



UNIVERSIDADE  
FEDERAL DO CEARÁ



## Aprendizagem de Máquina

César Lincoln Cavalcante Mattos

2020

# Agenda

- ① Regressão polinomial
- ② Generalização
- ③ Regularização
- ④ Seleção de hiperparâmetros e avaliação de modelos
- ⑤ Normalização dos dados
- ⑥ Tópicos adicionais
- ⑦ Referências

# Regressão polinomial

- Considere a tabela a seguir relacionando alturas de jovens pacientes e comprimentos do cateter correspondente:

<b>Altura (m)</b>	<b>Comprimento (cm)</b>
1.087	37
1.613	50
0.953	34
1.003	36
1.156	43
0.978	28
1.092	37
0.572	20
0.940	34
0.597	30
0.838	38
1.473	47

# Regressão polinomial

- Considere a tabela a seguir relacionando alturas de jovens pacientes e comprimentos do cateter correspondente:

Altura (m)	Comprimento (cm)
1.087	37
1.613	50
0.953	34
1.003	36
1.156	43
0.978	28
1.092	37
0.572	20
0.940	34
0.597	30
0.838	38
1.473	47

- **Problema:** Podemos criar atributos a partir do atributo já existente (**Altura**)?

# Regressão polinomial

- Podemos criar novos atributos através de **transformações não-lineares** de atributos já existentes:

Altura	Altura <sup>2</sup>	Comprimento
1.087	1.087 <sup>2</sup>	37
1.613	1.613 <sup>2</sup>	50
0.953	0.953 <sup>2</sup>	34
1.003	1.003 <sup>2</sup>	36
1.156	1.156 <sup>2</sup>	43
0.978	0.978 <sup>2</sup>	28
1.092	1.092 <sup>2</sup>	37
0.572	0.572 <sup>2</sup>	20
0.940	0.940 <sup>2</sup>	34
0.597	0.597 <sup>2</sup>	30
0.838	0.838 <sup>2</sup>	38
1.473	1.473 <sup>2</sup>	47

# Regressão polinomial

- Podemos criar novos atributos através de **transformações não-lineares** de atributos já existentes:

Altura	Altura <sup>2</sup>	Comprimento
1.087	1.087 <sup>2</sup>	37
1.613	1.613 <sup>2</sup>	50
0.953	0.953 <sup>2</sup>	34
1.003	1.003 <sup>2</sup>	36
1.156	1.156 <sup>2</sup>	43
0.978	0.978 <sup>2</sup>	28
1.092	1.092 <sup>2</sup>	37
0.572	0.572 <sup>2</sup>	20
0.940	0.940 <sup>2</sup>	34
0.597	0.597 <sup>2</sup>	30
0.838	0.838 <sup>2</sup>	38
1.473	1.473 <sup>2</sup>	47

- Podemos obter um modelo **não-linear nos dados** mas **linear nos parâmetros**:

# Regressão polinomial

- De maneira geral, para um polinômio de ordem  $P$ :

$$\hat{y}_i = \mathbf{w}^\top \mathbf{x}_i,$$

$$\mathbf{x}_i = [1 \ x_i \ x_i^2 \ \cdots \ x_i^P]^\top,$$

$$\mathbf{w} = [w_0 \ w_1 \ w_2 \ \cdots \ w_P]^\top.$$

- Na forma matricial, temos:

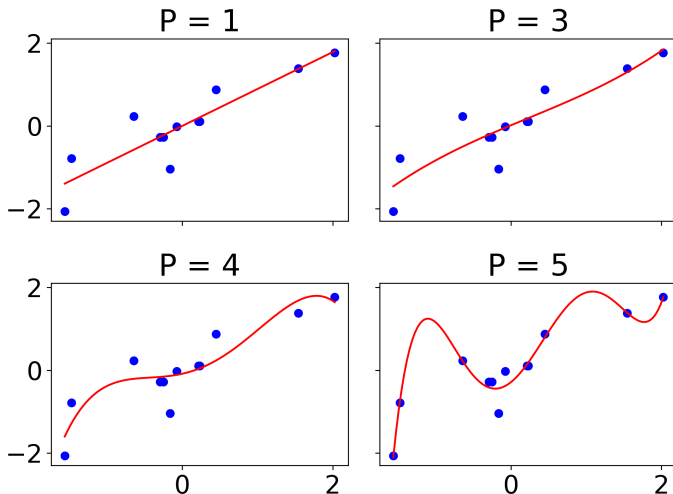
$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w},$$

$$\hat{\mathbf{y}} = [\hat{y}_1 \ \hat{y}_2 \ \hat{y}_3 \ \cdots \ \hat{y}_N]^\top,$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \mathbf{x}_3^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix}.$$

