CSCI E-82
Advanced Machine Learning, Data Mining & Artificial Intelligence
Lecture 1

Peter V. Henstock
Fall 2018

What to call me?

More polite

- ~~Most high excellency~~
- ~~* Sir *~~
- Professor Henstock
- Dr. Henstock
- ~~Professor Peter~~
- Professor
- ~~Mr. Henstock~~
- ~~Dr. Peter~~
- ~~Mr. Peter~~
- Peter
- ~~Prof-Man~~
- ~~You~~

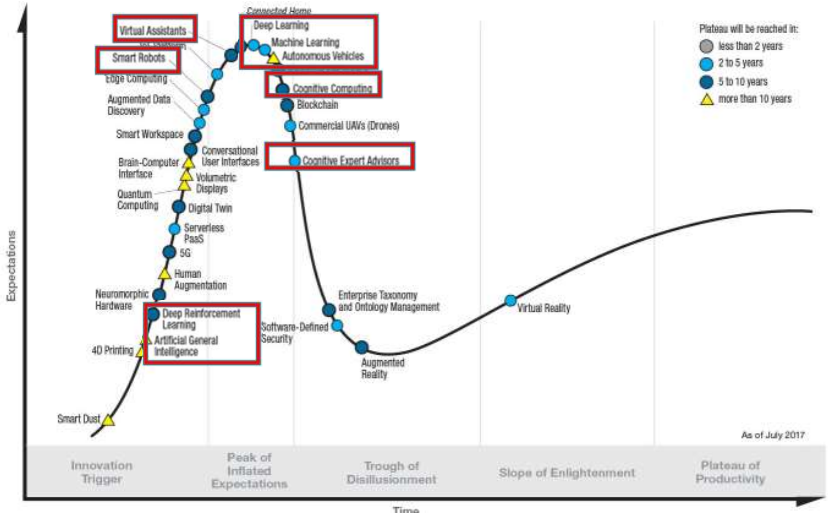
Less polite

Our Teaching Staff

Dave Dowey
Statistician & Economist
Based in Luxembourg

Rashmi Banthia
Data Scientist
Kaggle enthusiast
Software developer
Based in Atlanta

Gartner's Hype Cycle 2017



Harvard Extension School

- Elements of Data Science and Statistical Learning in R
- Fundamentals of Data Science
- Introduction to Data Science
- Python for Data Science
- Math for Computation and Data Science
- Big Data Systems
- Big Data in Healthcare Applications
- Principles of Big Data Processing
- Data Literacy in the Age of Machine Learning
- Building the Brain: A Survey of Artificial Language
- Deep Learning
- Networks

Who's Smarter than the Average Bear?

- https://en.wikipedia.org/wiki/Yogi_Bear#/media/File:Yogi_Bear_Yogi_Bear.png
- https://www.bostonglobe.com/metro/2015/11/17/yale-most-popular-course-harvard-class/OsJs36GYID8tViXKDUEaqM/story.html

What is intelligence?

What is intelligence?

- Ideally like to have a universal metric across all humans (and life forms)
- Use of language
- Use of tools
- Ability to acquire & apply knowledge and skills

What about language?

www.koko.org

http://voices.nationalgeographic.com/2013/04/24/elephants-communicate-in-sophisticated-sign-language-researchers-say/

Lev Vygotsky 1896-1934

- Russian psychologist
- Acquisition of new knowledge depends on previous learning and 'instruction'
- "Zone of proximal development" ZPD
  - Delta of what one can do with and w/o help
  - Learning through experiences in this zone
- Proposed using tools: uniquely human

https://en.wikipedia.org/wiki/Tool_use_by_animals




What is intelligence? [Wikipedia]

- Capacity for
  - Logic
  - Understanding
  - Self-awareness
  - Learning
  - Emotional knowledge
  - Planning
  - Creativity
  - Problem solving
- Ability to perceive information, and retain it as knowledge to be applied towards adaptive behaviors within an environment or context.

Learning…

- Can machines learn?
- Or is that a human trick?
- Or is more generally a biological skill?

Training People vs. Machines

- Classification problem: Aka vs. Ao
- Do parents teach by positive or negative examples?

