Redução de Dimensionalidade

1. Introdução

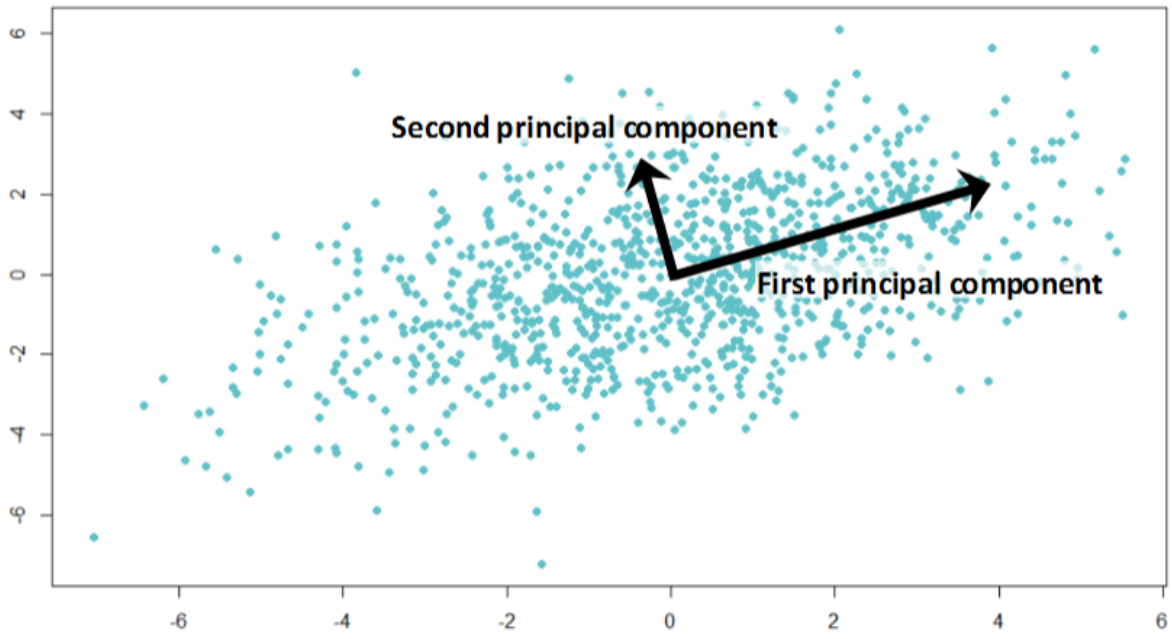
Com o avanço tecnológico e com aumento gradativo da geração e armazenamento de dados, os conjuntos de dados a serem trabalhados por modelos de *Machine Learning* e *Data Science* estão cada vez maiores e mais complexos, implicando também em um aumento gradativo do poder e tempo de processamento de máquinas, sejam elas locais ou em processamento em nuvem.

Mas, com o objetivo de minimizar estes impactos, existem algumas técnicas complementares visando auxiliar e diminuir esta carga de processamento, uma dessas técnicas seria a **redução de dimensionalidade**. Redução de dimensionalidade é uma técnica bastante usada em conjunto de dados, normalmente grandes, com o objetivo de aumentar a interpretabilidade dos dados minimizando a quantidade de informação perdida no processo.

Algumas das principais técnicas de redução de dimensionalidade utilizadas são o **Análise de Componente Principal (PCA)** e a **Análise Discriminante Linear (LDA)**, técnicas que serão detalhadas a seguir.

2. Análise de Componente Principal (PCA)

A Análise de Componente Principal (*Principal Component Analysis* em inglês) é a técnica para reduzir a dimensionalidade desses conjuntos de dados, aumentando a interpretabilidade concomitante a minimização da perda de informações. Isso é feito criando variáveis não correlacionadas, preservando o máximo de variabilidade possível. O processo matemático por trás disso consiste em uma transformação linear buscando calcular os **autovetores** e indicando as direções principais deste conjunto de dados.



Fonte: Analytic Vidhya

Este processo gera um novo conjunto de dados ainda com n variáveis, mas agora não correlacionadas, estas variáveis não correlacionadas são denominadas componentes principais. A redução da dimensionalidade do conjunto de dados deve-se a justamente escolher a quantidade de componentes principais a serem utilizados, retornando pelo PCA uma quantidade n de componentes principais que representam a variabilidade dos dados.