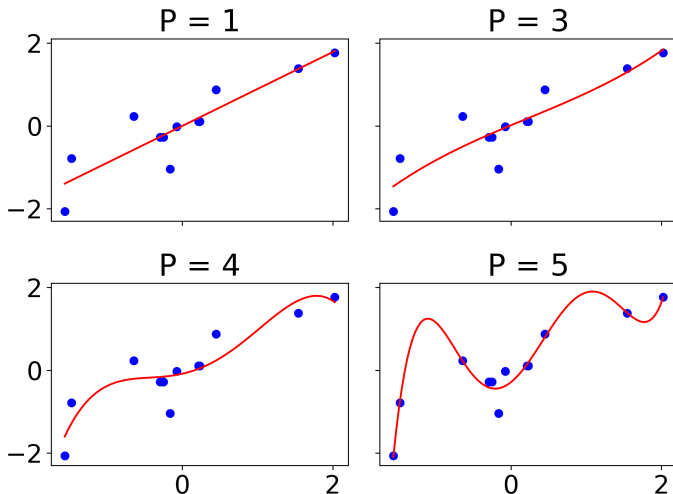
- Os parâmetros  $\mathbf{w}$  podem ser estimados via GD, SGD ou OLS.

# Regressão polinomial





# Regressão polinomial



- **Problema:** Como escolher a ordem do polinômio?

# Agenda

- 1 Regressão polinomial
- 2 Generalização
- 3 Regularização
- 4 Seleção de hiperparâmetros e avaliação de modelos
- 5 Normalização dos dados
- 6 Tópicos adicionais
- 7 Referências

# Generalização

- Estamos interessados na capacidade do modelo em prever corretamente dados não utilizados para ajustar seus parâmetros.

# Generalização

- Estamos interessados na capacidade do modelo em prever corretamente dados não utilizados para ajustar seus parâmetros.
- Procedimento básico:
  - ① Separe os dados em dois conjuntos diferentes: **treinamento** e **teste**;
  - ② Ajuste (**treine**) os parâmetros do modelo usando somente o conjunto de treinamento;
  - ③ Verifique a capacidade de **generalização** do modelo no conjunto de teste.

# Generalização

## Overfitting (sobreajuste)

- Ocorre quando o modelo se ajusta demasiadamente aos dados usados para encontrar seus parâmetros.

# Generalização

## Overfitting (sobreajuste)

- Ocorre quando o modelo se ajusta demasiadamente aos dados usados para encontrar seus parâmetros.

## Underfitting (subajuste)

- Ocorre quando o modelo não possui expressividade suficiente para se ajustar aos dados disponíveis.

# Generalização

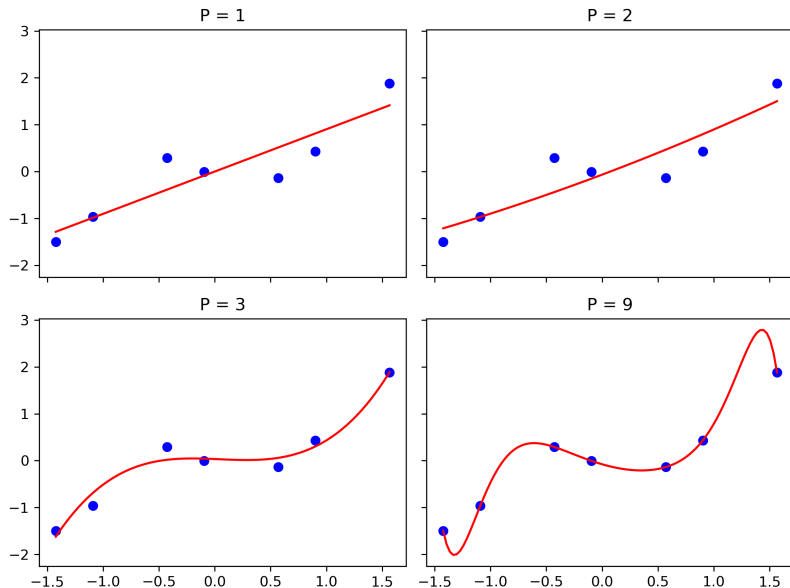
## Overfitting (sobreajuste)

- Ocorre quando o modelo se ajusta demasiadamente aos dados usados para encontrar seus parâmetros.