Artificial Intelligence

- "The theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages" [Google]

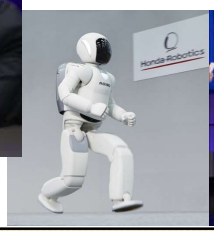
Strong vs. Weak AI

https://en.wikipedia.org/wiki/Data_(Star_Trek)
http://www.suggestkeyword.com/YzNwMA/
http://spectrum.ieee.org/automaton/robotics/humanoids/honda-robotics-unveils-next-generation-asimo-robot
http://wembleypark.com/discover/our-story/on-screen/

Strong vs. Weak AI

- Weak AI = everyday uses of AI
  - Dedicated to a specific task
  - Doesn't display human-like intelligence in general

http://www.cultofmac.com/285528/siri-may-helped-2-year-old-girl-just-save-mothers-life/
http://spectrum.ieee.org/automaton/robotics/humanoids/honda-robotics-unveils-next-generation-asimo-robot
http://www.geekwire.com/2013/ibm-takes-watson-cloud/
http://www.wired.com/2012/09/deep-blue-computer-bug/
https://techcrunch.com/2016/03/15/google-ai-beats-go-world-champion-again-to-complete-historic-4-1-series-victory/



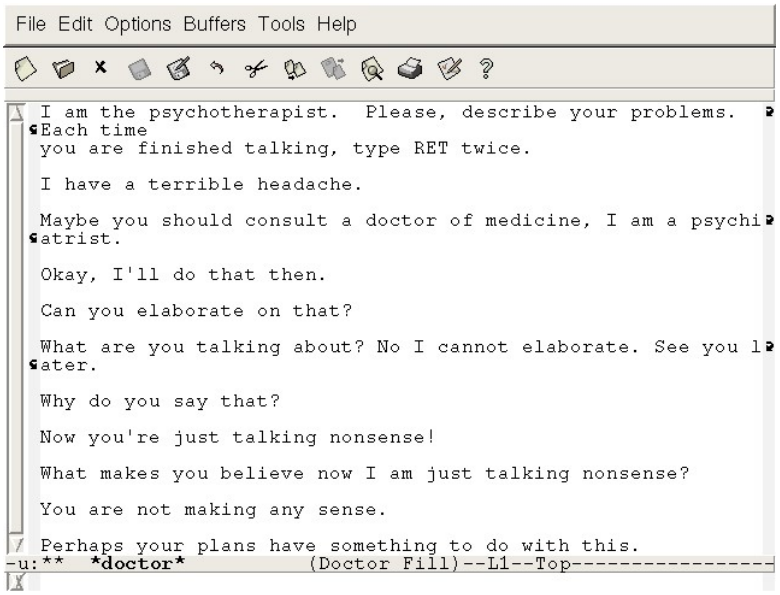
Artificial Intelligence

- Machine that mimics functionality we associate with human cognitive ability
- Machine that perceives its environment and takes actions to maximize its chance of success at some goal

AI vs. Humans

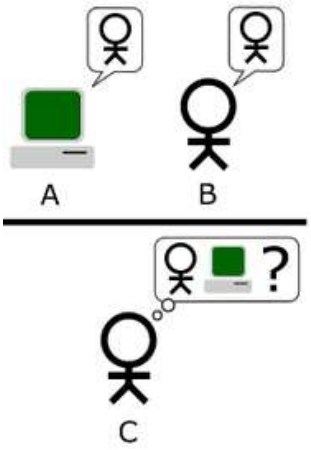
- Should machines and humans learn things in the same way?
- Is the journey important?
- Or does only the destination matter?

ELIZA



- 1964-1966 by Wizenbaum at MIT

Turing Test



The "standard interpretation" of the Turing Test, in which player C, the interrogator, is given the task of trying to determine which player – A or B – is a computer and which is a human. The interrogator is limited to using the responses to written questions to make the determination.

