Redução de Dimensionalidade para Visualização e Análise de Dados

Wesley Seidel
Tiago Lubiana

Por que estamos falando de Redução de Dimensionalidade?

- Dados de alta dimensionalidade por todos os lados
  - Ciencias omicas
  - Análises de texto
  - Provavelmente, qualquer dataset interessante!

PubMed.gov

"dimensionality reduction" Search

Advanced Create alert Create RSS User Guide

Save Email Send to Sorted by: Best match Display options

RESULTS BY YEAR

3,165 results

Quando falamos de redução de dimensionalidade, falamos geralmente de duas coisas

- Seleção de variáveis
- Criação de novas variáveis num espaço dimensional reduzido (projeção de variáveis)

Nosso foco é na segunda parte, por que o tempo é curto.

Não focaremos nos algoritmos e detalhes matemáticos, mas no uso prático para ciencias de dados

Para que criar novas variáveis num espaço dimensional reduzido?

- Menor custo computacional
- Evitar sobreajuste
- Visualizar dados

Menor custo computacional

- Menos colunas, economia para algoritmos caros e menos espaço na memória

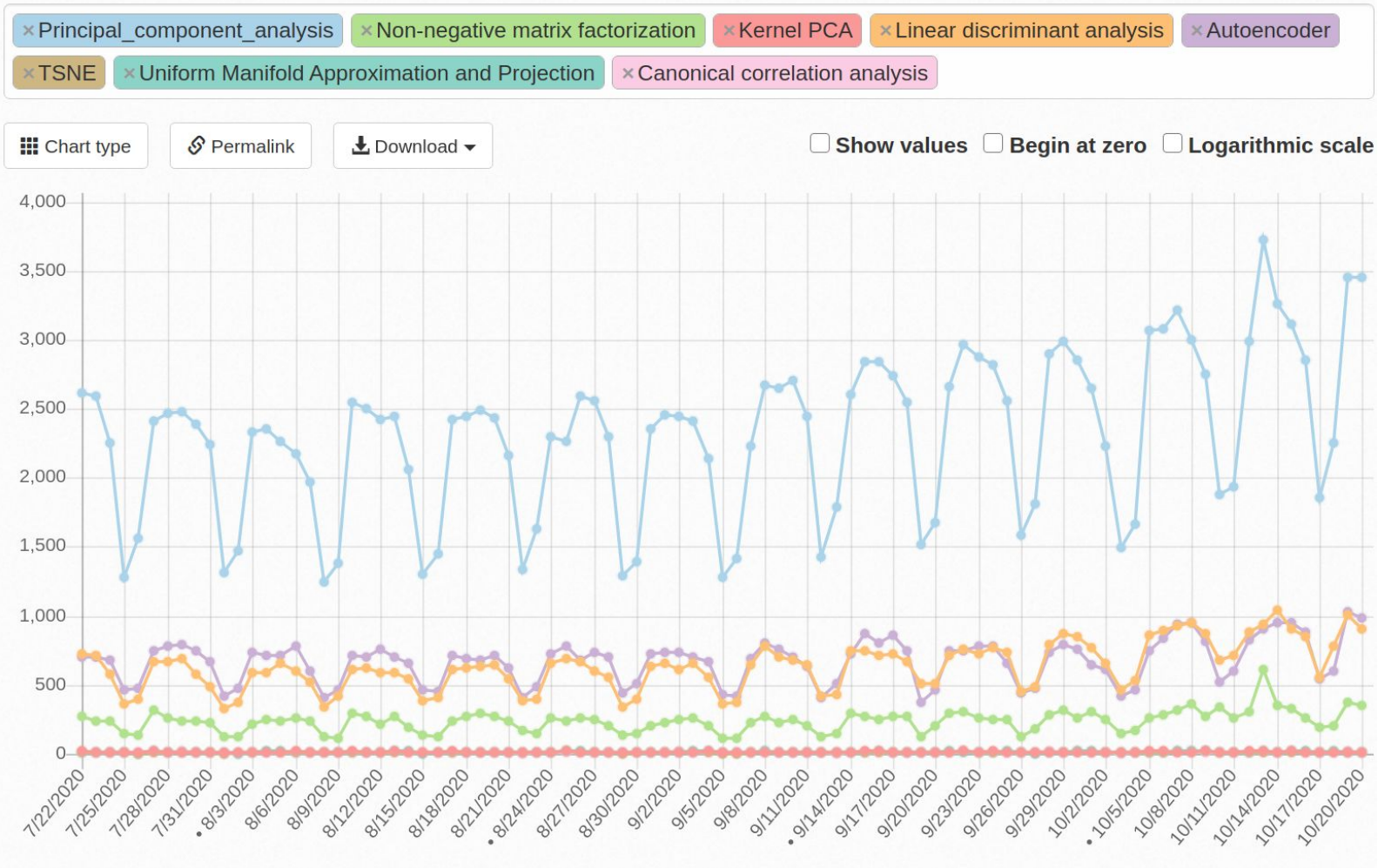
Evitar sobreajuste

- Menos variáveis, menos espaço para alguma medição espúria correlacionar aleatoriamente com a classe alvo

