Redução de dimensionalidade em base de imagens manuscritas

Paola de Oliveira Prado

01/04/2021

Contextualização

Atualmente, bases de imagens são bastante utilizadas para a classificação de objetos na área de aprendizado de máquina, *Machine Learning*. O objetivo é analisar duas técnicas de redução de dimensionalidade, escalonamento multidimensional e *t-SNE*.

O escalonamento multidimensional foi desensolvido para permitir a visualização dos dados em uma dimensão menor que a original. A técnica baseia-se em relações lineares das matrizes de distâncias entre os objetos, ou seja, tem o foco em preservar as dissimilaridades na nova dimensão.

Com o avanço computacional, outras técnicas mais elaboradas tem sido desenvolvidas, uma delas é a técnica *t-SNE*,*t-Distributed Stochastic Neighbor Embedding*, que vem apresentando melhores resultado na visualização dos dados, principalmente os dados mais complexos. Essa técnica é não linear, ou seja, tem o foco em preservar as similaridades na nova dimensão.

Base de Dados

Para analise, será utilizada a base *MNIST*, *Modified National Institute of Standards and Technology*. A base é composta pelo arquivo de imagens, que são pequenas imagens de dígitos de 0 a 9 escritos à mão, no qual os dígitos foram normalizados em tamanho e centralizados em uma imagem de tamanho fixo de forma que cada imagem ocupe 28x28 pixels, ou seja, 784 dimensões e pelo arquivo de rótulos, que contém os dígitos de 0 a 9 associados na ordem das imagens do arquivo de imagens. A base de dados vem divida no conjunto de treinamento com 60.000 observações e nos dados de teste com 10.000 observações. Devido a problemas computacionais, será feita uma amostragem com 4.000 observações.

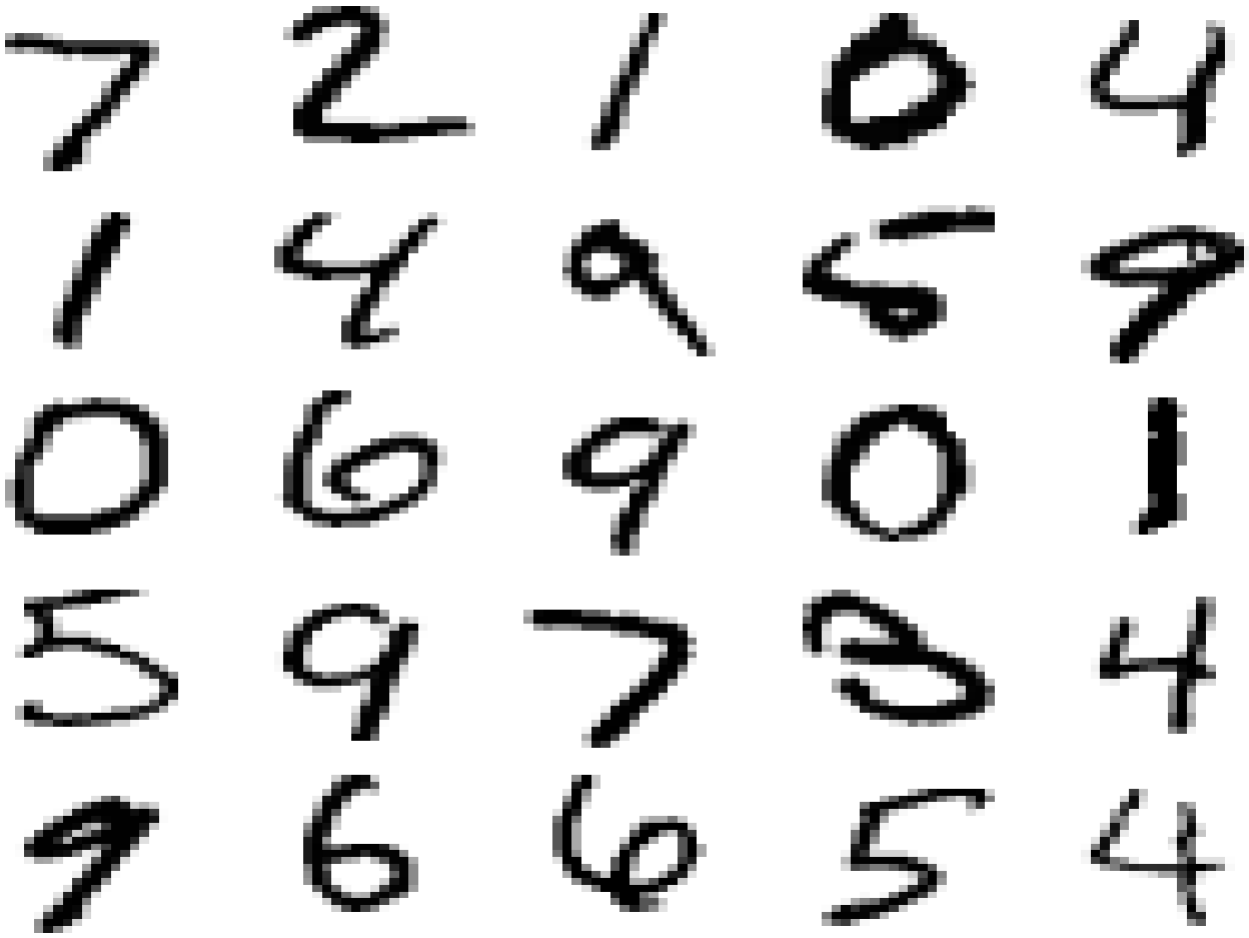
A base *MNIST* está disponível em http://yann.lecun.com/exdb/mnist/ (http://yann.lecun.com/exdb/mnist/)

