ponderá os valores de **x** de acordo como a função a qual está inserido. Na figura 4 têm-se os valores de w para as ordens M = [0, 1, 6, 9].

Examindo os valores obtidos vê-se que, à medida que a ordem M aumenta, a magnitude dos coeficientes de w acompanham esse crescimento. Destaca-se o polinômio de ordem M = 9, os coeficientes de w foram adequados com os dados por meio do desenvolvimento de grandes valores positivos e negativos, de modo que a função polinomial correspondente resulta exatamente a cada um dos pontos dos dados, mas entre os pontos dos dados a função exibe grandes oscilações. A justificativa para esse fato é que os polinômios mais flexíveis com valores maiores de M estão se tornando cada vez mais sintonizados com o ruído nos valores de entrada.

	M = 0	M = 1	M = 6	M = 9
0	0.05927	1.719962	-1.460417	4.451998e+02
1	0.00000	-3.458262	24.966523	-1.153040e+04
2	0.00000	0.000000	-46.503044	1.268016e+05
3	0.00000	0.000000	-203.485636	-7.762542e+05
4	0.00000	0.000000	858.080379	2.926097e+06
5	0.00000	0.000000	-1148.170949	-7.069723e+06
6	0.00000	0.000000	531.856471	1.098441e+07
7	0.00000	0.000000	0.000000	-1.061436e+07
8	0.00000	0.000000	0.000000	5.803690e+06
9	0.00000	0.000000	0.000000	-1.371347e+06

Figura 4. Coeficientes de w

2. Atividade 02 – Regularização

A regularização é uma técnica frequentemente usada para controlar o problema de *over-fitting*, ela funciona da seguinte maneira: adiciona-se um termo de penalidade à função de erro equação 3, com o propósito de "controlar" a magnitude dos coeficientes. O termo mais simples dessa penalidade assume a forma de uma soma dos quadrados de todos os coeficientes descrito na equação 5.

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{\mathrm{N}}\{y(x_{\mathrm{n}},\boldsymbol{w}) - \boldsymbol{t}_n\}^2 + \frac{\lambda}{2}\|\mathrm{w}\|^2 \tag{5}$$

2.1 Curvas com Parâmetro de Regularização λ

A Figura 5 mostra os resultados do polinômio da ordem M = 9 com a regularizações λ como mostrado na equação 5. O gráfico da esquerda está com o regularizado para λ = -18, e para esse cenário o excesso de ajuste foi suprimido e agora tê-se uma representação muito mais próxima da equação 2 sem a presença do ruido. O gráfico da direita está com regularização λ = 0 tem um resultado insatisfatorio, apesar de não haver o caso de *over-fitting* o modelo resultante não descreve a cursa de origem.

