APRENDIZAGEM DE MÁQUINA

PROF. JOSENALDE OLIVEIRA

josenalde.oliveira@ufrn.br
https://github.com/josenalde/machinelearning

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - UFRN

MODELOS LINEARES DE REGRESSÃO

Machine Learning SUPERVISIONADA: estimar modelos que, embora simplificações da realidade, apresentem a melhor aderência possível entre os valores **reais (observados)** e os valores preditos

Esta é a equação geral da Regressão Linear Múltipla, com n features. Se considerarmos apenas uma, o objetivo é encontrar uma equação que apresente a relação entre uma variável dependente (target) e uma variável explicativa (feature)

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n, \ \text{RLM}$$

$$\hat{y} = \theta_0 + \theta_1 x_1, \ \text{RLS, onde } \theta_0 : \ \text{viés, intercepto, coef. lineear}$$

$$x_i : \ \text{é a i-ésima feature e } \theta_j \text{ é o j-ésimo parâmetro do modelo}$$

Cada parâmetro do modelo (com exceção do intercepto) na verdade reflete o PESO de determinada feature, então vamos adotar uma notação mais comum em aprendizado de máquina, o peso como sendo a letra W

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = \mathbf{W}.\mathbf{x} = \mathbf{W}^{\mathbf{T}}\mathbf{x}$$

MODELOS LINEARES DE REGRESSÃO

Notação

Medidas de desempenho comum em Regressão

Scikit-Learn.metrics – root_mean_squared_error (scikit-learn >= 1.4)

$$\text{Raiz do Erro Médio Quadrático: RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\mathbf{W}^{\mathbf{T}}\mathbf{x}^{(\mathbf{i})} - \mathbf{y}^{(\mathbf{i})}\right)^2}$$

Peso maior para grandes erros

$$\text{Erro Médio Absoluto: MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|\mathbf{W}^{\mathbf{T}}\mathbf{x}^{(\mathbf{i})} - \mathbf{y}^{(\mathbf{i})}\right|$$

Indicado para bases com outliers...

m: número de instâncias no conjunto de dados

$\mathbf{x}^{(\mathbf{i})}$ é um vetor de todos os valores das features (excluindo o rótulo) da $\mathbf{i}-\mathbf{esima}$ instância

$y^{(i)}$ é seu rótulo, valor desejado da saída para aquela instância

$$\mathbf{x}^{(\mathbf{1})} = \begin{pmatrix} -118.20 \\ 33.91 \\ 1416 \\ 38372 \end{pmatrix} \quad y^{(1)} = 156400 \qquad \mathbf{X} = \begin{pmatrix} (\mathbf{x}^{(\mathbf{1})})^{\mathbf{T}} \\ (\mathbf{x}^{(\mathbf{2})})^{\mathbf{T}} \\ \vdots \\ (\mathbf{x}^{(\mathbf{m})})^{\mathbf{T}} \end{pmatrix} = \begin{pmatrix} -118.29 & 33.91 & 1416 & 38372 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

Exemplo: longitude, latitude, n. de habitantes, renda média anual com saída valor médio da moradia

MODELOS LINEARES DE REGRESSÃO

Notação

Medidas de desempenho comum em Regressão

O R-quadrado é uma medida estatística também conhecida como **coeficiente de determinação,** que serve para avaliar a existência de uma relação útil entre a variável dependente(Y) e as variáveis independentes (Xi) em um modelo de regressão.

Ele representa a proporção da variância de Y que foi explicada pelas variáveis independentes no modelo.

Ele fornece uma indicação da qualidade do ajuste e, portanto, uma medida de quão bem as amostras não vistas podem ser previstas pelo modelo, por meio da proporção da variância explicada.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

O R-quadrado está sempre entre 0 e 1:

* 0 indica que o modelo não explica nada da variabilidade dos dados de resposta ao redor de sua média.