## Underfitting (subajuste)

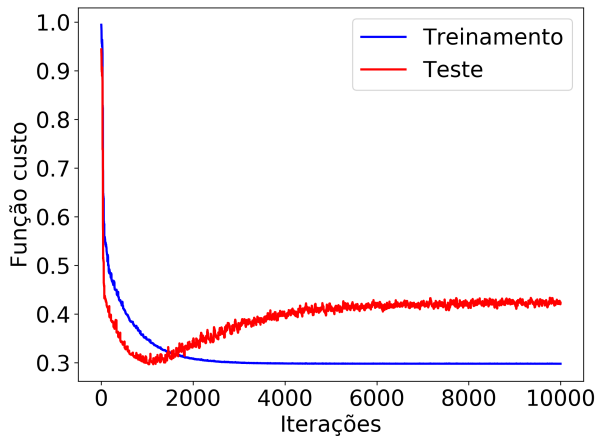
- Ocorre quando o modelo não possui expressividade suficiente para se ajustar aos dados disponíveis.
- Em ambos os cenários o modelo pode apresentar dificuldade de **generalização**.

# Generalização





# Generalização



Curva de aprendizagem em um cenário de overfitting.

# Generalização

- Polinômios de **ordem baixa** podem resultar em **underfitting**.
  - O modelo é **pouco flexível/expressivo**.

# Generalização

- Polinômios de **ordem baixa** podem resultar em **underfitting**.
  - O modelo é **pouco flexível/expressivo**.
- Polinômios de **ordem alta** podem resultar em **overfitting**.
  - O modelo é **muito flexível/expressivo**.

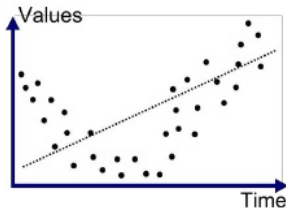
# Generalização

- Polinômios de **ordem baixa** podem resultar em **underfitting**.
  - O modelo é **pouco flexível/expressivo**.
- Polinômios de **ordem alta** podem resultar em **overfitting**.
  - O modelo é **muito flexível/expressivo**.

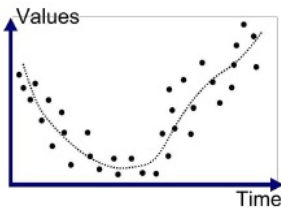
## Dilema viés-variância

- **Underfitting** apresenta alto viés e baixa variância.
- **Overfitting** apresenta baixo viés e alta variância.
- O ideal é que viés e variância sejam equilibrados pelo modelo.

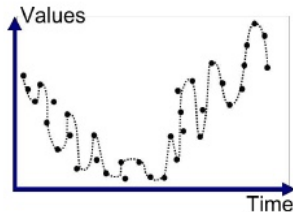
# Generalização



Alto viés  
Baixa variância



Médio viés  
Média variância



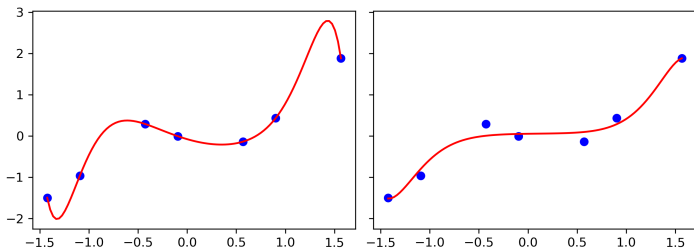
Baixo viés  
Alta variância

# Agenda

- ① Regressão polinomial
- ② Generalização
- ③ Regularização**
- ④ Seleção de hiperparâmetros e avaliação de modelos
- ⑤ Normalização dos dados
- ⑥ Tópicos adicionais
- ⑦ Referências

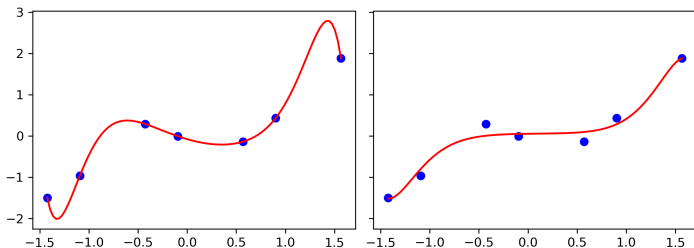
# Regularização

- Modelos com **parâmetros grandes** (em módulo), tendem a **generalizar menos**.
  - Modelo com overfitting  $\rightarrow \|\hat{w}\|^2 = 2.372582$
- Modelos com **parâmetros pequenos** (em módulo), tendem a **generalizar mais**.
  - Modelo adequado  $\rightarrow \|\hat{w}\|^2 = 0.1221954$



