Dimension reduction

From modelling to visualization

Farnoosh Khodakarami

This material is prepared by Farnoosh khodakarami and Ali madani

Webinar outline

Introduction

1) Why do we need dimension reduction?
2) What are the widely-used dimension reduction methods

Dimension reduction in practice

1) Implementation in Python
2) Assumptions and parameters

DataSets

Features(Attribute/variable)

Data records (samples)

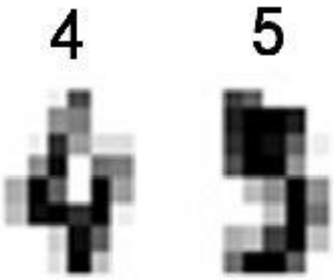
ID	Address	# Bed	#Bath	...	School Score	Year Build	Crime Rate
				...			
				...			

 # Features = Dimension of dataset

Number of dimensions in images

Number of dimensions (features) is equal to number of pixels if we use them directly as features of our models.

8*8=64 → 2048*1536=3,145,728



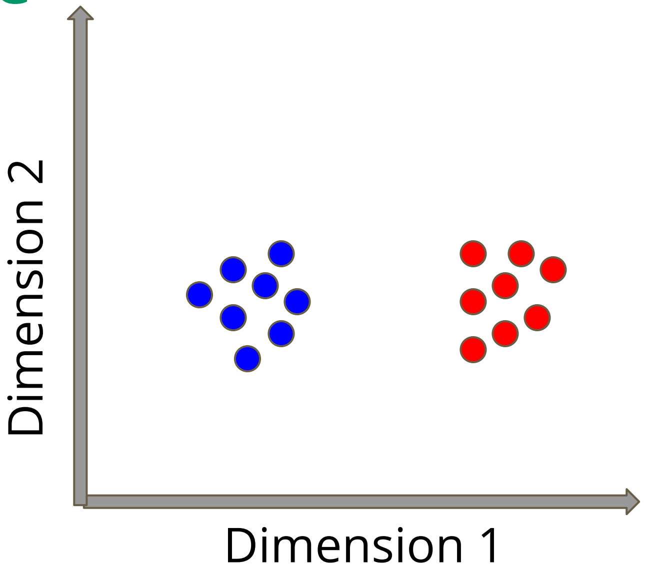
UCI ML hand-written digits

Why do we need to reduce number of dimensions?

- **May help to eliminate irrelevant features or reduce noise**

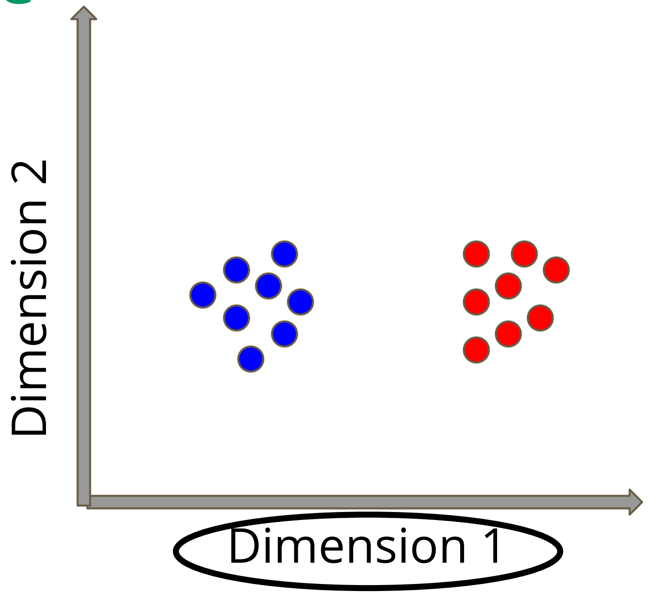
Why do we need to reduce number of dimensions?

- **May help to eliminate irrelevant features or reduce noise**



Why do we need to reduce number of dimensions?