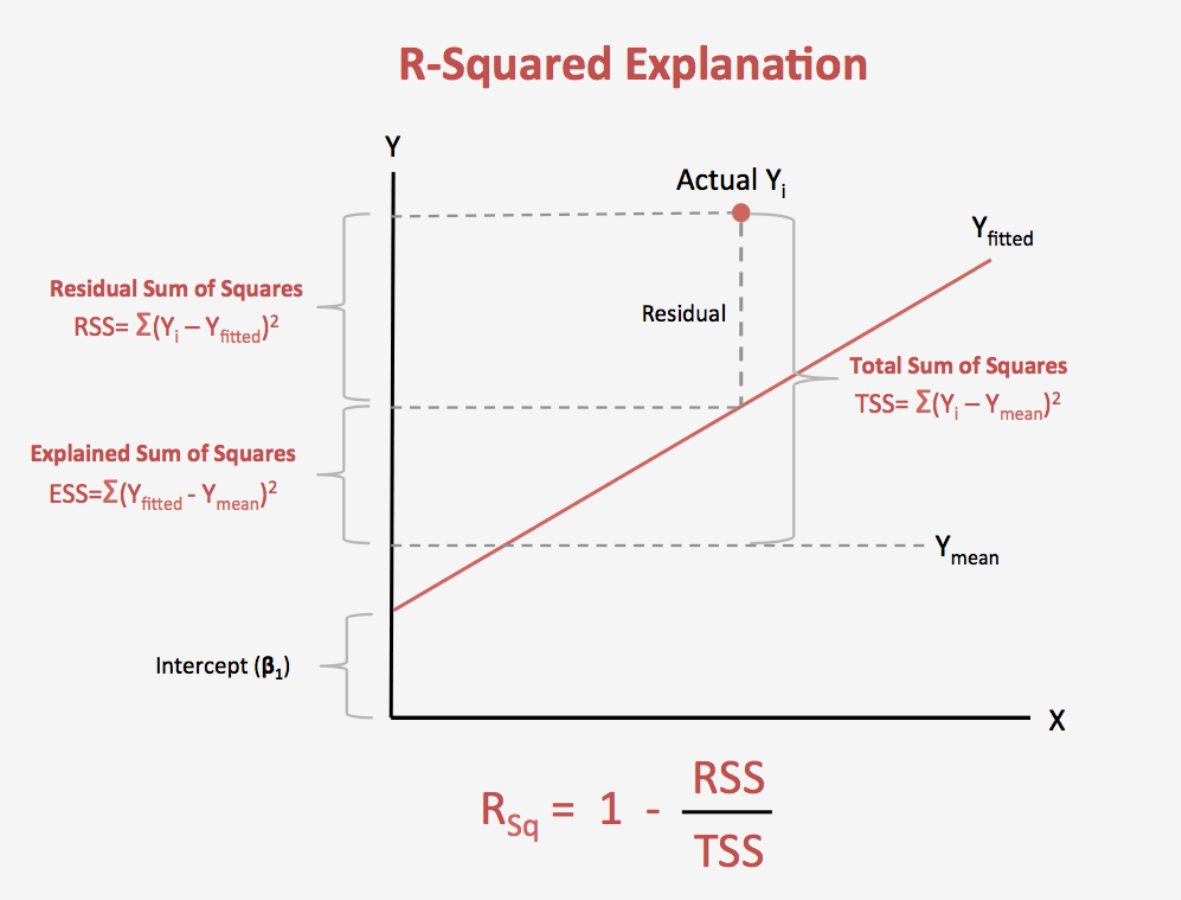
* 1 indica que o modelo explica toda a variabilidade dos dados de resposta ao redor de sua média.

Em geral, quanto maior o R-quadrado, melhor o modelo se ajusta aos seus dados.