# Regularização

- Modelos com **parâmetros grandes** (em módulo), tendem a **generalizar menos**.
  - Modelo com overfitting  $\rightarrow \|\hat{\mathbf{w}}\|^2 = 2.372582$
- Modelos com **parâmetros pequenos** (em módulo), tendem a **generalizar mais**.
  - Modelo adequado  $\rightarrow \|\hat{\mathbf{w}}\|^2 = 0.1221954$



- **Ideia:** Como evitar o **overfitting** e melhorar a **generalização** mesmo em modelos flexíveis/expressivos?



# Regularização

## Regularização L2

- Adiciona à função custo um termo proporcional à norma quadrática dos parâmetros:

$$\mathcal{J}(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2,$$

$$\|\mathbf{w}\|^2 = \mathbf{w}^\top \mathbf{w} = \sum_{d=1}^D w_d^2.$$

- O termo de regularização induz parâmetros menores.
- $\lambda > 0$  é um **hiperparâmetro de regularização**.
- O parâmetro  $w_0$  não é regularizado.
- Também é chamada de **ridge regression** ou **weight decay**.

# Regularização

## Regularização L2

- Pode ser aplicada aos métodos de otimização estudados:

- **Gradiente descendente (GD) regularizado:**

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \frac{\alpha}{N} \left[ \sum_{i=1}^N e_i(t-1) \mathbf{x}_i - \lambda \mathbf{w}(t-1) \right].$$

- **Gradiente descendente estocástico (SGD) regularizado:**

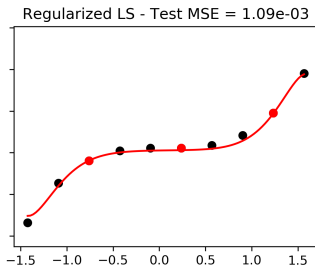
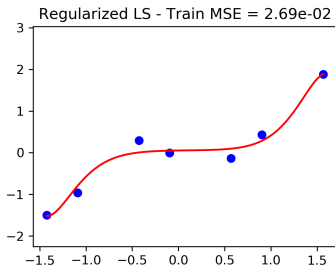
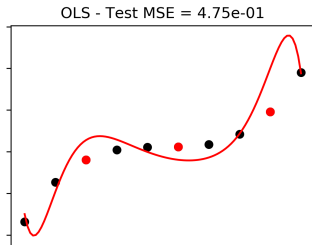
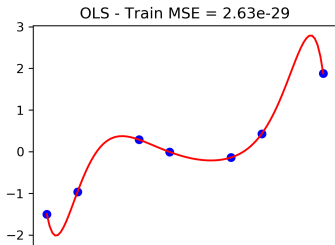
$$\mathbf{w}(t) = \mathbf{w}(t-1) + \alpha [e_i(t-1) \mathbf{x}_i - \lambda \mathbf{w}(t-1)].$$

- **Mínimos quadrados regularizado:**

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}.$$

- Lembre-se que o parâmetro  $w_0$  não deve ser regularizado.

# Regressão polinomial ( $P = 9$ )



Dados de treinamento: pontos **azuis** (com ruído) e **pretos** (sem ruído).  
Dados de teste: pontos **vermelhos**.

# Regularização L2

- De onde vem o termo de regularização  $\frac{\lambda}{2} \|w\|^2$ ?

# Regularização L2

- De onde vem o termo de regularização  $\frac{\lambda}{2} \|\mathbf{w}\|^2$ ?
- Lembre que temos uma verossimilhança Gaussiana:

$$p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^N \mathcal{N}(y_i | \mathbf{w}^\top \mathbf{x}_i, \sigma^2)$$

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \sum_{i=1}^N \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(y_i - \mathbf{w}^\top \mathbf{x}_i)^2}{2\sigma^2} \right)$$

$$\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) = -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

# Regularização L2

- Em vez de maximizar  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})$  (solução ML), encontramos o vetor  $\mathbf{w}$  que maximiza  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$  (solução MAP).