Wikipedia Image adapted from Saygin, 2000

Machine Learning

- Subfield of Computer Science
  - Part of Artificial Intelligence
- Sometimes subfield of Statistics
  - Computational statistics typically used for predictive modeling
  - Optimization and fitting

AI vs. Machine Learning

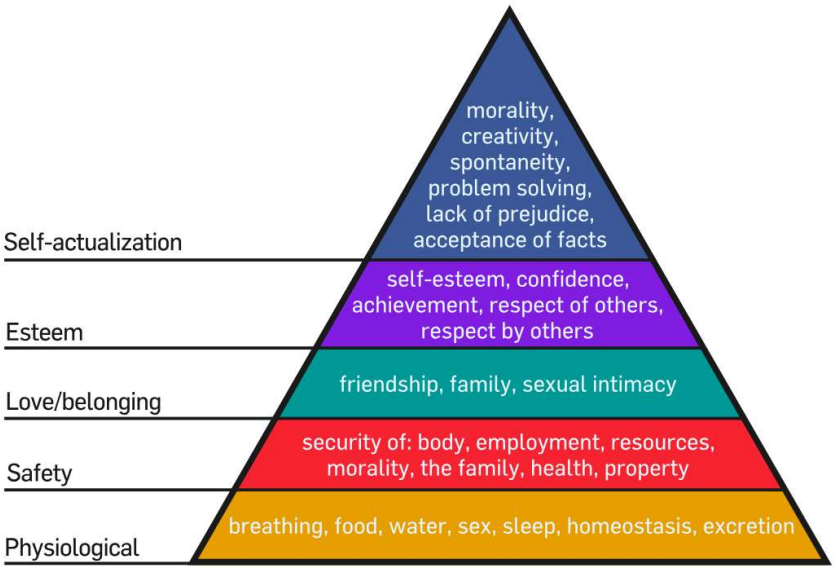
"…machine learning is the only viable approach to building AI systems that can operate in complicated real world environments"

Goodfellow, Bengio & Courville

Reason for Machine Learning

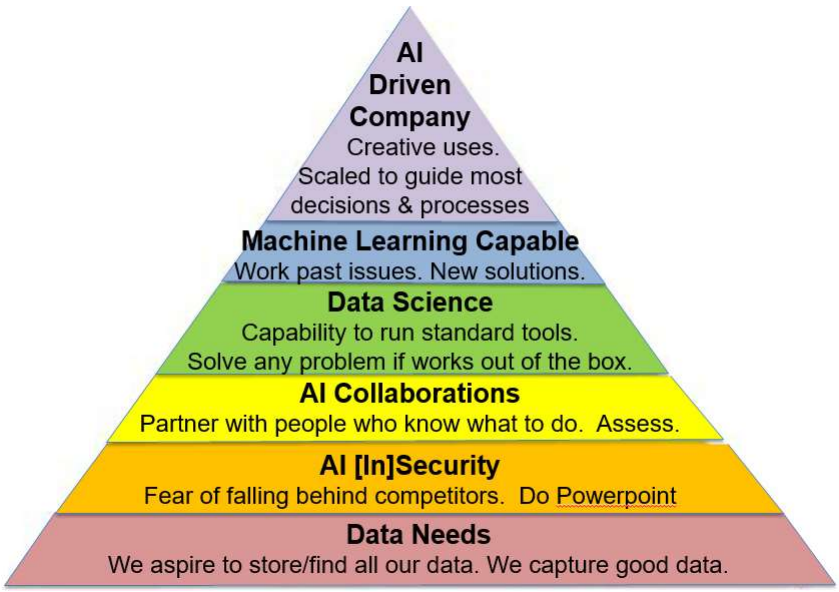
- Alternative forms of intelligence
  - Generation of rules
  - Hand-crafting rules
- Issues:
  - Representation
  - Ability to scale for real world