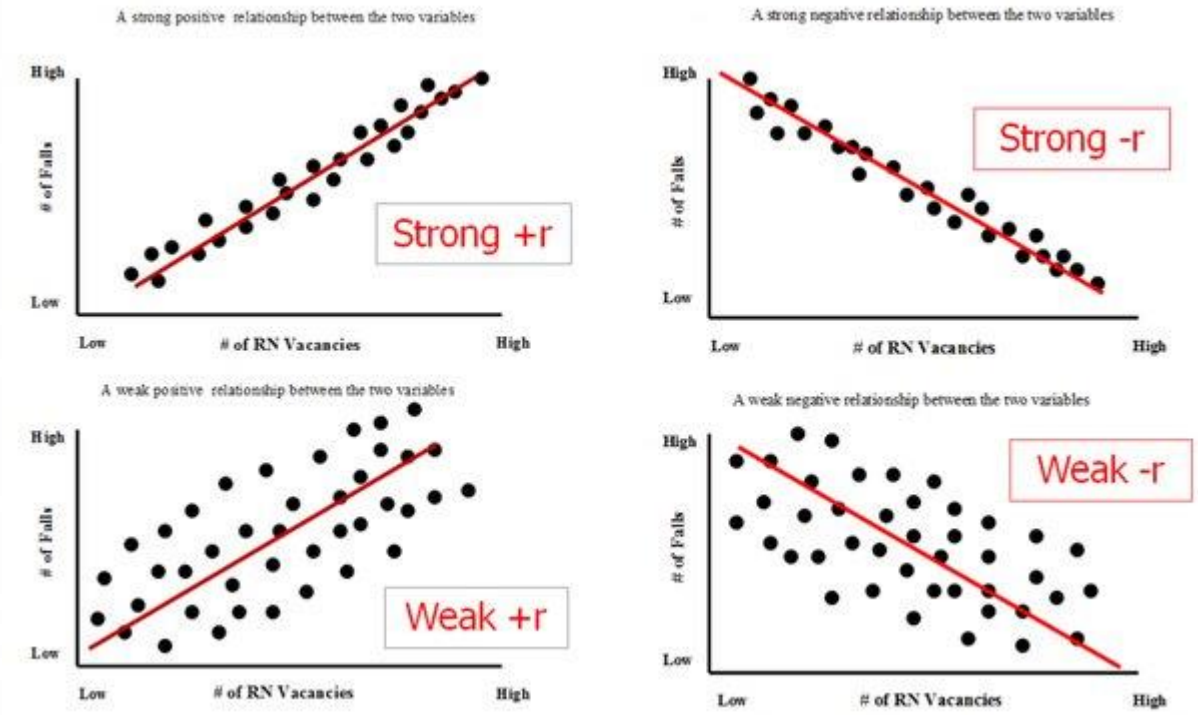
MODELOS LINEARES DE REGRESSÃO

Notação

Medidas de desempenho comum em Regressão



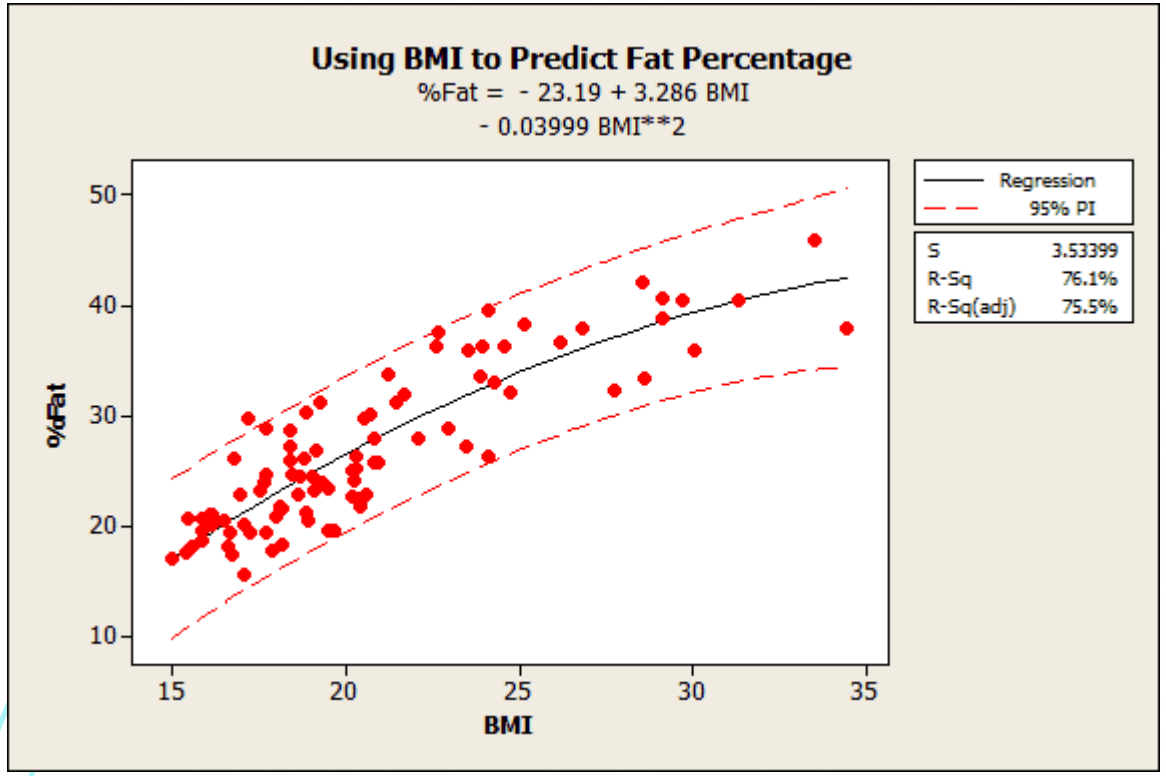
Intervalo de Confiança 95%



MODELOS LINEARES DE REGRESSÃO

Notação

Medidas de desempenho comum em Regressão



Existe uma **probabilidade de 95%** de que, no futuro, o verdadeiro valor do parâmetro da população (por exemplo, média) caia no intervalo **X** (limite inferior) e **Y** (limite superior).

O intervalo de confiança é importante para indicar a margem de incerteza (ou imprecisão) frente a um cálculo efetuado. Esse cálculo usa a amostra do estudo para estimar o tamanho real do resultado na população de origem.

O cálculo de um intervalo de confiança é uma estratégia que considera a amostragem de erro. A dimensão do resultado do seu estudo e seu intervalo de confiança caracterizam os valores presumíveis para a população original.