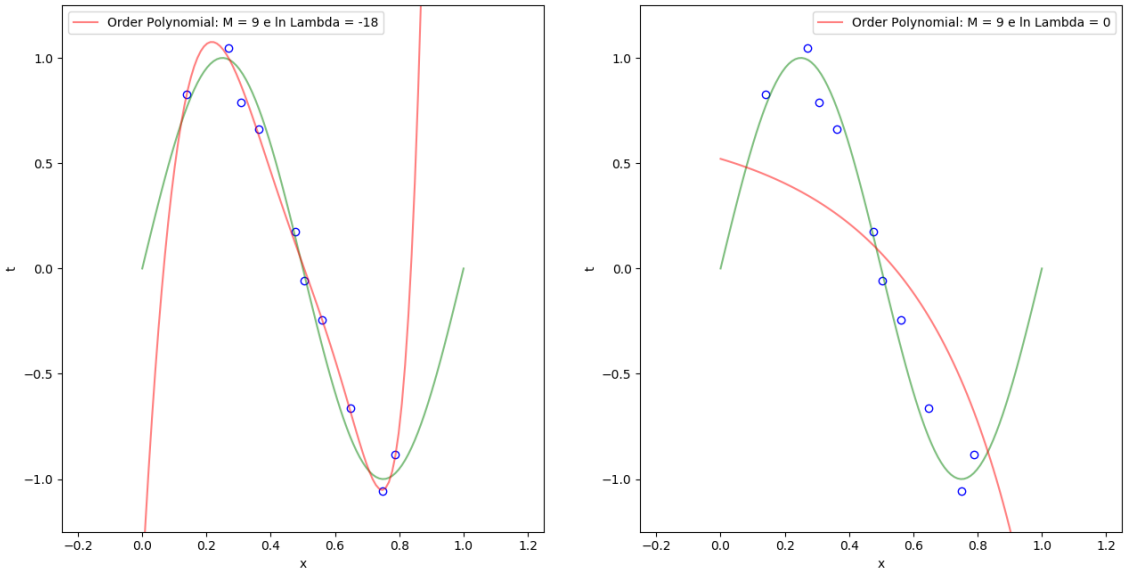

Figura 5. Gráficos com ordem M = 9 e Regularização

2.2 Tabela de Coeficientes de w

Os coeficientes correspondentes dos polinômios ajustados a equação 5 são apresentados na figura 6. Analisando os dados obtidos em comparação aos da figura 4 é possível identificar que quanto maior for o valor de λ, menor será o efeito da regularização em cima da equação e quanto menor for o valor de λ, maior o efeito de regularização será apresentado em cima do modelo gerado.

	ln lambda = -inf	ln lambda = -18	ln lambda = 0
0	4.451998e+02	-122.101180	0.520720
1	-1.153040e+04	2503.736992	-0.455178
2	1.268016e+05	-20210.290078	-0.533141
3	-7.762542e+05	84155.094813	-0.438690
4	2.926097e+06	-189632.904906	-0.332196
5	-7.069723e+06	197833.883103	-0.245561
6	1.098441e+07	31680.717296	-0.180605
7	-1.061436e+07	-304106.314464	-0.133068
8	5.803690e+06	290804.121597	-0.098445
9	-1.371347e+06	-93171.197159	-0.073170

Figura 6. Coeficientes de w para a equação 5

2.3 Função Erro (ERMS)

Analisando os resultados obtidos no erro da raiz quadrada média apresentam uma taxa muito baixo, próximo de zero, tanto para os valores de treinamento quanto para os valores de teste. Neste caso o coeficiente que controla a complexidade do modelo e determina o grau de ajuste excessivo é o parametro λ. Para a obtenção dos dados de teste foi realizado o mesmo processo descrito na seção 1.4.

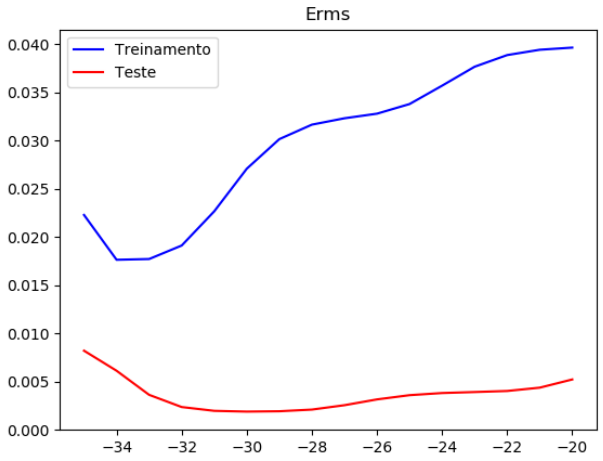

Figura 7. Função Erro com Regularização

2.4 A Maldição da Dimensionalidade

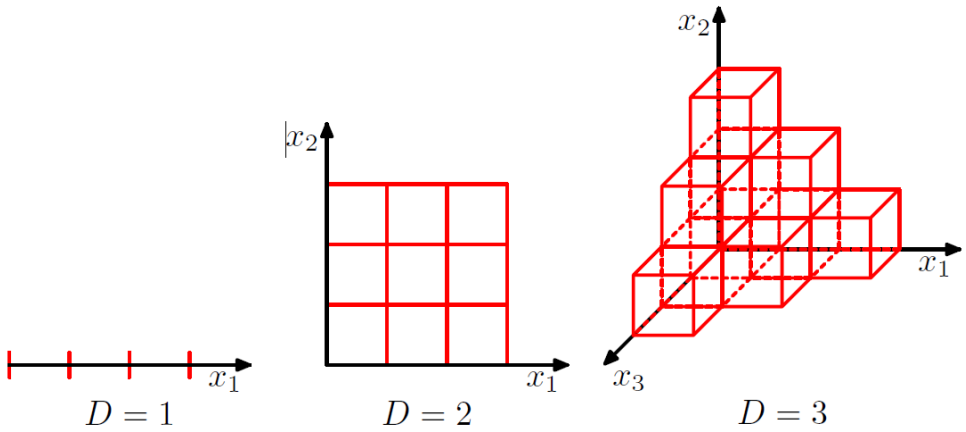

Figura 8. A Maldição da Dimensionalidade – Fonte: Pattern Recognition and Machine Learning

A figura 8 (BISHOP, 2006) mostra as divisões de uma região de espaços de uma, duas e três dimensões em células regulares. Observa-se que as células crescem exponencialmente de acordo com essas divisões. No cenário em que é apresentado este artigo, considera-se essas células como os atributos (os dados utilizados).