Algumas das vantagens desse processo seria o ganho de uma **interpretação gráfica dos dados** minimizando a perda de informação e também um processo interessante para utilizar em testes de modelo onde seria necessário utilizar um **conjunto de dados muito grande**, pois testa-se o modelo com poucas variáveis, mas sem perder o valor e a variabilidade dos dados originais.

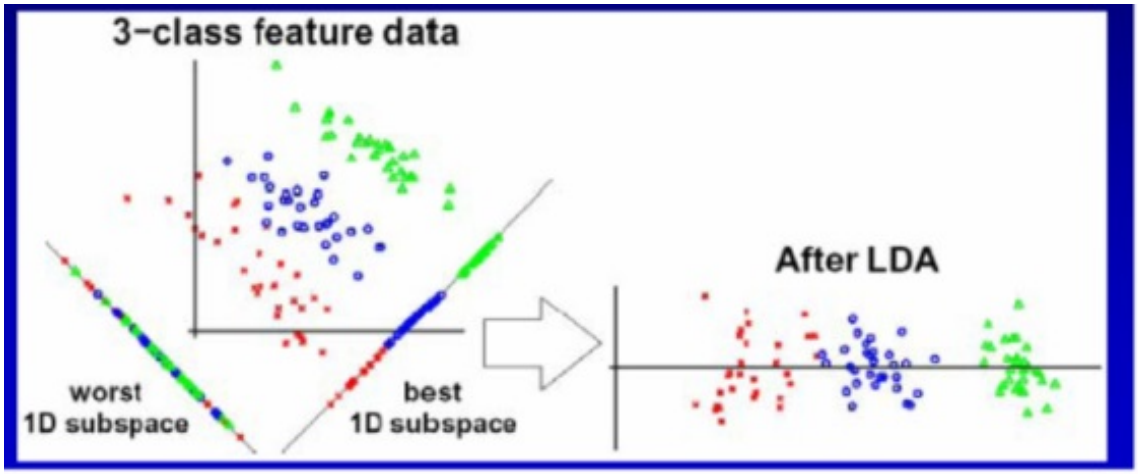
A implementação em *Python* para o PCA é dada pelo bloco de código abaixo:

```
# Carregando o PCA do Scikit-Learn
from sklearn.decomposition import PCA

# Instanciando o PCA
pca = PCA(n_components = 2,  # Quantidade de componentes que serão utilizadas
          random_state = 42) # Semente Aleatório

# Transforma os dados e cria o número de componentes necessários
X_pca = pca.fit_transform(X)
```

3. Análise Discriminante Linear (LDA)

A Análise Discriminante Linear é uma técnica que visa transformar as observações multivariadas, por meio de combinações lineares dessas variáveis, em observações univariadas de tal forma que as variáveis transformadas se apresentassem o mais separadas possível. Portanto, a análise discriminante é uma técnica de análise multivariada utilizada para diferenciar ou discriminar populações e classificar ou alocar indivíduos em populações pré-definidas.

Vale ressaltar que esta técnica também pode ser usada apenas para reduzir a dimensionalidade. Por ser usada para modelar diferenças em grupos, ou seja, separar duas ou mais classes, ao ser usada como redutora de dimensionalidade, a transformação linear garante que no novo espaço as classes terão máxima separabilidade. O processo matemático por trás do LDA, consiste em projetar ortogonalmente o conjunto de dados em subespaços que maximize a separação das classes oriundas:



Fonte: ML Algorithm

A implementação em *Python* para o LDA é dada pelo bloco de código abaixo:

```
 # Carregando o LDA do Scikit-Learn
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

# Instanciando o PCA
lda = LinearDiscriminantAnalysis(n_components = 2)  # Quantidade de
componentes que serão utilizadas

# Transforma os dados e cria o número de componentes necessários
X_lda = lda.fit_transform(X)
```

Materiais Complementares

Canal *StatQuest*, vídeo sobre *PCA Step-by-Step*;

Canal *StatQuest*, vídeo sobre LDA Clearly Explained;

Documentação no Scikit-Learn sobre o PCA;

Documentação no Scikit-Learn sobre o LDA;

Artigo publicado por Chaitanyanarava no *Analytics Vidhya* - *A Complete Guide On Dimensionality Reduction*.

Referências

James, Gareth, et al. An Introduction to Statistical Learning: With Applications in R. Alemanha, Springer New York, 2013;

Bruce A., Bruce P. Estatística Prática para Cientistas de Dados. Segunda Edição, Alta books, 2019;

Tópico anterior

Próximo Tópico