Quanto mais estreito for o intervalo de confiança, maior é a probabilidade da porcentagem da população de estudo representar o número real da população de origem dando maior certeza quanto ao resultado do objeto de estudo.

https://www.significados.com.br/intervalo-de-confianca/

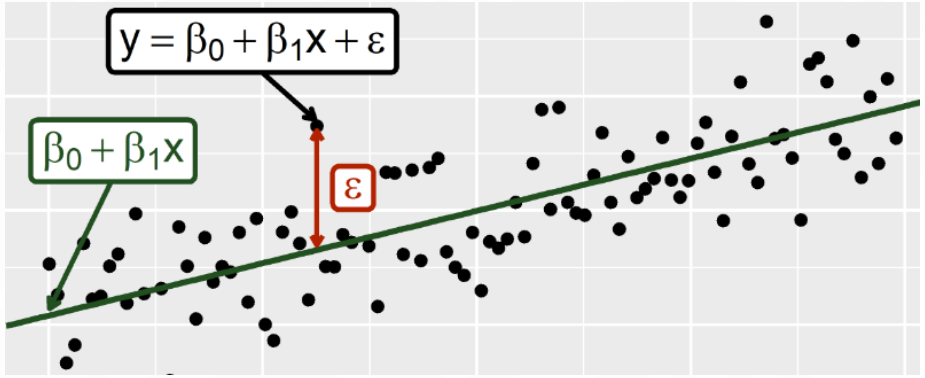
MODELOS LINEARES DE REGRESSÃO

A obtenção dos pesos do modelo (e do bias) consiste em resolver um problema de otimização de minimização da função de perda do erro quadrático médio (mean squared error – MSE) entre os valores observados e os valores preditos com a restrição de que este erro seja nulo

Ou seja, encontrar o vetor de pesos tal que $\min \frac{1}{m} \sum_{i=1}^{m} \left(\mathbf{W}^{\mathbf{T}} \mathbf{x}^{(\mathbf{i})} - \mathbf{y}^{(\mathbf{i})} \right)^2$, s.a. $\epsilon = 0$ onde $y = w_0 + w_1 x_1 + \epsilon$

Uma solução é a equação normal $\hat{\mathbf{W}} = \left(\mathbf{X}^{\mathbf{T}} \mathbf{X} \right)^{-1} \mathbf{X} \mathbf{y}$

Mas há uma inversa de produto de matrizes a resolver (n+1) x (n+1). A classe LinearRegression do scikit-learn é baseada na função scipy.linalg.lstsq() – mínimos quadrados, que usa uma versão mais efetiva com decomposição de valores singulares (SVD).

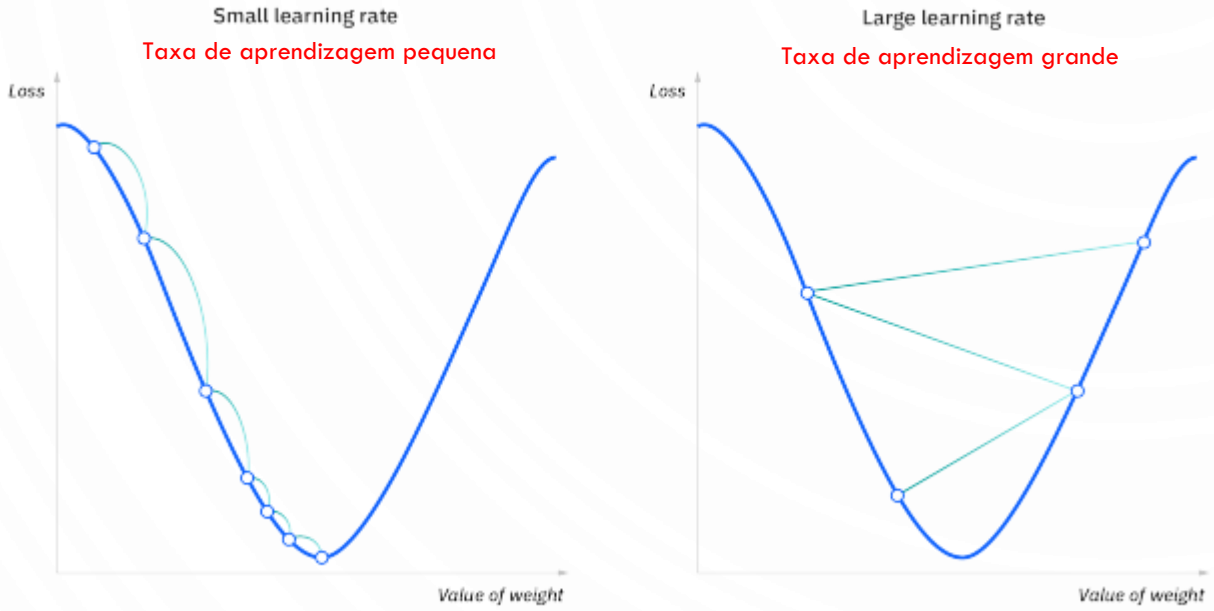


Complexidade eq. normal: $O(n^{2.4})$ a $O(n^3)$
Complexidade SVD scikit-learn: $O(n^2)$