- **May help to eliminate irrelevant features or reduce noise**



Why do we need to reduce number of dimensions?

- **May help to eliminate irrelevant features or reduce noise**
- **Reduce Time and Memory in computations**

Why do we need to reduce number of dimensions?

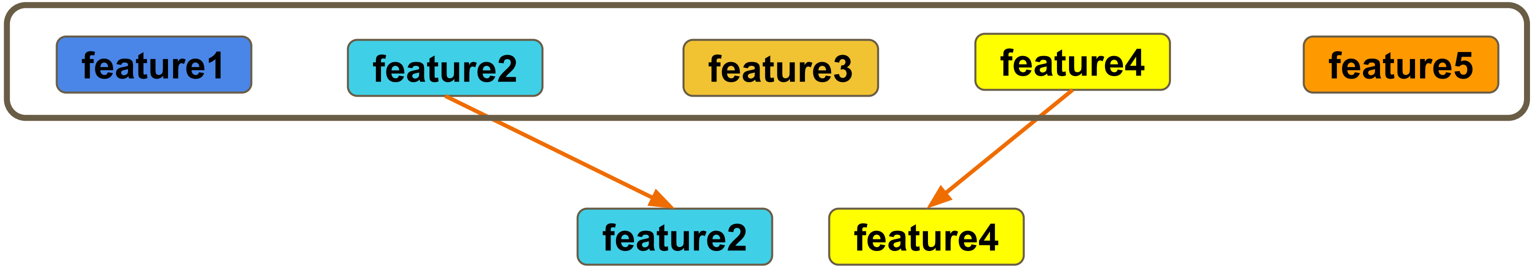
- **May help to eliminate irrelevant features or reduce noise**
- **Reduce Time and Memory in computations**
- **Allow data to be more easily visualized**

We can imagine things in 3D.

We can visualize, in an easy to interpret way, up to 2D.