Maslow's Hierarchy of Needs



https://commons.wikimedia.org/wiki/File:Maslow%27s_Hierarchy_of_Needs.svg

Hierarchy of Needs



October 2012 HBR

Harvard Business Review

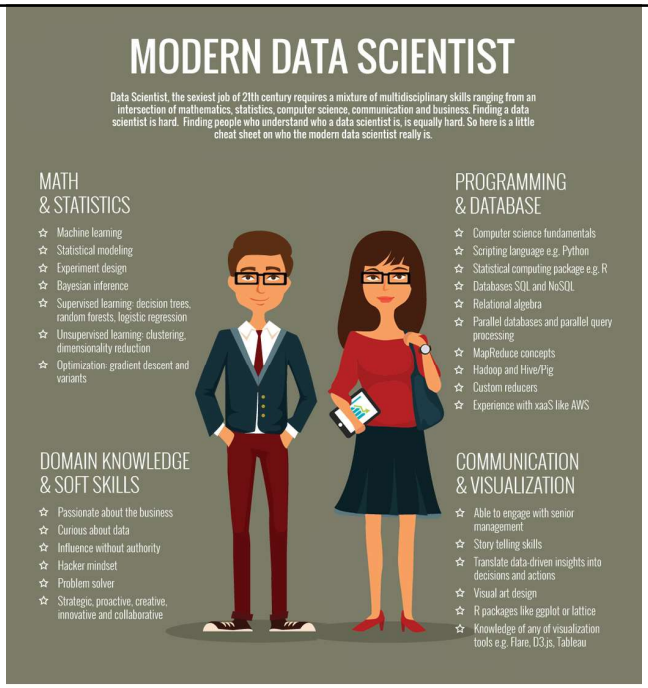
DATA

Data Scientist: The Sexiest Job of the 21st Century

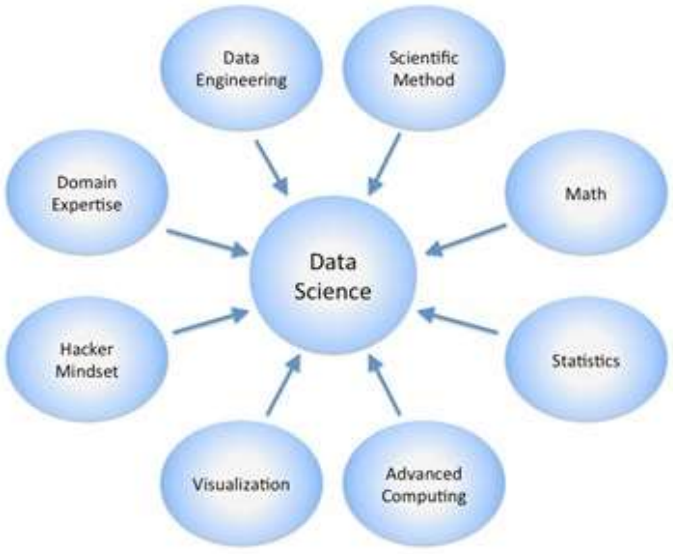
Data Science

- "A New Breed"
- "High-ranking professional with training"
- "traditional backgrounds of people you saw 10-15 years ago just don't cut it these days"
- "Hot job of the Decade"

- https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century/



Key Skills



- http://www.kdnuggets.com/2017/06/data-scientist-learn-skills-free.html

Course Structure

Elon Musk's Secret for Learning

- Master the core principles first
  - You can't learn what you can't connect
- Develop a schema of ideas
- Dive into advanced topics after mastering the fundamentals
- https://www.inc.com/jessica-stillman/heres-elon-musks-secret-for-learning-anything-fast.html

Ultimate Goal of the course

To empower you to make the world just a little bit better

Practical Goals of the Course

- Provide you with an understanding of various algorithms and approaches:
  - What works for what situations?
  - Why it works?
  - What are the limitations?
  - How will it scale?
- Practical skills to carry out solutions:
  - Python programming
  - Ability to navigate and use the python libraries
- Ability to go beyond data science:
  - To read, leverage the literature, and learn
  - To adapt and go beyond the baseline

Lectures Focus of Course

- Cover main techniques and variations
- Theory of how things work
- Limitations of the algorithms
- Computational aspects
- Simple examples of main types by hand
- Combination of:
  - Conceptual
  - Theoretical
  - Computational
- Examples of application domains
  - Text mining, recommendation systems, etc.