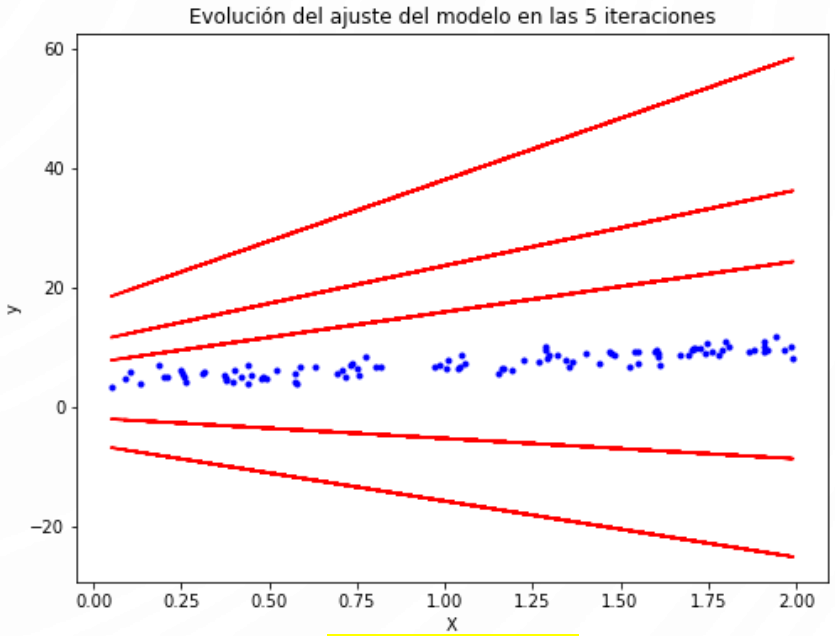
São contudo eficientes para lidar com grandes conjuntos de treinamento: $O(m)$
Para predição são lineares para número de instâncias e de características

GRADIENTE DESCENDENTE - GD

O gradiente descendente é um algoritmo de otimização que costuma ser usado para treinar modelos de aprendizado de máquina e redes neurais. Ele treina modelos de aprendizado de máquina minimizando os erros entre os resultados previstos e os reais, ou seja minimizando a função de perda associada.