A grande questão é que com um número exponencialmente grande de atributos será necessária uma quantidade de dados de treinamento para garantir que não haja valores vazios. Outro reflexo desse efeito é aumento na complexidade do cálculo da equação 1, mostrado na figura 9 (BISHOP, 2006) que apresenta a equação 1 para D = 3 (três dimensões), e por conseguinte, há um aumento de complexidade em todas as etapas posteriores para o ajuste de curva.

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D}\sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D}\sum_{j=1}^{D}\sum_{k=1}^{D} w_{ijk} x_i x_j x_k. \quad (1.74)$$

Figura 9. A Maldição da Dimensionalidade Equação 1.74 – Fonte: Pattern Recognition and Machine Learning

3. Atividade 03 – *Linear Models for Regression*

A regressão polinomial (equação 1) é um exemplo particular do modelo de regressão linear, no qual existe uma única variável de entrada x, e as funções básicas assumem a forma de potências de x. Todavia, este modelo de regressão apresenta uma limitação, pois trata-se de funções globais da variável de entrada, de modo que as alterações em uma região do espaço de entrada afetam todas as outras regiões.

Uma maneira de intermediar esta questão é dividir o espaço de entrada em regiões e ajustá-las a um polinômio diferente em cada região equação 6. Para aplicar essa teoria será utilizada o problema e os dados de treinamento da seção 1 com duas diferentes funções base: a Gaussiana e a Sigmoide.

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M-1} \boldsymbol{w}_j \varphi_j(\boldsymbol{x}) \quad (6)$$

3.1. Função Base do Tipo Gaussiana

Para utilizar a função de base Gaussiana substitui-se o $\varphi_j(x)$ na equação 6 pela equação 7.

$$\varphi_j(\boldsymbol{x}) = \boldsymbol{e}\left\{-\frac{(\boldsymbol{x} - \boldsymbol{\mu}_j)^2}{2s^2}\right\} \quad (7)$$

Na equação 7 o μj governa os locais das funções básicas no espaço de entrada e o parâmetro *s* governa sua escala espacial, para a geração desses valores foi realizado os comandos abaixo.

μj ← [função linspace do numpy, com o intervalo entre x_0 e x_n com M valores]

μj ← [função var do numpy, cálculo da variância em t]

Observa-se que quando M = 0 e M = 1 têm um comportamento semelhante aos apresentados na seção 1, pelo grau do polinômio ser insuficiente para a representação da função. O polinômio ordem M = 3 consegue ajustar um modelo muito próximo da função de origem, melhor do que o apresentado na seção 1, esse feito está ligado não somente ao grau do polinômio, mas também, ao ajuste da equação Gaussiana. Deferentemente da seção 1, o polinômio de ordem M = 9 apresenta uma curvatura análoga a da função original em relação ao seu domínio, porém na imagem tem um deslocamento que está ligado ao parâmetro s da equação 7. Assim como na seção 1 o melhor modelo encontrado é da ordem M = 3.