# Regularização L2

- Em vez de maximizar  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})$  (solução ML), encontramos o vetor  $\mathbf{w}$  que maximiza  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$  (solução MAP).
- Escolhendo  $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma_w^2 \mathbf{I})$ , o termo adicional será:

$$\log p(\mathbf{w}) = \log \left[ \frac{1}{(2\pi\sigma_w^2)^{D_w/2}} \exp \left( -\frac{1}{2\sigma_w^2} \mathbf{w}^\top \mathbf{w} \right) \right]$$

$$\log p(\mathbf{w}) = -\frac{D_w}{2} \log(2\pi\sigma_w^2) - \frac{1}{2\sigma_w^2} \mathbf{w}^\top \mathbf{w}.$$

# Regularização L2

- Em vez de maximizar  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})$  (solução ML), encontramos o vetor  $\mathbf{w}$  que maximiza  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$  (solução MAP).
- Escolhendo  $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma_w^2 \mathbf{I})$ , o termo adicional será:

$$\log p(\mathbf{w}) = \log \left[ \frac{1}{(2\pi\sigma_w^2)^{D_w/2}} \exp \left( -\frac{1}{2\sigma_w^2} \mathbf{w}^\top \mathbf{w} \right) \right]$$

$$\log p(\mathbf{w}) = -\frac{D_w}{2} \log(2\pi\sigma_w^2) - \frac{1}{2\sigma_w^2} \mathbf{w}^\top \mathbf{w}.$$

- Ignorando o termo constante e fazendo  $\lambda = \frac{1}{\sigma_w^2}$  e  $\mathbf{w}^\top \mathbf{w} = \|\mathbf{w}\|^2$ , recuperamos o termo  $\frac{\lambda}{2} \|\mathbf{w}\|^2$  da regularização L2.



# Regularização L2

- Em vez de maximizar  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})$  (solução ML), encontramos o vetor  $\mathbf{w}$  que maximiza  $\log p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})$  (solução MAP).
- Escolhendo  $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma_w^2 \mathbf{I})$ , o termo adicional será:

$$\log p(\mathbf{w}) = \log \left[ \frac{1}{(2\pi\sigma_w^2)^{D_w/2}} \exp \left( -\frac{1}{2\sigma_w^2} \mathbf{w}^\top \mathbf{w} \right) \right]$$
$$\log p(\mathbf{w}) = -\frac{D_w}{2} \log(2\pi\sigma_w^2) - \frac{1}{2\sigma_w^2} \mathbf{w}^\top \mathbf{w}.$$

- Ignorando o termo constante e fazendo  $\lambda = \frac{1}{\sigma_w^2}$  e  $\mathbf{w}^\top \mathbf{w} = \|\mathbf{w}\|^2$ , recuperamos o termo  $\frac{\lambda}{2} \|\mathbf{w}\|^2$  da regularização L2.
- **Observação:** A regularização L2 equivale a uma **priori Gaussiana** para os parâmetros e uma **solução MAP**.

# Agenda

- 1 Regressão polinomial
- 2 Generalização
- 3 Regularização
- 4 Seleção de hiperparâmetros e avaliação de modelos
- 5 Normalização dos dados
- 6 Tópicos adicionais
- 7 Referências

# Seleção de hiperparâmetros e avaliação de modelos

- Qual a diferença entre um **parâmetro** e um **hiperparâmetro**?

# Seleção de hiperparâmetros e avaliação de modelos

- Qual a diferença entre um **parâmetro** e um **hiperparâmetro**?
- Como escolher o hiperparâmetro de regularização  $\lambda$ ?

# Seleção de hiperparâmetros e avaliação de modelos

- Qual a diferença entre um **parâmetro** e um **hiperparâmetro**?
- Como escolher o hiperparâmetro de regularização  $\lambda$ ?

## Validação cruzada (hold-out validation)

- Separe um terceiro conjunto de dados de **validação** para ajuste dos parâmetros.
- O conjunto de validação pode ser usado para estimar a **generalização** do modelo antes de usar o conjunto de teste.
- O modelo final deve ser obtido a partir dos conjuntos de treinamento e validação.
- O conjunto de teste **não** deve ser usado até a obtenção do modelo final.

# Seleção de hiperparâmetros e avaliação de modelos

- Qual a diferença entre um **parâmetro** e um **hiperparâmetro**?
- Como escolher o hiperparâmetro de regularização  $\lambda$ ?