Pacotes

```
library(tidyverse)
library(Rtsne)
```

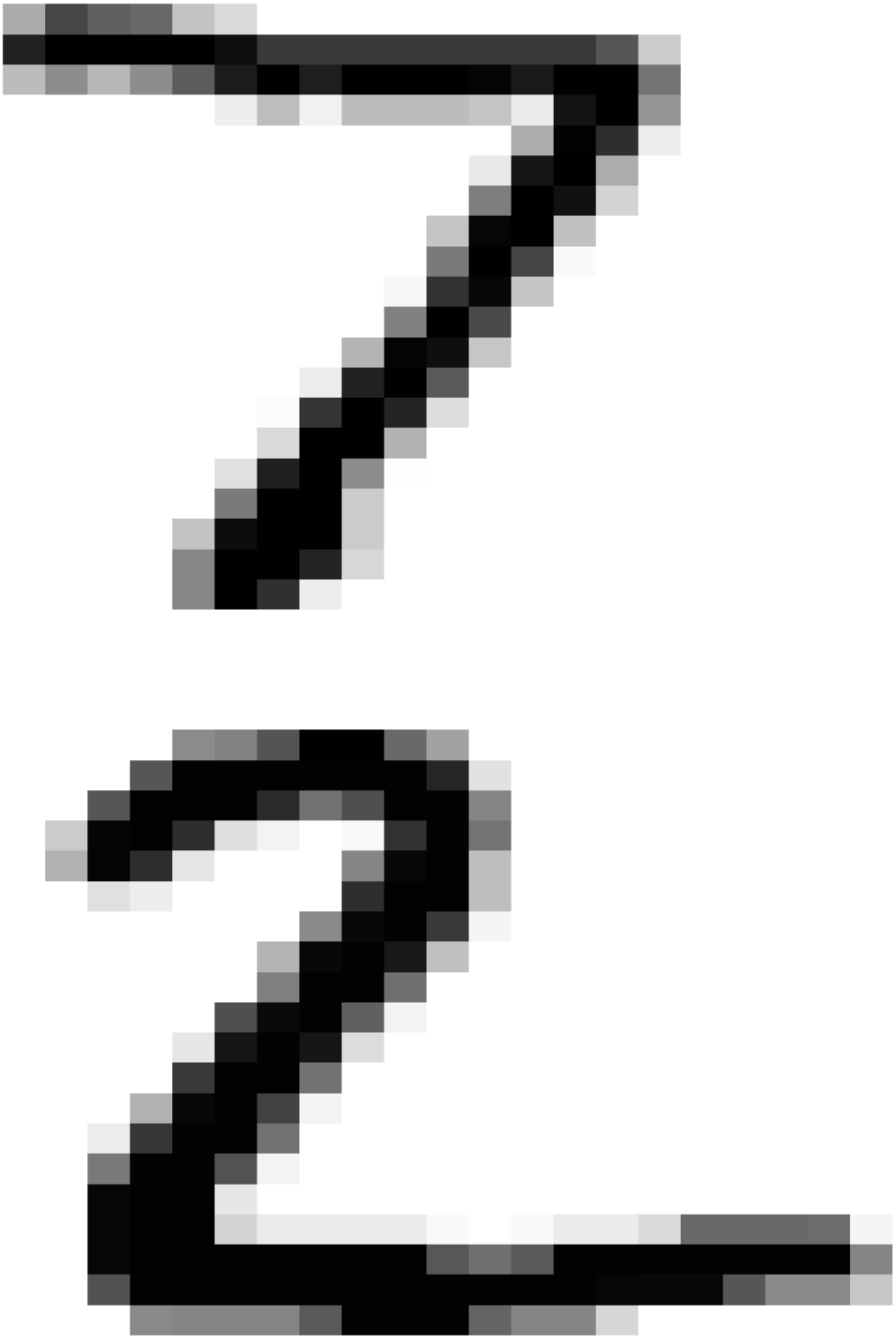
Visualização da base *MNIST*

```
## Vendo as 25 primeiros imagens
to.read = gzfile("t10k-images-idx3-ubyte.gz", "rb")
readBin(to.read, integer(), size=4, n=4, endian="big")
par(mfrow=c(5,5))
par(mar=c(0,0,0,0))
for(i in 1:25){
  m <- matrix(readBin(to.read,integer(), size=1, n=28*28, endian="big", signed = F),
              nrow=28,byrow=T)
  image(t(m)[,28:1],col = gray(255:0 / 255), axes = F)
}
```



```
close(to.read)

## Vendo as duas primeiras imagem
to.read = gzfile("t10k-images-idx3-ubyte.gz", "rb")
readBin(to.read, integer(), size=4, n=4, endian="big")
par(mfrow=c(1,1))
for(i in 1:2){
  m <- matrix(readBin(to.read,integer(), size=1, n=28*28, endian="big", signed = F),
              nrow=28,byrow=T) #Aqui vejo a matriz de rótulos do dígito
  image(t(m)[,28:1],col = gray(255:0 / 255), axes = F)
}
```



```
close(to.read)
```