Dimensionality Reduction

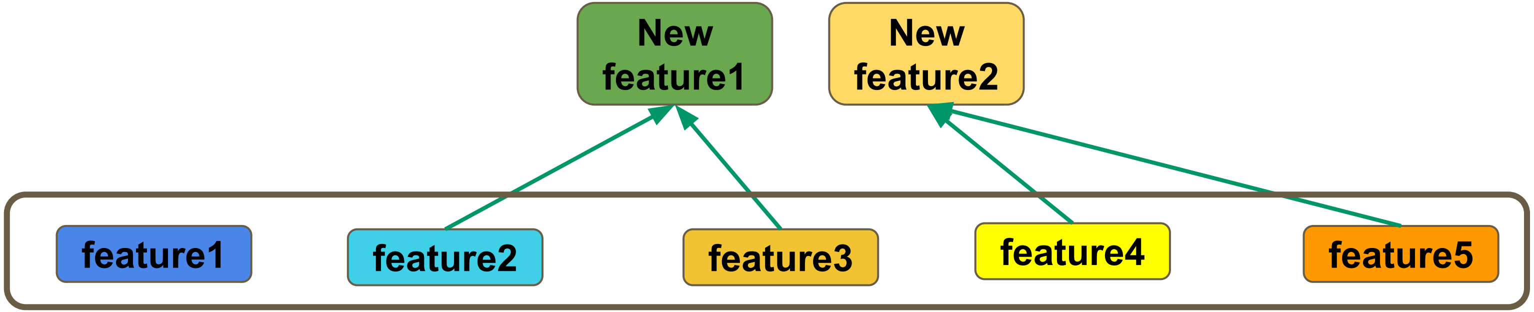
Dimensionality Reduction



Feature Selection

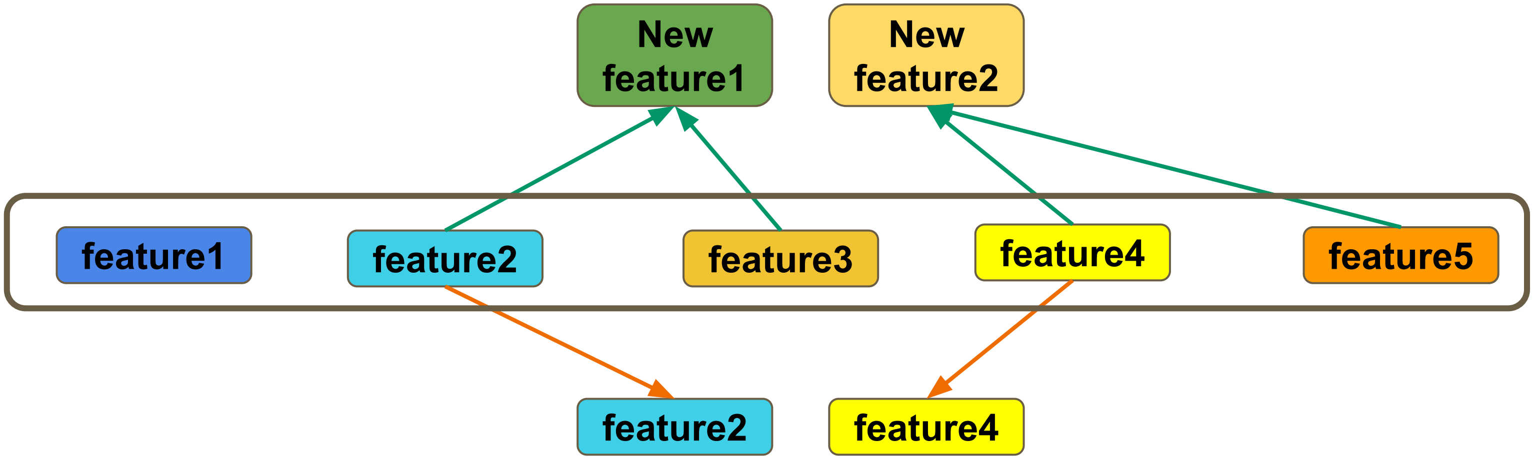
Dimensionality Reduction

Feature Extraction



Dimensionality Reduction

Feature Extraction



Feature Selection

Example of Feature Extraction

Risk of Diabetes

	Weight(lb)	**Hight(ft)**
Joe	170	6’
James	150	5’3’’

Example of Feature Extraction

Risk of Type 2 Diabetes

	Weight(lb)	**Hight(ft)**
Joe	170	6’
James	150	5’3’’

Is risk of type 2 diabetes higher for Joe?

Example of Feature Extraction

Risk of Type 2 Diabetes

	Weight(lb)	Hight(ft)
Joe	170	6’
James	150	5’3’’

	BMI
Joe	23.1
James	26.6

$$BMI = \frac{Weight\,(kg)}{[Height(m)]^2}$$

Example of Feature Extraction

Risk of Type 2 Diabetes

	BMI
Joe	23.1
James	26.6

Risk of type 2 diabetes is higher for James

Ganz, Michael L., et al. "The association of body mass index with the risk of type 2 diabetes: a case–control study nested in an electronic health records system in the United States." *Diabetology & metabolic syndrome* 6.1 (2014): 50.

Feature Extraction

1 Principal Component Analysis (PCA)

PHILOSOPHICAL TRANSACTIONS A

rsta.royalsocietypublishing.org

Review

CrossMark
click for updates

Cite this article: Jolliffe IT, Cadima J. 2016 Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc. A* **374**: 20150202.
http://dx.doi.org/10.1098/rsta.2015.0202

Principal component analysis: a review and recent developments

Ian T. Jolliffe[1] and Jorge Cadima[2,3]

[1]College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, UK
[2]Secção de Matemática (DCEB), Instituto Superior de Agronomia, Universidade de Lisboa, Tapada da Ajuda, Lisboa 1340-017, Portugal
[3]Centro de Estatística e Aplicações da Universidade de Lisboa (CEAUL), Lisboa, Portugal