Visualizar dados

- Gráficos em duas dimensoes são mais confortáveis para o (meu) cérebro humano

Os algoritmos de redução dimensional visam preservar propriedades do espaço multidimensional para otimizar esses objetivos.


Principal_component_analysis
Non-negative matrix factorization
Kernel PCA
Linear discriminant analysis
Autoencoder
TSNE
Uniform Manifold Approximation and Projection
Canonical correlation analysis
Chart type
Permalink
Download
Show values
Begin at zero
Logarithmic scale
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
7/22/2020
7/25/2020
7/28/2020
7/31/2020
8/3/2020
8/6/2020
8/9/2020
8/12/2020
8/15/2020
8/18/2020
8/21/2020
8/24/2020
8/27/2020
8/30/2020
9/2/2020
9/5/2020
9/8/2020
9/11/2020
9/14/2020
9/17/2020
9/20/2020
9/23/2020
9/26/2020
9/29/2020
10/2/2020
10/5/2020
10/8/2020
10/11/2020
10/14/2020
10/17/2020
10/20/2020

O PCA tem seu charme e apelo por que é:

- Relativamente simples de entender
- Relativamente simples de implementar
- Já é muito usado pela comunidade

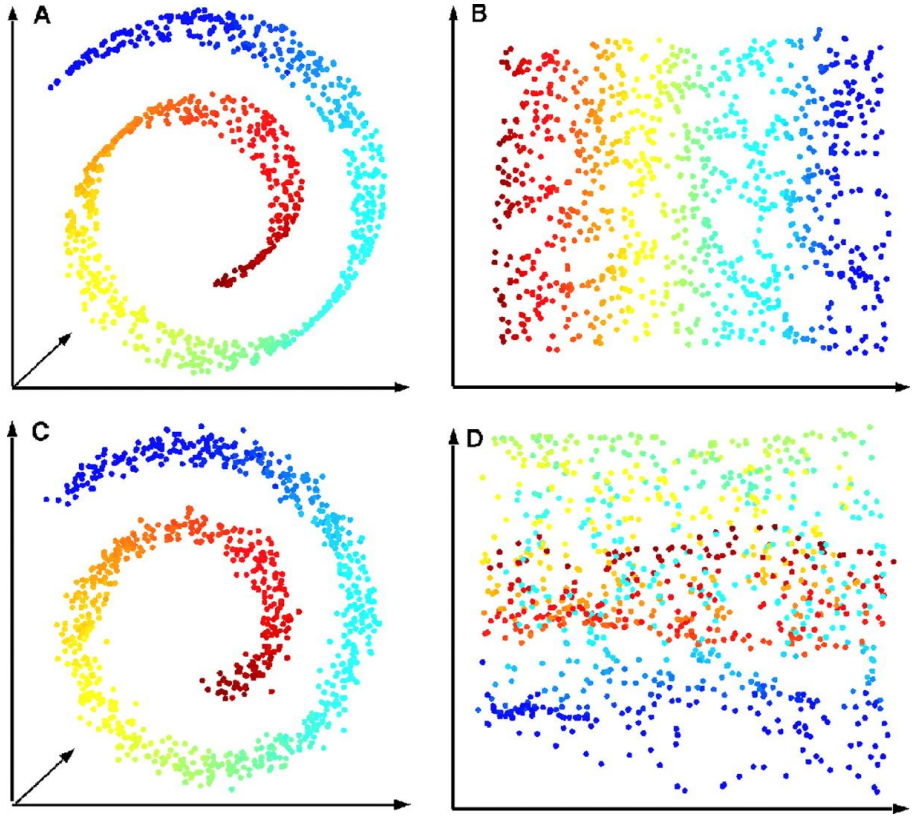
(Explicar a matemática do PCA)

(Exemplos práticos de uso do PCA, talvez com o dataset MNIST)

Apesar de ser extremamente útil, o PCA busca correlações lineares, e nem sempre é o que queremos.

Casos complicados ...