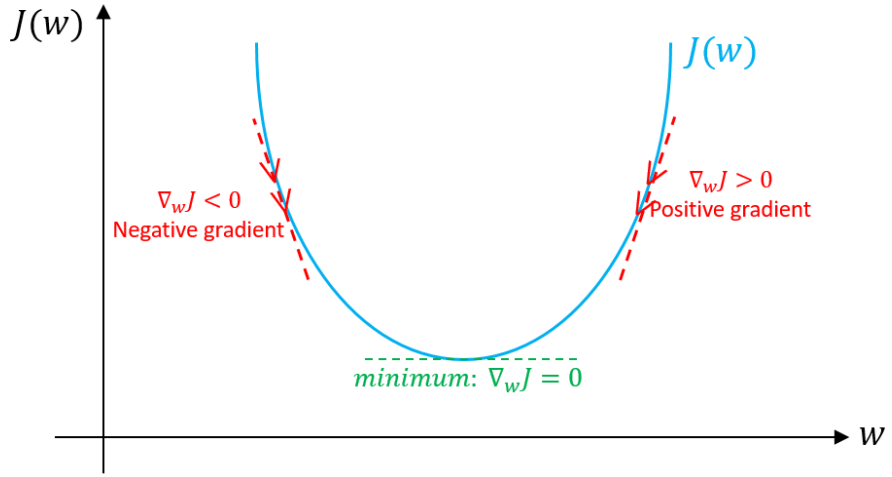
Fonte: IBM

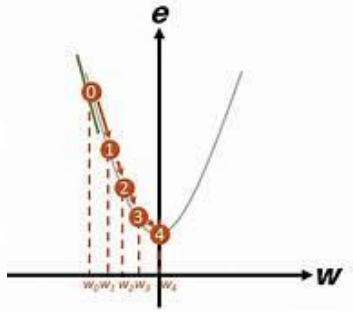


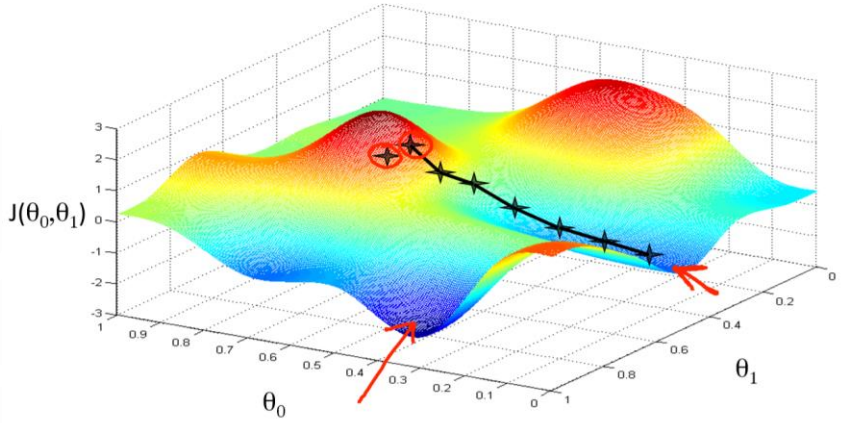
Aprendizado

Parâmetros do modelo inicializados aleatoriamente. A cada passo calcula o gradiente (ou seja, a derivada parcial da função de perda em relação ao vetor de pesos W e segue em direção ao gradiente descendente, até o zero (convergir para o mínimo). O tamanho da etapa de aprendizado é proporcional à inclinação da função de perda, logo as etapas ficam gradualmente menores à medida que se os pesos se aproximam do mínimo

GRADIENTE DESCENDENTE EM LOTE (BATCH) - GD







É necessário calcular o gradiente da função de perda em relação à cada peso do modelo, ou seja, quanto a função mudará caso se modifique um pouco o peso. A cada etapa TODO o conjunto (lote de dados) é usado, sendo lento para conjunto de treinamento muito grandes. Contudo, escalona bem o número de características, sendo muito mais rápido do que a equação normal ou SVD.

$$\frac{\partial}{\partial w_j} MSE(\mathbf{W}) = \frac{\mathbf{2}}{\mathbf{m}} \sum_{\mathbf{i=1}}^{\mathbf{m}} \left(\mathbf{W^T x^{(i)}} - \mathbf{y^{(i)}} \right) \mathbf{x_j^{(i)}}$$

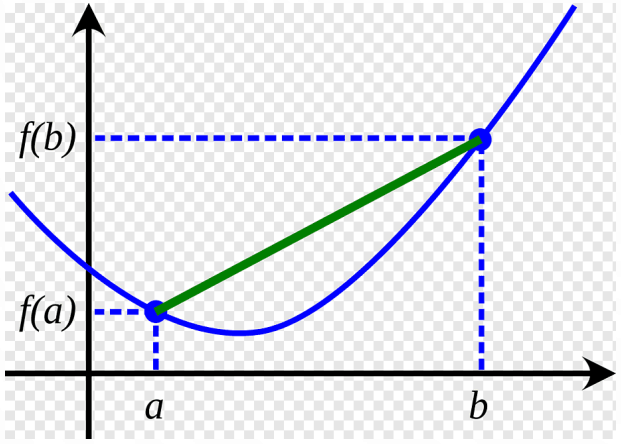
$$\nabla_w MSE(\mathbf{W}) = \begin{pmatrix} \frac{\partial}{\partial w_0} MSE(\mathbf{W}) \\ \frac{\partial}{\partial w_1} MSE(\mathbf{W}) \\ \vdots \\ \frac{\partial}{\partial w_n} MSE(\mathbf{W}) \end{pmatrix} = \frac{\mathbf{2}}{\mathbf{m}} \mathbf{X^T} (\mathbf{XW} - \mathbf{y})$$

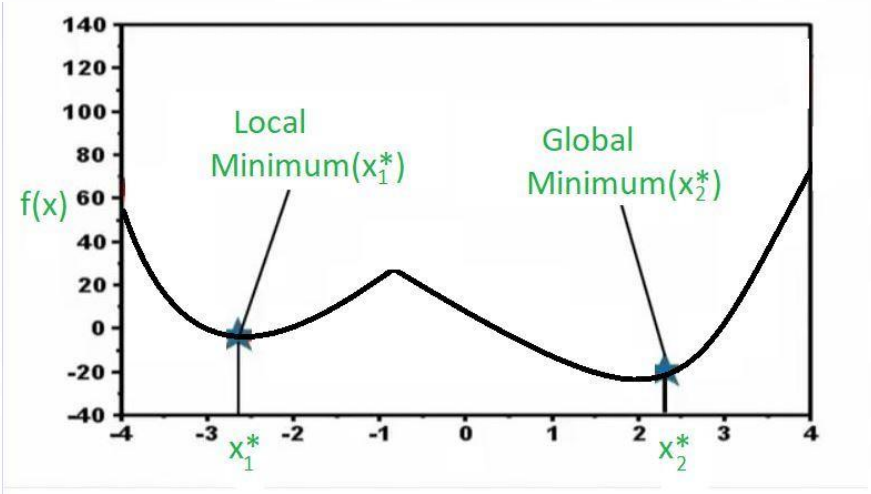
GRADIENTE DESCENDENTE EM LOTE (BATCH) - GD

Expressão de aprendizado – etapa do GD, supondo iterador k:

$$\mathbf{W}(\mathbf{k}+\mathbf{1}) = \mathbf{W}(\mathbf{k}) - \eta \nabla_{\mathbf{w}} \mathbf{MSE}(\mathbf{W}), \text{ onde } \eta\text{: taxa de aprendizagem}$$

Taxa de aprendizagem baixa: muitas iterações para convergir, lento
Taxa de aprendizagem alta: pode atravessar o vale e ir para o outro lado em lugar ainda mais alto do que onde estava, divergindo
Com funções convexas (como MSE) se pressupõe que não há mínimos locais, apenas um mínimo global e não muda abruptamente. A BUSCA é feita no espaço de pesos do modelo. Quanto mais pesos, mas complexa a busca.
Contudo é muito importante que as features do conjunto de treinamento tenham a mesma escala!