Feature Extraction

1 Principal Component Analysis (PCA)

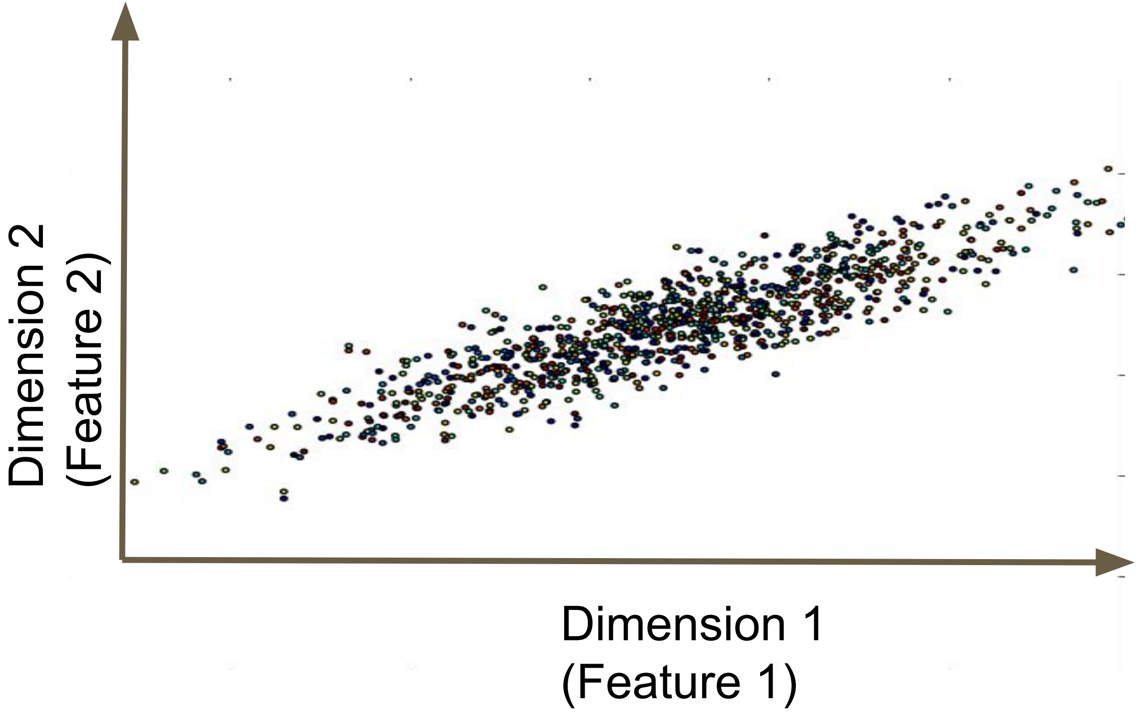
LIII. ***On lines and planes of closest fit to systems of points in space***

Karl Pearson F.R.S.

To cite this article: Karl Pearson F.R.S. (1901) LIII. *On lines and planes of closest fit to systems of points in space* , The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2:11, 559-572, DOI: 10.1080/14786440109462720