- pedem algoritmos mais complexos.
    - por exemplo desenrolar um rocambole,



Algoritmo mais complexo

PCA

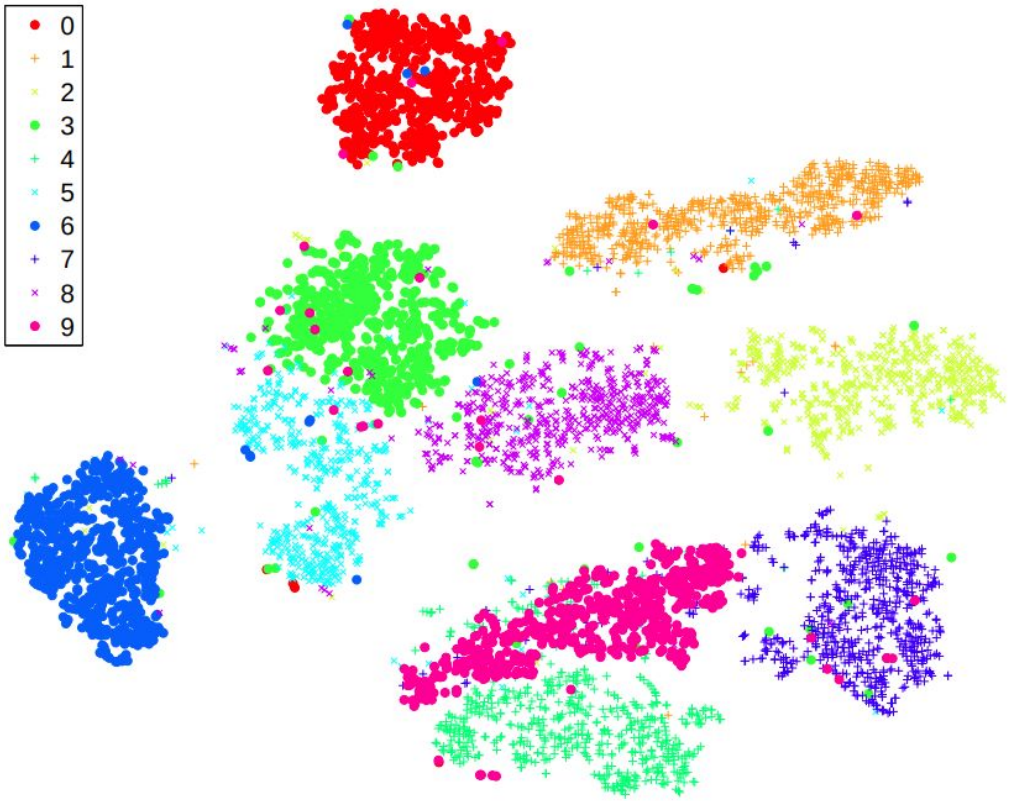
The Isomap Algorithm and Topological Stability

t-SNE

O artigo de Laurens van der Maaten e Geoffrey Hinton entitulado Visualizing Data using t-SNE foi divulgado em 2008 e tem mais de 15 mil citações até hoje. (veja aqui)

O motivo do sucesso é de ordem prática:

- Os dados ficam bonitos.
- Coisas que acreditamos que deviam ficar próximas ficam próximas