## Validação cruzada (hold-out validation)

- Separe um terceiro conjunto de dados de **validação** para ajuste dos parâmetros.
  - O conjunto de validação pode ser usado para estimar a **generalização** do modelo antes de usar o conjunto de teste.
  - O modelo final deve ser obtido a partir dos conjuntos de treinamento e validação.
  - O conjunto de teste **não** deve ser usado até a obtenção do modelo final.
- 
- Validação cruzada pode ser usada para escolher o hiperparâmetro de regularização  $\lambda$ , a ordem do polinômio  $P$  e/ou o passo de aprendizagem  $\alpha$ .

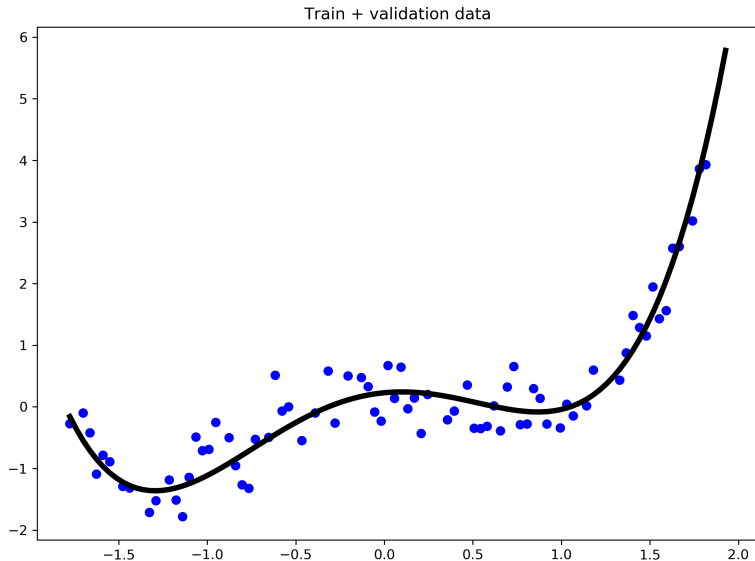
# Seleção de hiperparâmetros

## Grid search

- 1 Separe os dados em 3 conjuntos: **treinamento**, **validação** e **teste**;
- 2 Construa uma lista de hiperparâmetros candidatos;
- 3 Treine o modelo para o primeiro hiperparâmetro candidato;
- 4 Verifique a qualidade do modelo obtido no conjunto de validação;
- 5 Repita os dois passos anteriores para os demais candidatos;
- 6 Escolha o hiperparâmetro com melhor avaliação no conjunto de validação;
- 7 Retreine o modelo usando o conjunto de treinamento e de validação;
- 8 Verifique a generalização do modelo no conjunto de teste.