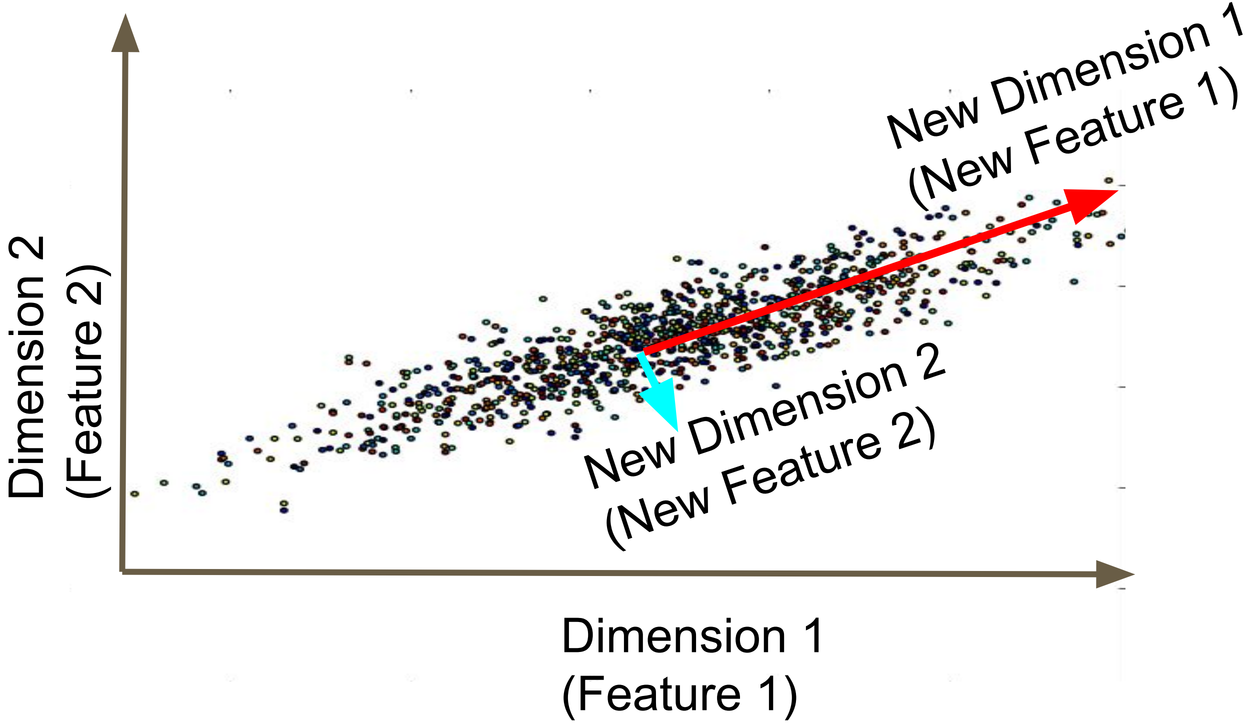
Feature Extraction

1 **Principal Component Analysis (PCA)**

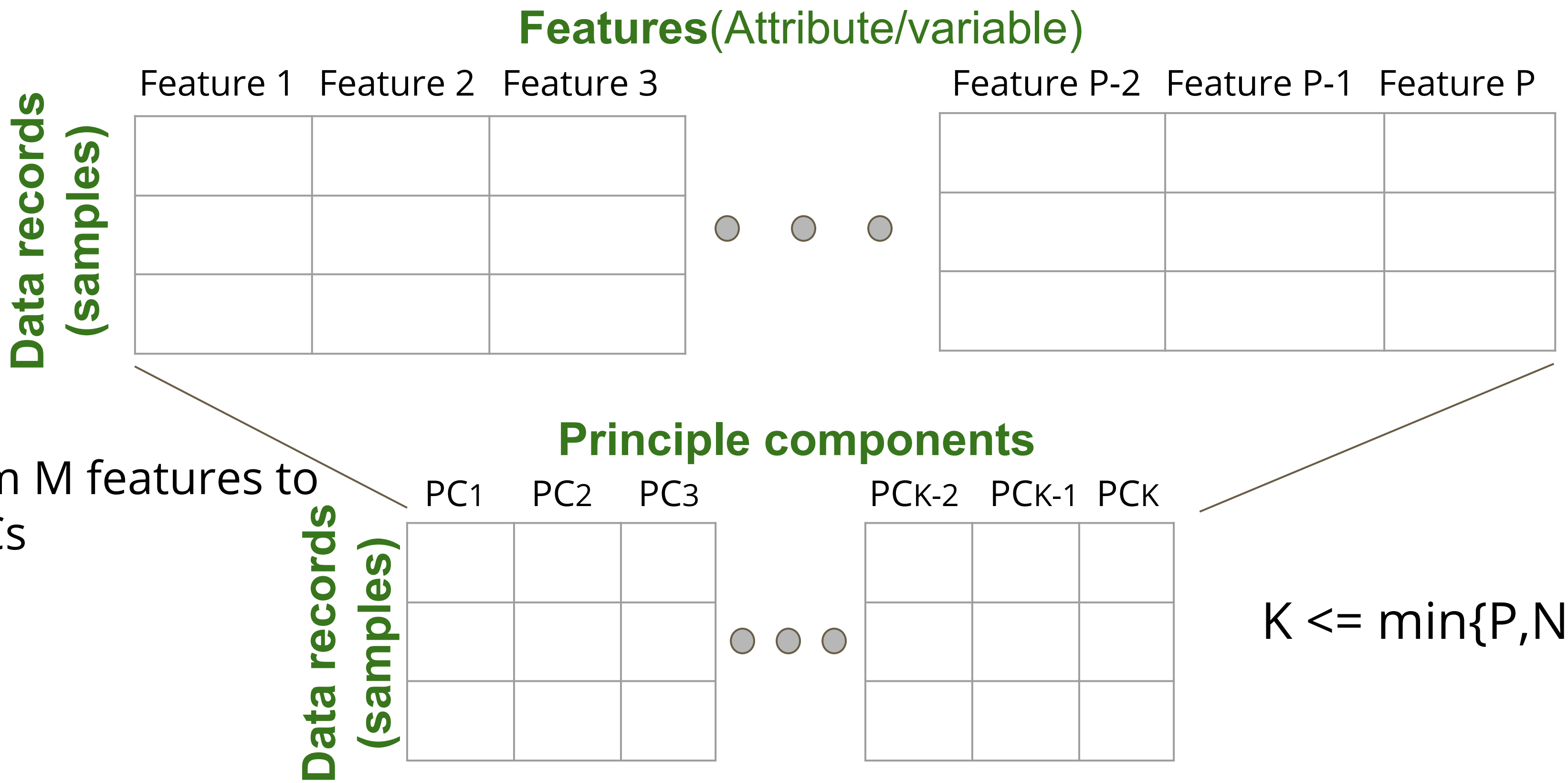


Feature Extraction

1 PCA: Principal Component Analysis



PCA: Principal Component Analysis



From M features to K PCs

K <= min{P,N}

Feature Extraction

2 t-SNE: t-Distributed Stochastic Neighbor Embedding

Journal of Machine Learning Research 9 (2008) 2579-2605 Submitted 5/08; Revised 9/08; Published 11/08

Visualizing Data using t-SNE

Laurens van der Maaten LVDMAATEN@GMAIL.COM
TiCC
Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands

Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

Editor: Yoshua Bengio

Amazing GitHub page

https://lvdmaaten.github.io/tsne/

t-SNE: t-Distributed Stochastic Neighbor Embedding

t-SNE gives you a feel or intuition of how the data is arranged in a high-dimensional space

PCA vs t-SNE:

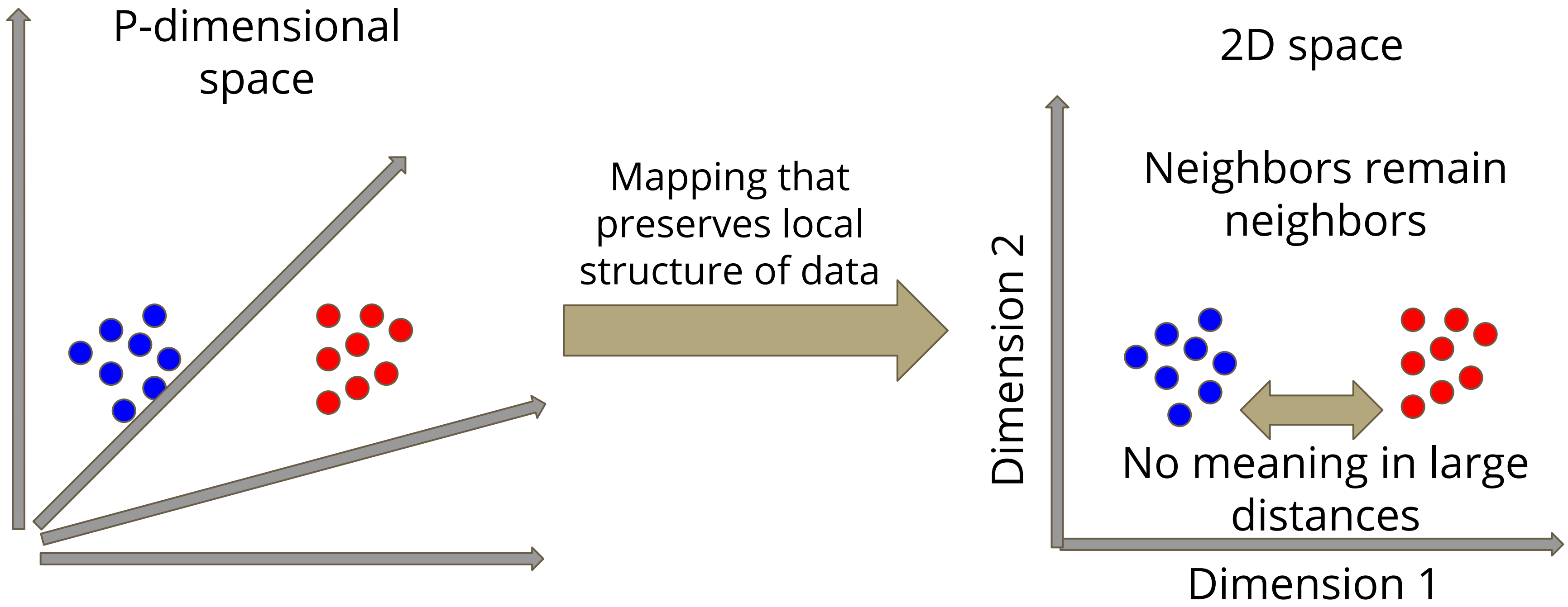
- PCA is a linear dimension reduction technique that seeks to maximize variance and preserves large pairwise distances.
- PCA can lead to poor visualization especially when dealing with non-linear manifold structures.
- t-SNE differs from PCA by preserving only small pairwise distances or local similarities

t_SNE

The t-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space

Tries to optimize these two similarity measures using a cost function.

t-SNE: t-Distributed Stochastic Neighbor Embedding



The embedding does not preserve global structure of data

Feature Extraction

3 UMAP: Uniform Manifold Approximation and Projection

UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

Leland McInnes
Tutte Institute for Mathematics and Computing
leland.mcinnes@gmail.com