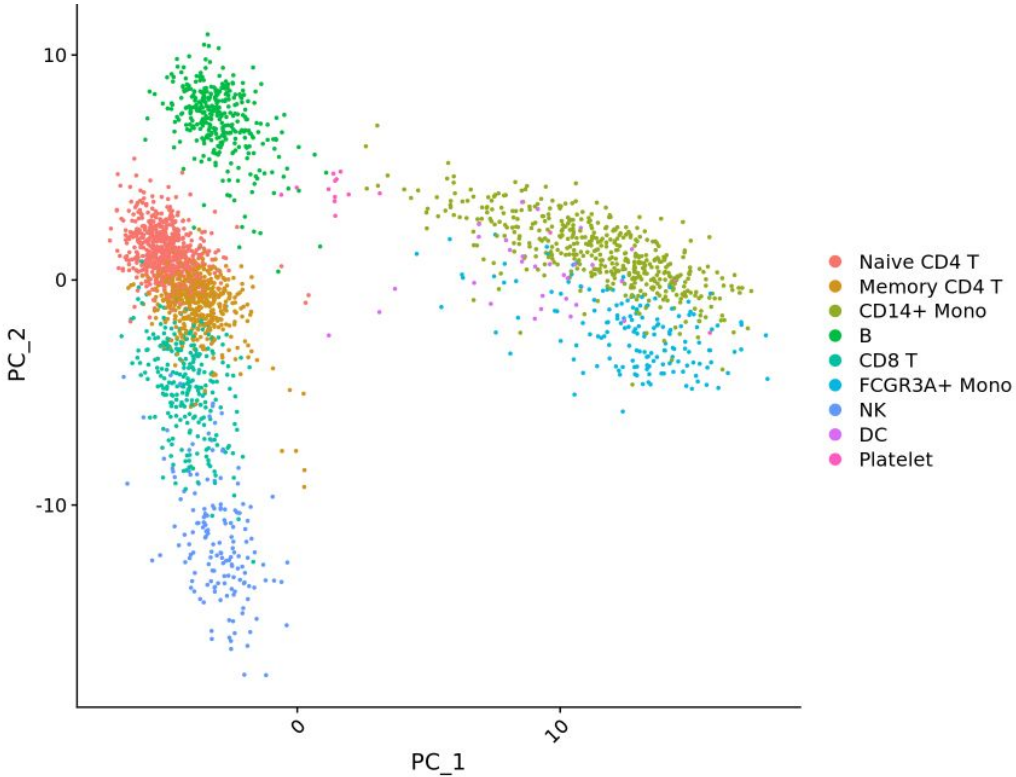


(a) Visualization by t-SNE.

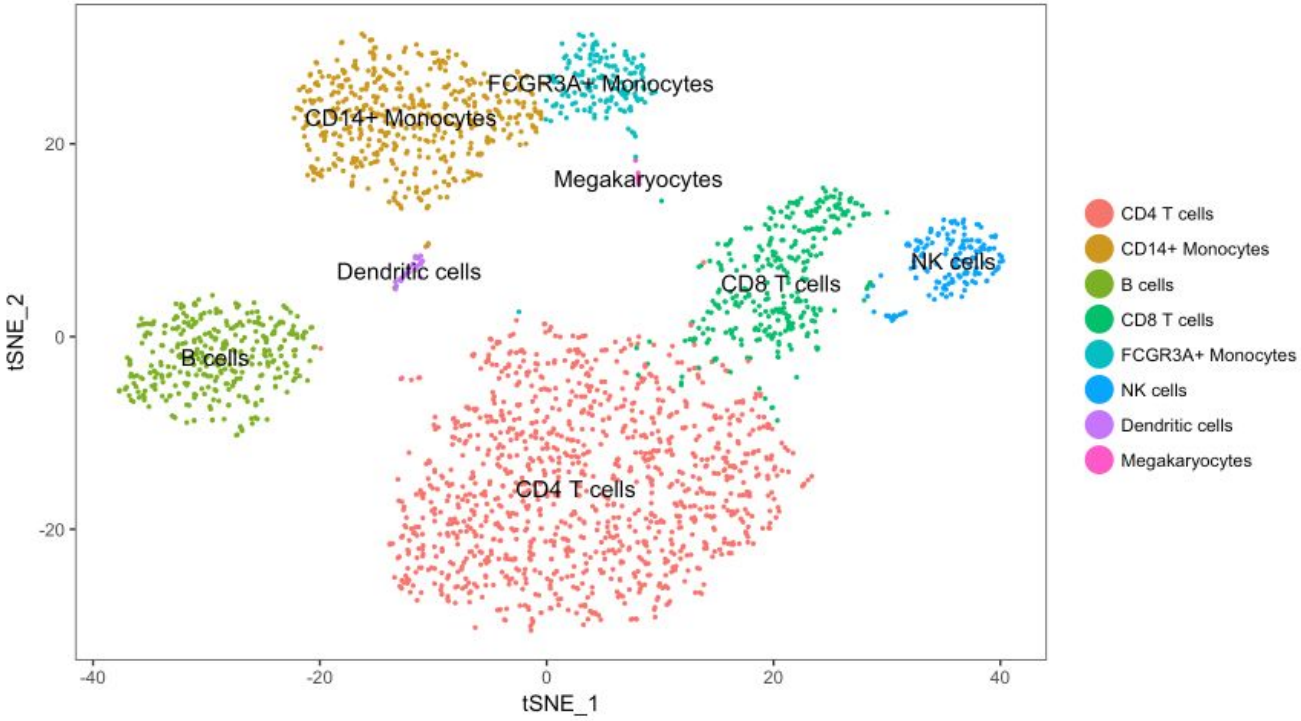
- Entrada:
    - Conjunto de dados numéricos
    - Parametros ajustáveis (perplexidade é o principal)
- Saída:
    - n dimensões
- Implementações diversas, ex: class sklearn.manifold.TSNE()
- NOTA: alguns algoritmos de tSNE rodam um PCA antes nos dados!
- A ideia geral do tSNE é de preservar “comunidades” em vez de preservar variancia. (talvez dar mais o feeling)