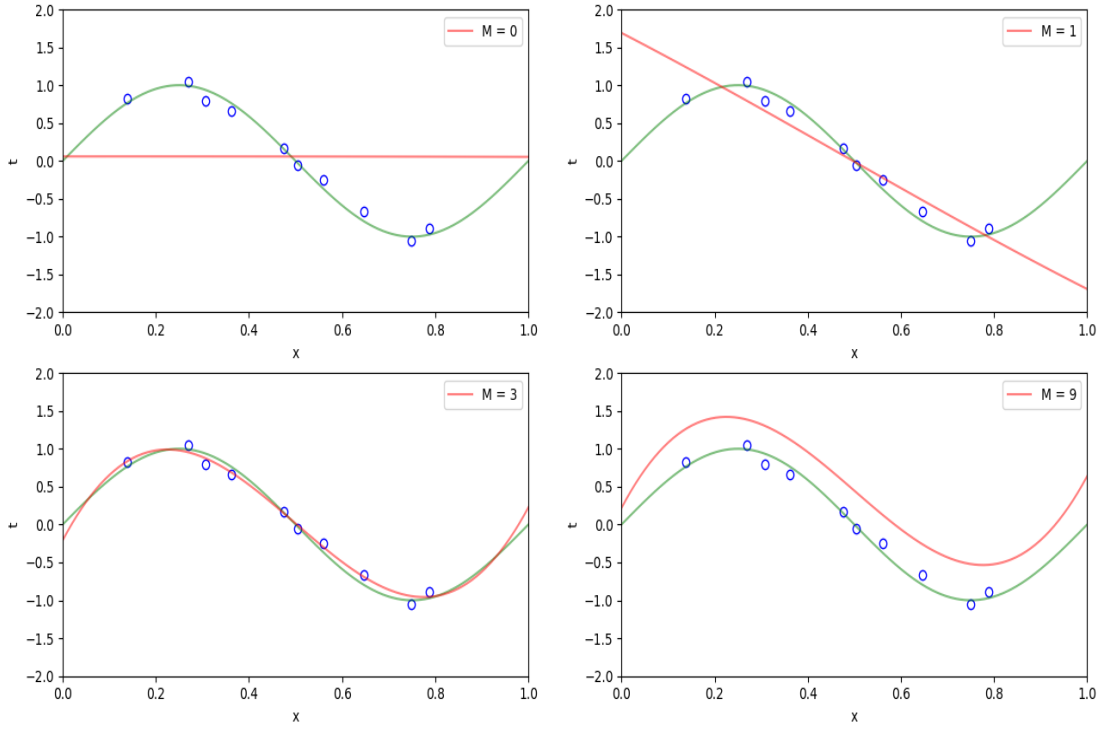

Figura 10. Gráficos Para Função de Base Gaussiana

A figura 11 apresenta os resultados do cálculo de erro para os dados de treinamento e teste. Os dados de treinamento mostra que a cada acréscimo da ordem M a erro da raiz quadrada média apresenta um valor menor de M = 0 a M = 5, após essa ordem acréscimo do valor do erro e quando M = 9 volta ao padrão de M = 5. O conjunto de teste, apesar de ter valores diferentes, apresenta um comportamento semelhante ao conjunto de treino descrito acima.

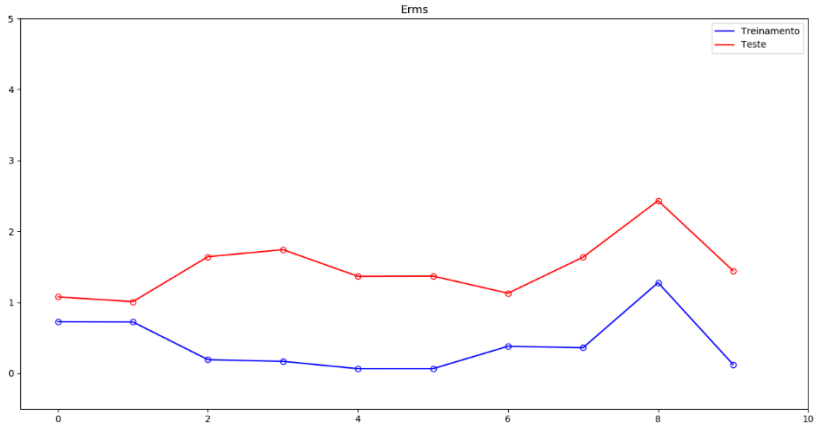

Figura 11. Cálculo do Erro (RMS) Para Função Base Gaussiana

Na figura 12 têm-se os valores de w para as ordens M = [0, 1, 6, 9]. Examindo os valores obtidos vê-se que, à medida que a ordem M aumenta, a magnitude dos coeficientes de w acompanham esse crescimento. Comparando com a figura 4 têm-se valores descritos aqui são muito mais significativos do que os coeficientes de w para a função polinomial, fato este coerente visto que os modelos gerados pela função Gaussiana não sofre de *over-fitting* e consegue realizar um melhor ajuste de curva.