Dados dois pontos, o segmento de reta que os une nunca cruza a curva

GRADIENTE DESCENDENTE ESTOCÁSTICO (SGD)

Seleciona uma instância aleatória no conjunto de treinamento a cada etapa e calcula os gradientes com base apenas nesta instância única

Mais rápido, menos dados a cada iteração

Treinamento em grandes conjuntos de dados, pois só uma precisa estar na memória a cada iteração

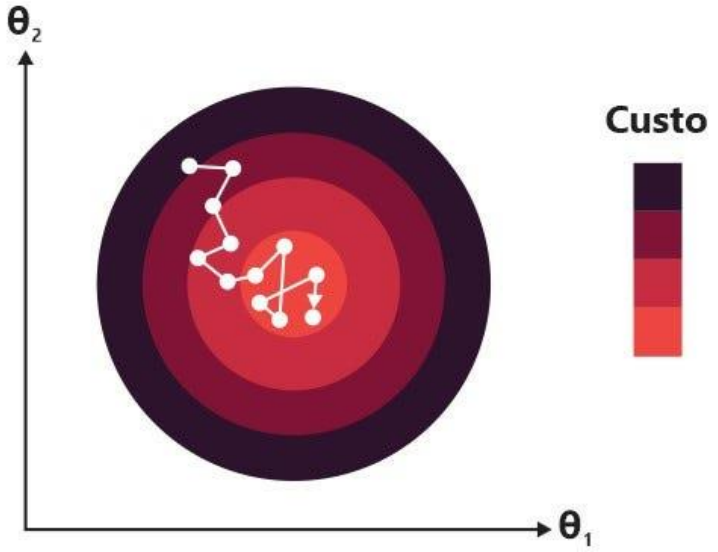
Contudo, é aleatório, menos regular que o em batch. Não desce suave, mas fica oscilando, pode gerar solução subótima

Para funções de perda irregulares (diferente da MSE) pode ajudar a fugir de mínimos locais e chegar ao global

Mas pode nunca se estabilizar no mínimo, e existem soluções de reduzir gradualmente a taxa de aprendizado
Implementar CRONOGRAMA DE APRENDIZADO

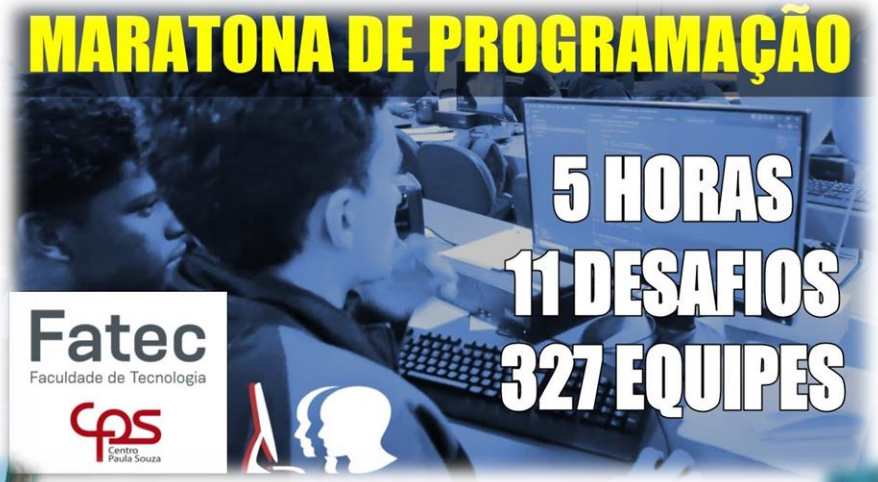
No Scikit-Learn, SGDRegressor

Embaralhar o conjunto de treinamento junto com os rótulos, e as instâncias precisam ser independentes



GRADIENTE DESCENDENTE ESTOCÁSTICO (SGD)

ÉPOCA DE TREINAMENTO: processo de FORWARD PASS (calcular as saídas) e ao processo de BACKWARD PASS (atualizar os pesos) em todo o conjunto de treinamento



Bob Burnquist - Skatista

Final Feminina – Carabina – Tiro esportivo

GRADIENTE DESCENDENTE MINI-BATCH

Calcula os gradientes com base em pequenos conjuntos aleatórios de instâncias

Permite obter ganho de desempenho com a otimização de hardware nas operações da matriz, especialmente se usar GPU