Leitura da Base *MNIST*

```
set.seed(12345)

to.read = gzfile("t10k-images-idx3-ubyte.gz", "rb")  # Lendo a base de dados das imagens
readBin(to.read, integer(), size=4, n=4, endian="big")

m = matrix(0,ncol=28*28, nrow=10000)
for(i in 1:10000){
  m[i,] <- readBin(to.read,integer(), size=1, n=28*28, endian="big", signed = F)
}
close(to.read)

amostra=sample(1:10000,4000) #Gerando uma amostra com 4000 observaçõe
imagem_nova=m[amostra,]     #Fazendo amostragem das imagens

##LENDO BASE DE DADOS DOS LABELS ASSOCIADOS AS IMAGENS#

to.read = gzfile("t10k-labels-idx1-ubyte.gz", "rb")
readBin(to.read, integer(), size=4, n=2, endian="big")
a = readBin(to.read,integer(), size=1, n=10000, endian="big") #size:número de bytes
close(to.read)

label_nova=a[amostra];label_nova #Fazendo amostragem do label
```

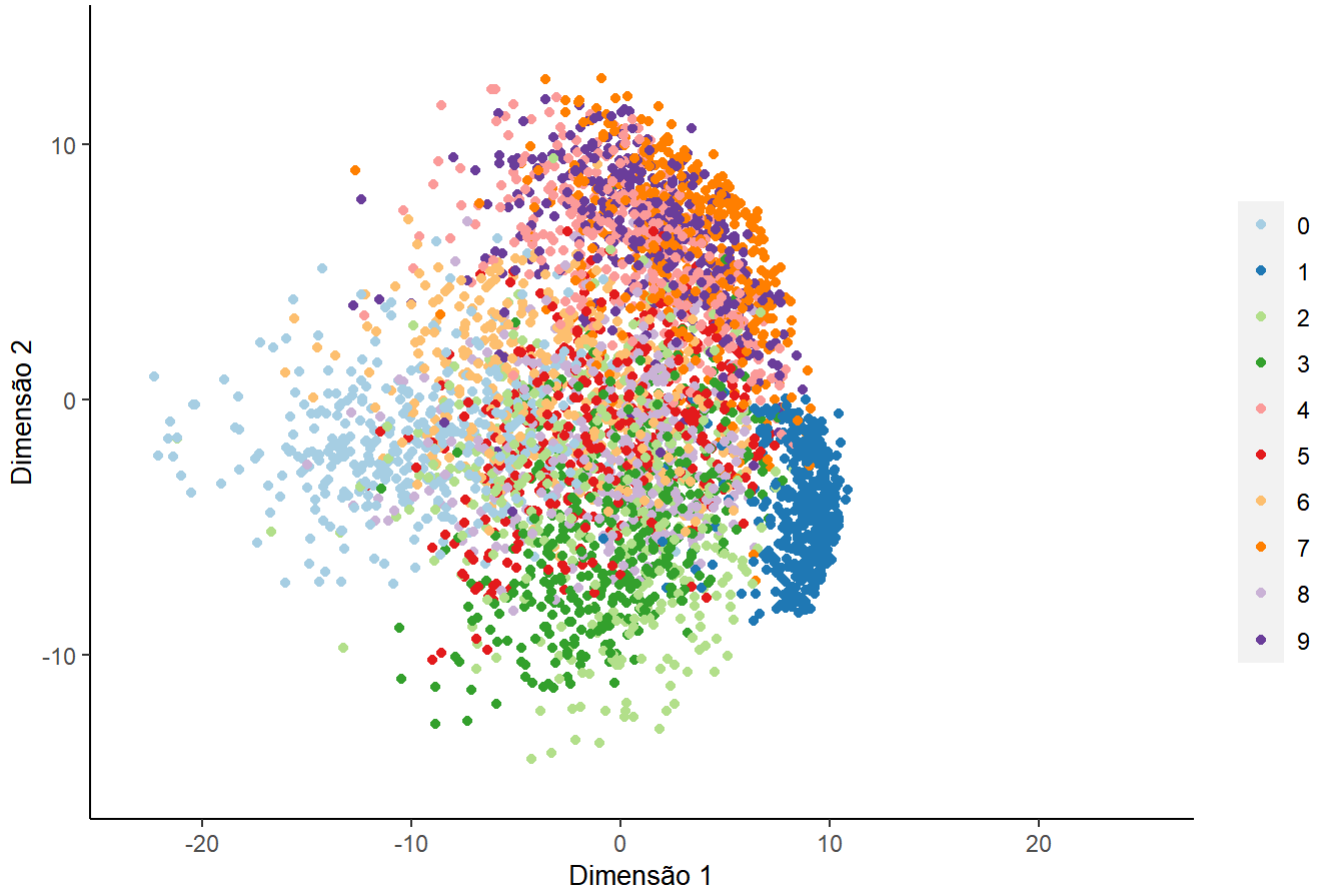
Escalonamento Multidimensional

```
D=dist(imagem_nova)
mds=cmdscale(D, k=2, eig=TRUE)

x=mds$points[,1]
y=mds$points[,2]

mds_4k=data.frame(x,y)/100

ggplot(mds_4k, aes(x = mds_4k$x, y = mds_4k$y, colour = as.factor(label_nova))) +
  geom_point() +
  labs(title="", x = "Dimensão 1", y = "Dimensão 2") +
  scale_x_continuous(limits = c(-23,25))+
  scale_y_continuous(limits = c(-15,14))+
  theme(axis.title = element_text(family = "calibri", size = 10, colour="Black"),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(), #Linha no fundo do gráfico na horizontal
        panel.grid.minor.y = element_blank(),
        panel.background = element_blank(),
        axis.line.x = element_line(size = 0.4, colour = "black"),
        axis.line.y = element_line(size = 0.4, colour = "black"),
        legend.title =element_text(family = "calibri", size = 10, colour="Black"))+
  scale_colour_brewer(palette="Paired", name="")
```