Tabela Atividade 3.1

	M = 0	M = 1	M = 6	M = 9
0	0.05927	19.954878	-152096.193802	-97675.671273
1	0.00000	-19.834813	241168.107206	16737.603640
2	0.00000	0.000000	-91699.074932	-15784.374634
3	0.00000	0.000000	83991.571409	133867.649651
4	0.00000	0.000000	147533.912800	189074.476394
5	0.00000	0.000000	-481926.419788	137046.367116
6	0.00000	0.000000	253024.176814	-390084.460505
7	0.00000	0.000000	0.000000	-224219.233580
8	0.00000	0.000000	0.000000	165017.624287
9	0.00000	0.000000	0.000000	86014.337225

Figura 12. Coeficientes de w Para a Função Base Gaussiana

3.2. Função Base do Tipo Sigmoide

Para utilizar a função de base Sigmoide substitui-se o $\varphi_j(x)$ na equação 6 pela equação 8.

$$\varphi_j(x) = \frac{1}{1 + e\left\{-\frac{x - \mu_j}{s}\right\}} \quad (8)$$

Na equação 8 o μj e o *s* têm a mesma função que possuem na equação 7. E os dados de teste são obtidos da mesma maneira que descrito na seção 3.1.

Nota-se que quando M = 0 têm um comportamento semelhando a ordem M = 0 da seção 1. Entretanto, a ordem M = 1 apresenta um comportamento totalmente diferente as demais funções bases apresentadas até aqui. Esse modelo possui uma leve curvatura em uma tentativa de descrever a função. O polinômio ordem M = 3 consegue ajustar um modelo muito próximo da função de origem, assim como de outras funções bases, porém, visualmente, o ajuste do modelo Gaussiano se mostra superior para este caso. O polinômio de ordem M = 9 apresenta um ajuste que tenta acompanhar os pontos, mas o modelo não apresenta um bom ajuste.

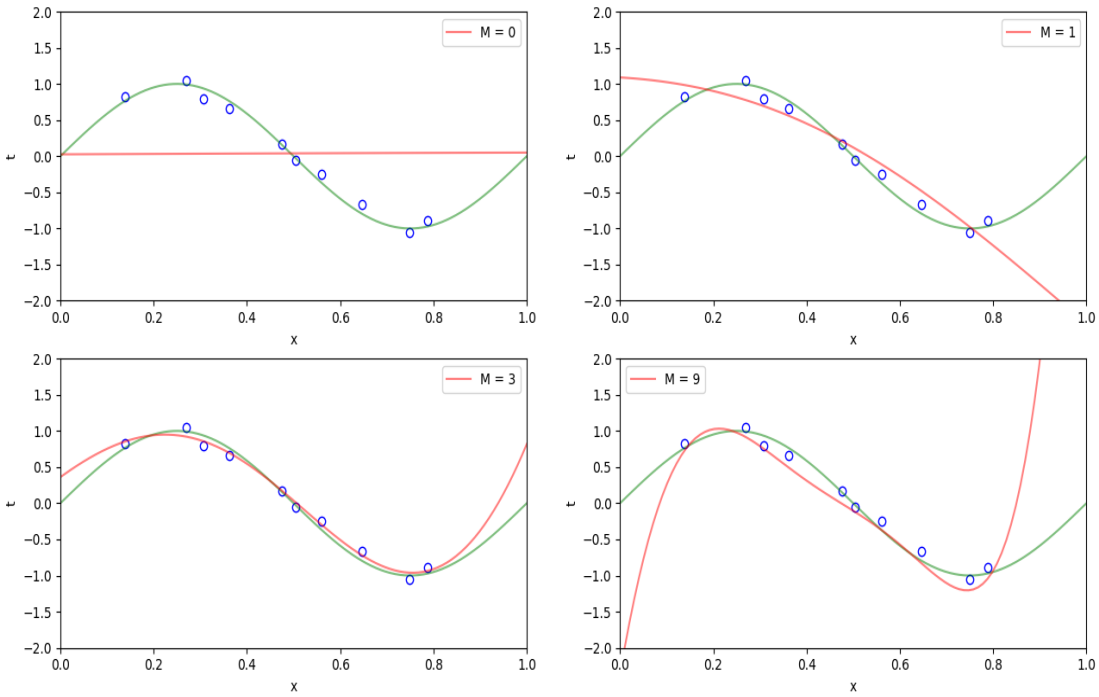

Figura 13. Gráficos Para a Função Base Sigmoide

O cálculo de erro para os dados de treinamento mostra que a cada acréscimo da ordem M a erro da raiz quadrada média apresenta um valor menor de M = 0 a M = 5, após essa ordem acréscimo do valor do erro e quando M = 9 volta ao padrão de M = 5. O conjunto de teste, apesar de ter valores diferentes, apresenta um comportamento semelhante ao conjunto de treino descrito acima, esses dados são mostrados na figura 14.