O progresso do algoritmo é menos irregular que o GD estocástico, principalmente com grandes mini-batches

Pode ser mais difícil escapar de mínimos locais

- O caminho do batch para no mínimo, SGD e mini-batch prosseguem, perto do mínimo.
- GD batch leva muito tempo para cada época, os outros também alcançariam com um bom cronograma de aprendizado

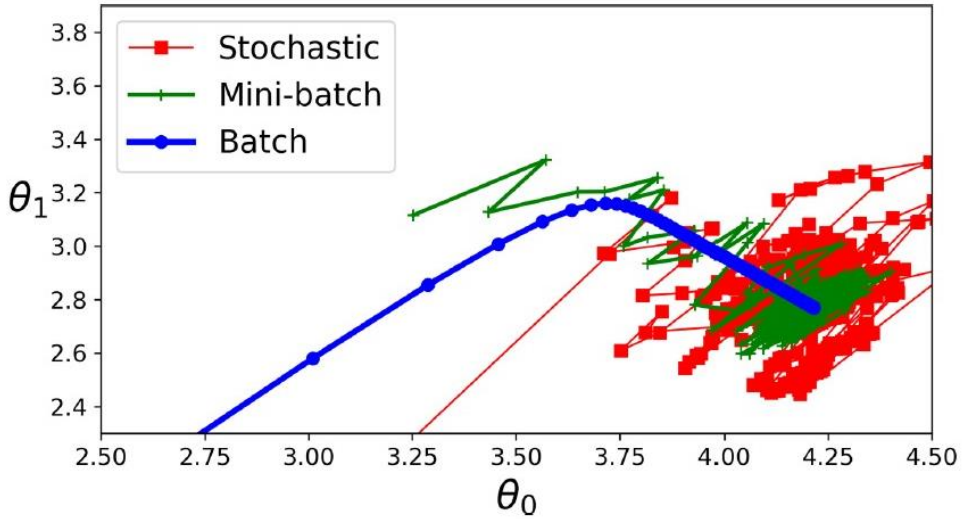


Figure 4-11. Gradient Descent paths in parameter space

(Géron, 2019)

REGRESSORES

Vários algoritmos na scikit-learn conhecidos como CLASSIFICADORES (Classifier), tem suas versões REGRESSORES. Exemplo:

KNeighborsRegressor (vizinhos mais próximos)
MLPRegressor (redes neurais)
DecisionTreeRegressor (árvore de decisão)
SGDRegressor
SVMRegressor (SVM)
RandomForestRegressor
XGBoost Regressor
etc.

Em geral se constrói uma classe para validação cruzada com Grid Search e comparação dos modelos

Várias empresas oferecem o serviço com modelos deste tipo. Exemplo: TOTVS

CUIDADOS AO USAR REGRESSÃO

Conforme lembrado pela AWS, analisar os resíduos do modelo obtido, para checar:

Relacionamento linear Correlação linear

Deve existir uma relação linear entre as variáveis independentes e dependentes. Para determinar essa relação, os cientistas de dados criam um gráfico de dispersão (uma coleção aleatória de valores x e y) para ver se eles se encaixam em uma linha reta. Se isso não acontecer, será possível aplicar funções não lineares, como raiz quadrada ou logarítmica, para criar matematicamente a relação linear entre as duas variáveis.

Independência residual ACF

Os cientistas de dados usam resíduos para medir a precisão da previsão. Um resíduo é a diferença entre os dados observados e o valor previsto. Resíduos não devem ter um padrão identificável entre eles. Por exemplo, os resíduos não devem crescer com o passar do tempo. Diferentes testes matemáticos, como o teste de Durbin-Watson, podem ser utilizados para determinar a independência residual. É possível usar dados fictícios para substituir qualquer variação de dados, como dados sazonais.

Normalidade

Técnicas de representação gráfica, como gráficos Q-Q, determinam se os resíduos são normalmente distribuídos. Os resíduos devem se encaixar ao longo de uma linha diagonal no centro do gráfico. Se eles não forem normalizados, você poderá testar os dados quanto à presença de outliers aleatórios ou valores atípicos. Remover desses outliers ou realizar transformações não lineares pode corrigir o problema. Box-cox

Homocedasticidade

A homocedasticidade supõe que os resíduos tenham uma variância constante ou desvio padrão da média para cada valor de x. Se esse não for o caso, os resultados da análise podem não ser precisos. Se essa suposição não for atendida, talvez seja necessário alterar a variável dependente. Como a variância ocorre naturalmente em conjuntos de dados grandes, faz sentido alterar a escala da variável dependente. Por exemplo, em vez de usar o tamanho da população para prever o número de postos de bombeiros em uma cidade, é possível usar o tamanho da população para prever o número de postos de bombeiros por pessoa.