John Healy
Tutte Institute for Mathematics and Computing
jchealy@gmail.com

James Melville
jlmelville@gmail.com

December 7, 2018

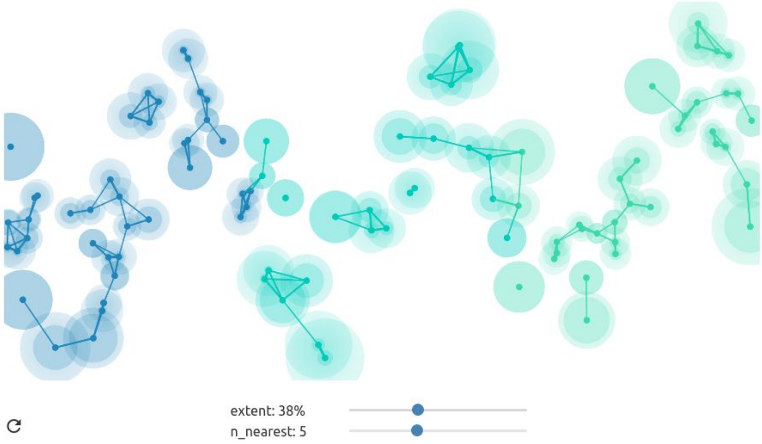
Amazing GitHub repository

https://github.com/lmcinnes/umap

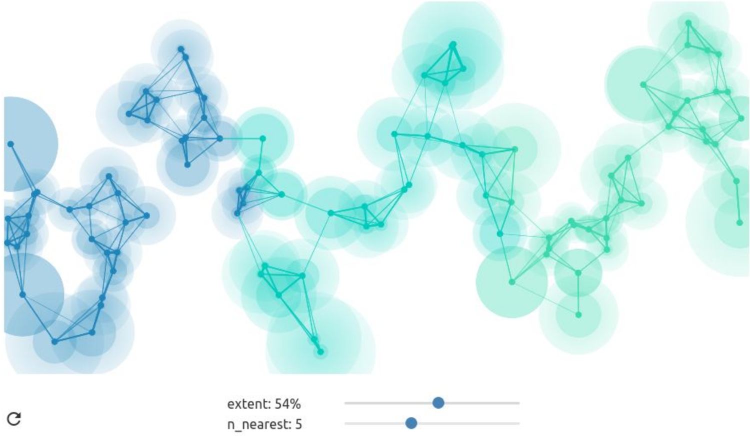
UMAP

UMAP constructs a high dimensional graph representation of the data then optimizes a low-dimensional graph to be as structurally similar as possible.

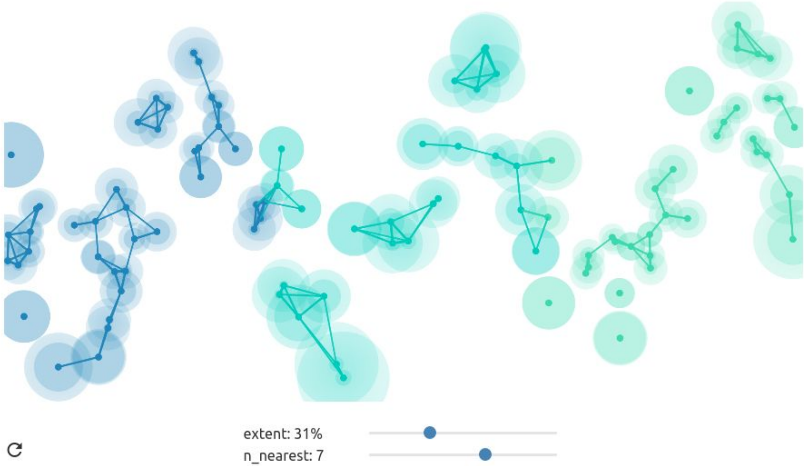
1) UMAP builds something called a "fuzzy simplicial complex".
   - This is really just a representation of a weighted graph, with edge weights representing the likelihood that two points are connected.
2) UMAP extends a radius outwards from each point, connecting points when those radii overlap.
3) Once the high-dimensional graph is constructed, UMAP optimizes the layout of a low-dimensional analogue to be as similar as possible.
   a) This process is essentially the same as in t-SNE, but using a few clever tricks to speed up the process.



extent: 38%
n_nearest: 5



extent: 54%
n_nearest: 5



extent: 31%
n_nearest: 7

UMAP vs t-SNE

- Both tSNE and UMAP were designed to predominantly preserve local structure

- UMAP is fast. It can handle large datasets and high dimensional data
  - For example, UMAP can project the 784-dimensional, 70,000-point MNIST dataset in less than 3 minutes, compared to 45 minutes for scikit-learn's t-SNE implementation.

- UMAP preserves more of the data global structure.

Feature Extraction

1. PCA: Principal Component Analysis
2. t-SNE: t-Distributed Stochastic Neighbor Embedding
3. UMAP: Uniform Manifold Approximation and Projection