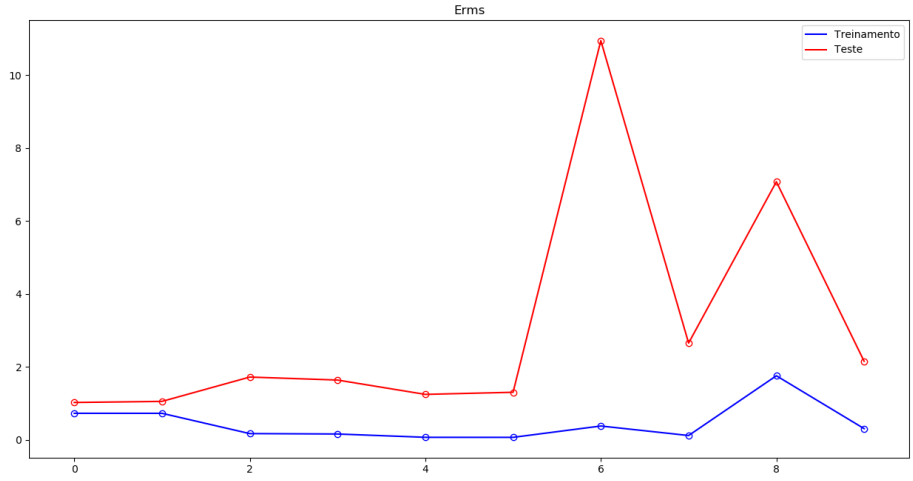

Figura 14. Erro (RMS) da Função Base Sigmoide

Na figura 15 têm-se os valores de w para as ordens M = [0, 1, 6, 9]. Examindo os valores obtidos vê-se que, à medida que a ordem M aumenta, a magnitude dos coeficientes de w acompanham esse crescimento. Comparando com a figura 4 têm-se valores descritos aqui têm um peso um pouco maior do que os coeficientes de w para a função polinomial, este "peso" é o responsavel pelo melhor ajuste e por não ter *over-fitting*.

Tabela Atividade 3.2

	M = 0	M = 1	M = 6	M = 9
0	0.05927	9.963135	-5.262463e+05	-8.509677e+05
1	0.00000	-17.797299	2.091753e+06	2.463023e+06
2	0.00000	0.000000	-2.512600e+06	-1.867367e+06
3	0.00000	0.000000	-6.839628e+05	8.447072e+05
4	0.00000	0.000000	4.184928e+06	-3.507702e+06
5	0.00000	0.000000	-3.589157e+06	4.258747e+06
6	0.00000	0.000000	1.036924e+06	-7.848033e+05
7	0.00000	0.000000	0.000000e+00	2.224131e+05
8	0.00000	0.000000	0.000000e+00	-1.652474e+06
9	0.00000	0.000000	0.000000e+00	8.756832e+05

Figura 15. Coeficientes de w Para a Função Base Sigmoide

3.3. O Problema de Ajuste de Curva - Tipos de Regularização

Na Seção 2 discutiu-se superficialmente a idéia de adicionar um termo de regularização a uma função de erro para controlar o ajuste excessivo, utilizando a regularização da soma dos quadrado equação 5. Essa escolha específica de regularizador é conhecida na literatura de aprendizado de máquina como decaimento de peso, porque em algoritmos de aprendizado sequencial, ele incentiva os valores de peso a decair em direção a zero, a menos que sejam suportados pelos dados.

Outro tipo de regularização utiliza os multiplicadores de Lagrange que permite que modelos complexos sejam treinados em conjuntos de dados de tamanho limitado sem excesso de ajuste, essencialmente limitando a complexidade efetiva do modelo. Uma das aplicações dessa regularização é utilizada em algoritmos *SVMs* que trabalha com otimizações quadraticas. No entanto, o problema de determinar a complexidade ideal do modelo é então deslocado de encontrar o número apropriado de funções básicas para determinar o valor adequado do coeficiente de regularização.