# Seleção de hiperparâmetros

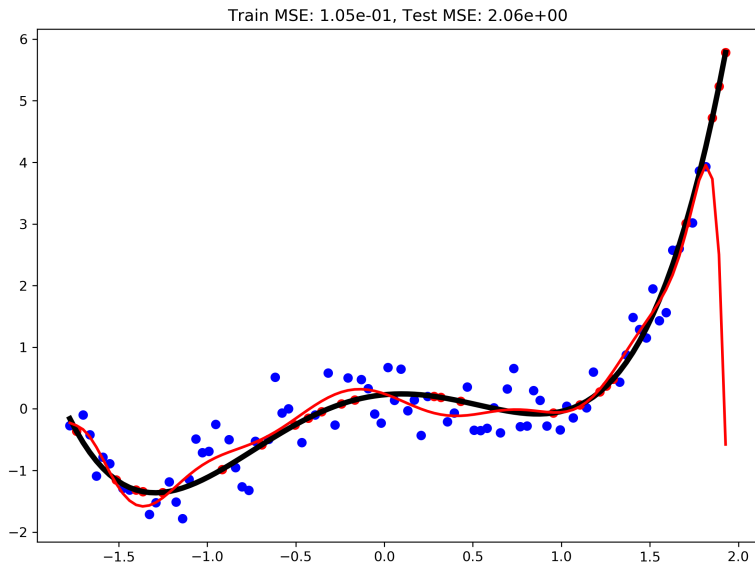
Dados de treinamento+validação em azul





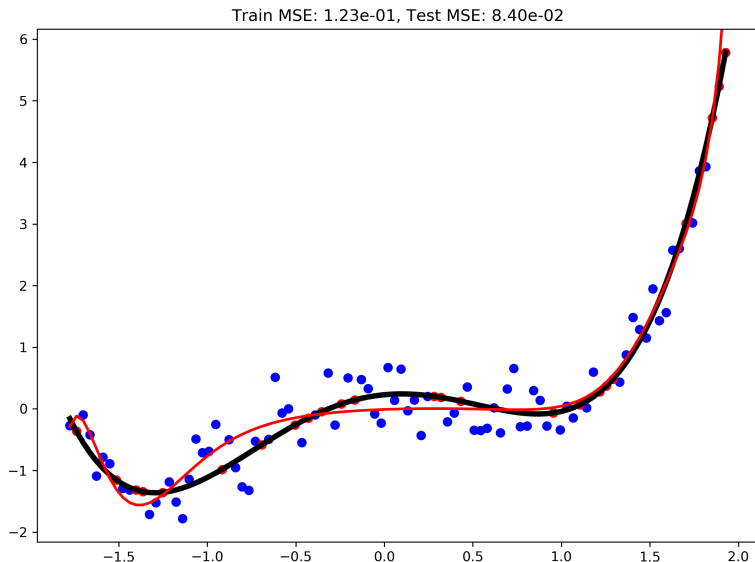
# Seleção de hiperparâmetros

Regressão polinomial ( $P = 15$ )



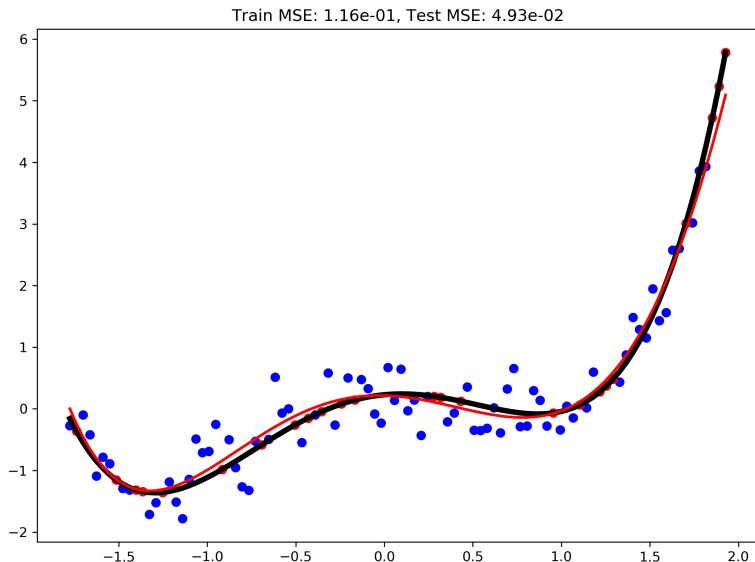
# Seleção de hiperparâmetros

Regressão polinomial ( $P = 15$ ) com grid search para  $\lambda = 7.196857$



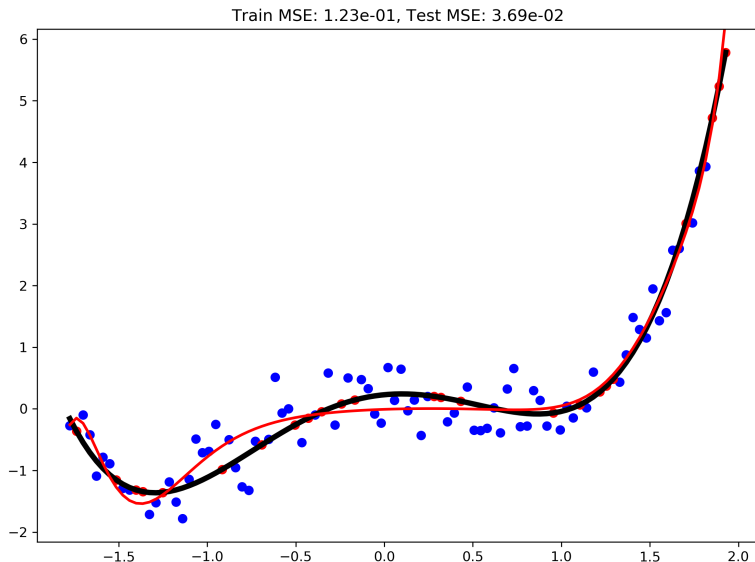
# Seleção de hiperparâmetros

Regressão polinomial com grid search para  $P = 5$



# Seleção de hiperparâmetros

Regressão polinomial com grid search para  $P = 13$  e  $\lambda = 7.196857$



# Seleção de hiperparâmetros e avaliação de modelos

## K-fold cross-validation

- ① Divida o conjunto de treinamento aleatoriamente em  $K$  partições iguais;
- ② 1 das partições é mantida para teste, enquanto as  $K - 1$  demais são usadas para treinar o modelo;
- ③ Repita o passo anterior para cada uma das partições;
- ④ Retorne o resultado médio obtido para cada partição.