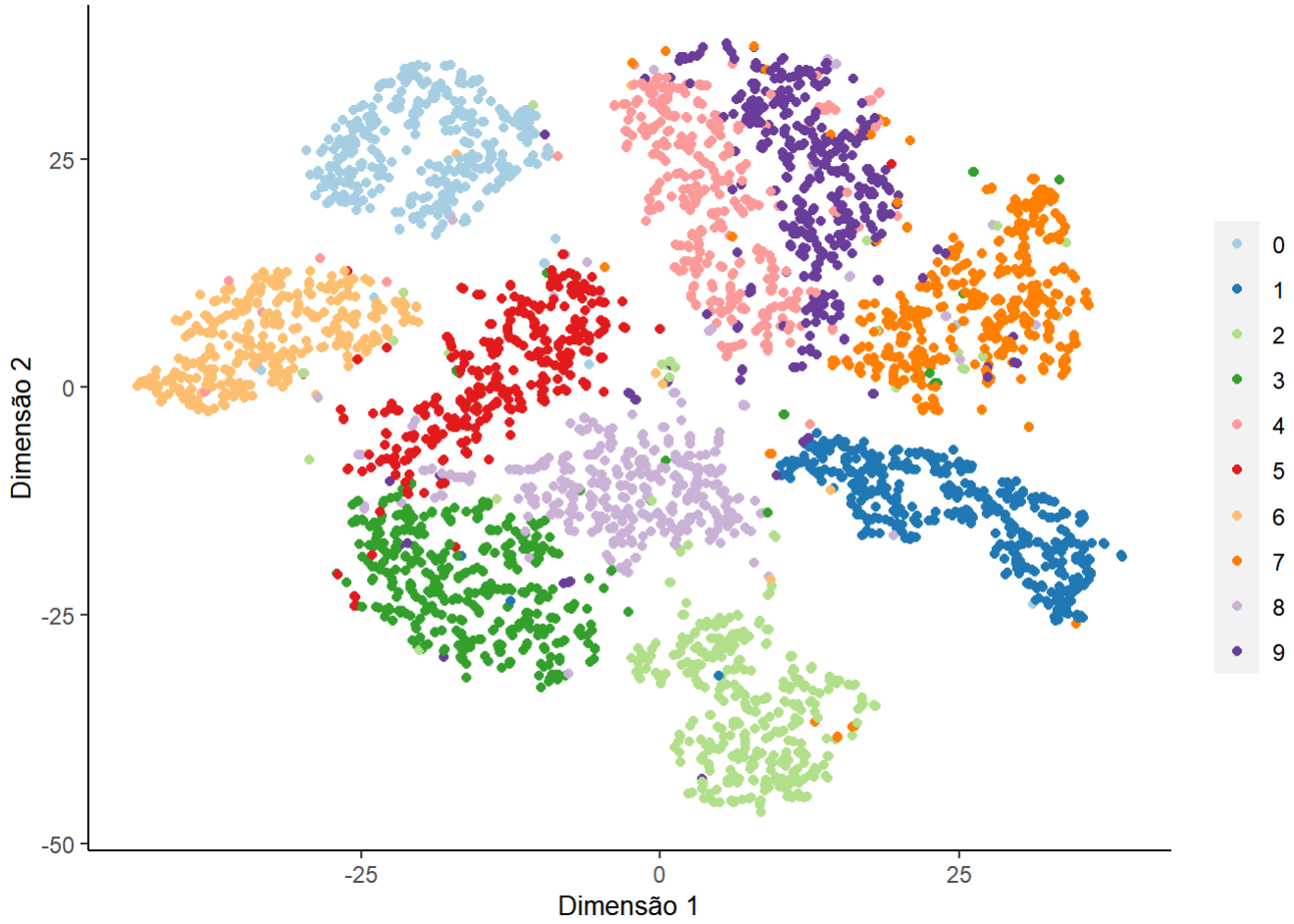


t-SNE

```
set.seed(2021)
tsne = Rtsne(imagem_nova, initial_dims=30, exaggeration_factor = 4, perplexity = 25)

tsne_pca = data.frame(y1 = tsne$Y[,1], y2 = tsne$Y[,2])

ggplot(tsne_pca, aes(x = y1, y = y2 , colour = as.factor(label_nova))) +
  geom_point() +
  labs(x = "Dimensão 1", y = "Dimensão 2") +
  theme(axis.title = element_text(family = "calibri", size = 10, colour="Black"),
        panel.grid.minor = element_blank(),
        panel.grid.major.y = element_blank(), #Linha no fundo do gráfico na horizontal
        panel.grid.minor.y = element_blank(),
        panel.background = element_blank(),
        axis.line.x = element_line(size = 0.4, colour = "black"),
        axis.line.y = element_line(size = 0.4, colour = "black"),
        legend.title =element_text(family = "calibri", size = 10, colour="Black"))+
  scale_colour_brewer(palette="Paired", name="")
```



Considerações Finais

Analisando o gráfico do escalonamento multidimensional, pode-se observar que apesar de alguns dígitos ficarem mais proximos como, principalmente, o dígito 1, não existe um padrão de forma que todos os outros dígitos estejam mais próximos dos seus semelhantes sem se sobrepor a outros dígitos. O contrário acontece com a

visualização do gráfico do *t-SNE*, nota-se que os dígitos semelhantes estão mais próximos, deste modo, formando um agrupamento natural dos dígitos com o mesmo rótulos. Com isso, o *t-SNE* produz resultados melhores do que no escalonamento multidimensional.