4. Atividade 04 – *Linear Models for Classification*

Dentro dos modelos lineares de classificação têm-se o LDA *Linear Discriminant Analysis* este metodo é uma generalização do discriminante linear de Fisher, muito utilizado para problemas de reconhecimento de padrões e na área de aprendizado de máquina para encontrar combinações lineares de recursos que caracteriza ou separa duas ou mais classes de objetos ou eventos.

4.1 LDA Para Duas Classes

A primeira etapa para o LDA é calcular a média dos valores dos atributos dos dados de acordo com a equação 9. Após esse passo, calcula-se a matriz de covariância dentro da classe nomeada S_w dado pela equação 10. Por fim, calcula-se o vetor w dado pela equação 11.

$$m = \frac{1}{N} \sum_{n \in c} x_n \tag{9}$$

$$S_w = \sum_{m \in k} \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T \tag{10}$$

$$w \propto S_w^{-1} (m_2 - m_1) \tag{11}$$

Aplicando as equações têm-se o resultado os gráficos da figura 16, os quais estão dispostos por dois atributos *Sepal Length* e o *Sepal Width.*

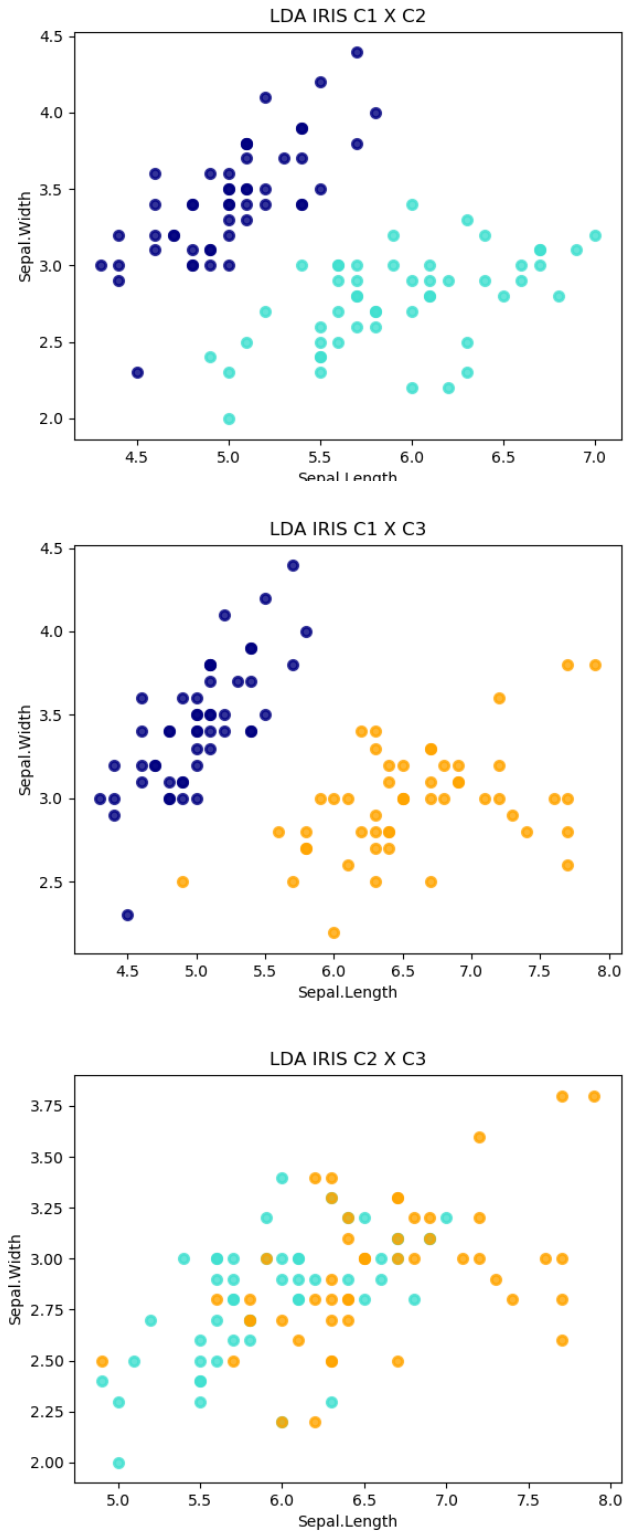

Figura 16. Gráficos Dataset Iris

Referências

Bishop, Christopher. (2006) “Pattern Recognition and Machine Learning”. Springer.