# Seleção de hiperparâmetros e avaliação de modelos

## K-fold cross-validation

- 1 Divida o conjunto de treinamento aleatoriamente em  $K$  partições iguais;
- 2 1 das partições é mantida para teste, enquanto as  $K - 1$  demais são usadas para treinar o modelo;
- 3 Repita o passo anterior para cada uma das partições;
- 4 Retorne o resultado médio obtido para cada partição.

## Leave-one-out (LOO)

- O mesmo que K-fold cross-validation quando  $K = N$ .

# K-fold cross-validation



# Agenda

- 1 Regressão polinomial
- 2 Generalização
- 3 Regularização
- 4 Seleção de hiperparâmetros e avaliação de modelos
- 5 Normalização dos dados**
- 6 Tópicos adicionais
- 7 Referências



# Normalização dos dados

- Os dados disponíveis podem estar em escalas muito diferentes.
  - Idade, altura, peso, salário, etc.
- Usualmente é recomendado escalar os dados (de entrada e saída) para uma escala comum.
- Podemos normalizar entre os intervalos  $[0, 1]$  ou  $[-1, 1]$  ou forçar média 0 e variância unitária.
- **Vantagens:** Maior controle dos valores dos parâmetros; maior facilidade em ajustar os hiperparâmetros.
- **Importante:** Os dados de validação/teste devem ser atualizados de acordo com as estatísticas dos dados de treinamento.

# Normalização dos dados

## Exemplo de normalização dos dados via z-score

- ① Calcule a média do vetor  $\mathbf{y}$  e da matriz  $\mathbf{X}$ , coluna a coluna:

$$\mu_y = \mathbb{E}[\mathbf{y}], \quad [\mu_x]_d = \mathbb{E}[[\mathbf{X}]_d], \forall d.$$

- ② Retire a média dos dados (as operações seguem o Numpy):

$$\mathbf{y} \leftarrow \mathbf{y} - \mu_y, \quad [\mathbf{X}]_d \leftarrow [\mathbf{X}]_d - [\mu_x]_d, \forall d.$$

- ③ Calcule o desvio padrão de  $\mathbf{y}$  e da matriz  $\mathbf{X}$ , coluna a coluna:

$$s_y = \sqrt{\mathbb{V}[\mathbf{y}]}, \quad [s_x]_d = \sqrt{\mathbb{V}[[\mathbf{X}]_d]}, \forall d.$$

- ④ Conclua a normalização dos dados:

$$\mathbf{y} \leftarrow \frac{\mathbf{y}}{s_y}, \quad [\mathbf{X}]_d \leftarrow \frac{[\mathbf{X}]_d}{[s_x]_d}, \forall d.$$

# Normalização dos dados

- Antes de calcular o erro de teste, é interessante “desnormalizar” as predições e usar os dados de teste não-normalizados.
- No caso de normalização via z-score teríamos (as operações seguem o Numpy):

$$\hat{\mathbf{y}} \leftarrow (\hat{\mathbf{y}} \times s_y) + \mu_y.$$

- **Importante:** Utilize as médias e desvios computados com os dados de treinamento para fazer a “desnormalização”.

# Agenda

- ① Regressão polinomial
- ② Generalização
- ③ Regularização
- ④ Seleção de hiperparâmetros e avaliação de modelos
- ⑤ Normalização dos dados
- ⑥ Tópicos adicionais
- ⑦ Referências

# Tópicos adicionais

- MAP  $\times$  regularização.
- Regularização L1 (least absolute shrinkage and selection operator - lasso).
- Regularização via elastic net (L1 + L2).
- Esparsidade via regularização.
- Métodos robustos a outliers (valores discrepantes).
- Double descent e generalização em modelos sobreparametrizados.

# Agenda

- 1 Regressão polinomial
- 2 Generalização
- 3 Regularização
- 4 Seleção de hiperparâmetros e avaliação de modelos
- 5 Normalização dos dados
- 6 Tópicos adicionais
- 7 Referências**

# Referências bibliográficas

- **Caps. 7 e 13\*** - MURPHY, Kevin P. **Machine learning: a probabilistic perspective**, 2012.
- **Cap. 9** - DEISENROTH, M. *et al.* **Mathematics for machine learning**. 2019.