Transcriptomica de células únicas

- É o que está bombando na data science biomédica, com o projeto Human Cell Atlas
    - Projeto Genoma dessa década
    - Medição de expressão de >20.000 genes por célula
    - Datasets de um único experimento com mihares a centenas de milhares de células



https://satijalab.org/seurat/v3.2/visualization_vignette.html



https://satijalab.org/seurat/v1.4/pbmc3k_tutorial.html

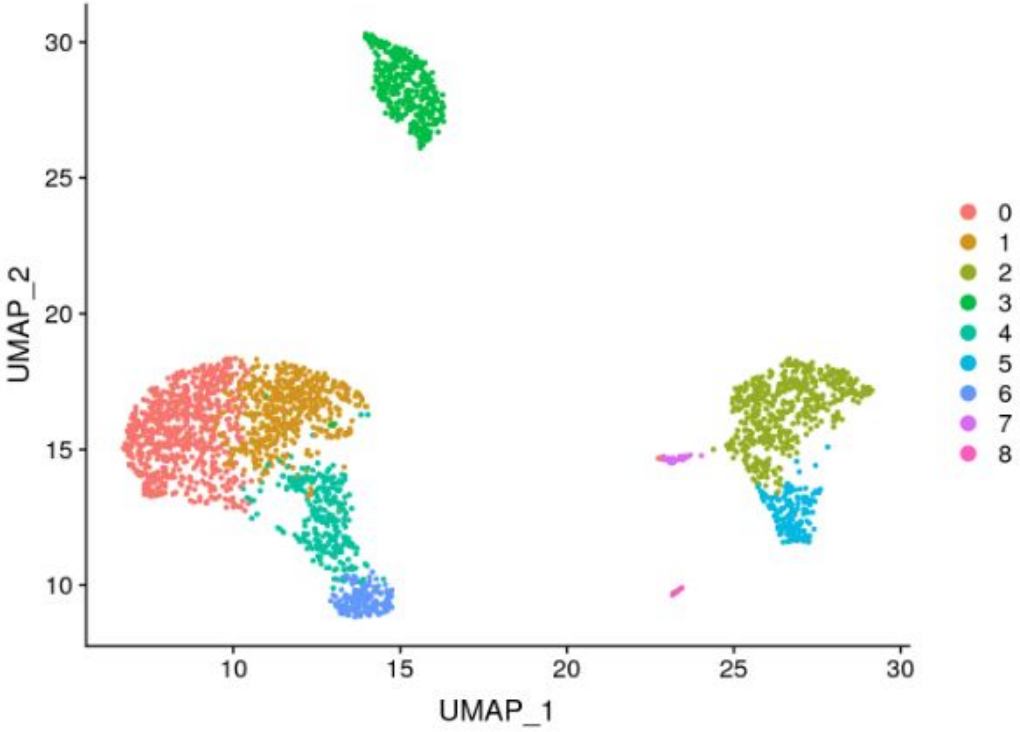
O tSNE tem grande sucesso, mas um concorrente de porte chegou em 2018:

UMAP

UMAP

- UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction
- “The UMAP algorithm is competitive with t-SNE for visualization quality, and arguably preserves more of the global structure with superior run time performance”

Independente de justificativas técnicas, o UMAP começou a ser muito adotado. Minha explicação para esse sucesso é simples: por que as visualizações são bonitas.


30
25
20
15
10
UMAP_2
10
15
20
25
30
UMAP_1
0
1
2
3
4
5
6
7
8

- Não há uma razão para escolha entre UMAP e tSNE que não de ordem estética (“fica mais bonito”) ou prática (“roda mais rápido”) .
- E o UMAP aparentemente roda mais rápido.
- Tem implementação em python com bons tutoriais: https://pypi.org/project/umap-learn/

== Resumão ==

- Por que reduzir dimensões?
  - Maior velocidade
  - Reduzir sobreajuste
  - Ver os dados

== Resumão ==

- PCA
  - A pedra fundamental da redução
  - Busca relações lineares
  - A escolha padrão para redução como pré-processamento

== Resumão ==

- tSNE
  - Uma alternativa para visualização que não é linear
  - Relativamente consolidado (2008)
  - Mais bonito e mais complicado que o PCA

== Resumão ==

- UMAP
    - Uma alternativa para visualização que não é linear
    - Mais bonito e mais complicado que o PCA
    - Mais rápido que o tSNE
    - Uso crescente (2018)

== Resumão ==

Todos com uso e implementação em Python. (R e Julia também)

Redução de Dimensionalidade para Visualização e Análise de Dados

Wesley Seidel
Tiago Lubiana