Section Focus of Course

- Hands-on practical aspects
- Use of python libraries
- Examples of applying machine learning
- Related examples to homework
- Questions/Answers

TA Tutorials (TBD)

- Overview of different current state-of-the-art tools used in the field
- Goals:
  - Give a quick dive into useful tools that may useful for your projects
  - Provide a perspective on what's available
  - Reduce the barrier of entry in getting started

Introduce our TAs

- David Dowey
  - Former statistician at NATO in Luxembourg
- Rashmi Banthia
  - SW engineer & Kaggle enthusiast in Atlanta, Georgia

Homework

- Individual homework (HW1)
  - Collaboration is not permitted
  - Scope is narrow
- Homework with a partner
  - Collaboration with one partner required
  - Scope is open with real problems that generally will require more time

Partner Logistics

- Seeking a good way of finding partners
  - Google Spreadsheet
- You will:
  - Register a partner before starting a project
  - Both submit the same homework
    - Staff will grade either arbitrarily
  - Both submit a partner assessment
    - We will provide a format
- You can change partners for future work
- We welcome ideas/feedback on this

Topic Presentations

- 3-person team presenting 1 topic
- 2 teams presenting top presentation presented weekly from week ~5
  - ~15 minutes to present (may change)
- Goals:
  - Learn about something you care about
  - Share your learning with the class for 15 min
  - Prepare original code to help others get started
  - Submit slides & code to share with class

What else do you have to do?

- Exam:
  - Unproctored
  - Timed: 2 hour window over ~48 hour period
  - Open-book
  - Open-computer
  - Not open-lifeline so no help
  - Concept-driven
- Exam will be ~11/15-18 (~Thu-Sun) before Thanksgiving in mid November

What else do you have to do?

- Paper presentation:
  - Choose a paper that's closely related to the lecture topics
  - Short presentation probably on YouTube
  - Currently re-thinking the approach due to the class size
  - Details will be forthcoming
- Paper summaries:
  - Practice to read and critique a few papers
  - Details will be forthcoming

What else do you have to do?

- Final Project
  - Choose and solve a problem with partner
  - Extend an existing method beyond the basics
  - Report and presentation
  - <10 min presentation 20+ groups→ 3+ hours
    - Some to present during the next to last class
    - Remainder present during a marathon class
    - Goal: learn from others' approaches

About Math

- Will show math and derive some equations to show where they came from
- Will you be required to do any derivations on tests? No; on homework? Unlikely
- Most mathematical parts of course are:
  - PCA using linear algebra
  - Optimization that often uses a gradient

Course Tools

- Canvas
  - All homework
  - Course announcements
  - Lecture and section notes/recordings
  - Email machinelearnmine@gmail.com for me
  - Please do not email staff using Canvas
- Piazza
  - All discussions and questions
  - Piazza has a “Note” and “Question” option
  - Please don’t post partial/full solutions publicly (private is fine)

About Homework

- All deadlines are EST
  - Canvas doesn't allow me to specify explicitly so just assume they refer to both Harvard day and time

Positive Review

It's clear Peter and his TAs thought deeply about how to present the course material, which had both huge breadth and huge depth. The course had an adequate amount of technical rigor, both mathematically and computationally, without it feeling like it was a course in algorithms or data structures. The course also surveyed the machine learning landscape fairly well, introducing computer vision, NLP, deep learning, and other popular techniques.

The sections made the class even better. During class, I found myself struggling to keep up with the mathematics and theories behind the algorithms. The sections are what brought them home for me. The Piazza board was good but became overwhelming for me. I appreciate the responsiveness of Peter and his TAs through the board and found that my issues were quickly clarified.

Negative Review

Ultimately though, the sheer amount of material covered in each lecture made it impossible to get strong intuition for anything; a single lecture covered enough material to fill out an entire semester's worth of studying, and fundamental concepts in ML were flown over and quickly forgotten a few minutes later. As such, this course is more of a survey course, and unless you put in (and even then) a very significant amount of extra self-study time, don't expect to come out of this course with a deep understanding of any concept you weren't already very familiar with.

I think it's a nice idea to want to provide actual intuition for ML concepts while covering a significant fraction of the field, but it's an impossible goal to achieve especially given the weak mathematical prerequisites given for the course. Intuition takes a long time spent over a single concept to develop, as well as a strong mathematical foundation in the case of ML. Given the prerequisites for the course and the sheer amount of material covered, it's effectively impossible to impart any long lasting true understanding of much of anything when you take into account the time required to complete assignments.

Finally, the expected number of hours given (12 hours/week) is completely unrealistic and was exceeded by at least a factor of 2 or 3 on almost each assignment after the first, and that's not counting study time to actually try to understand the algorithms used in the assignments; because simply watching lectures, as in any other field, is not enough to really learn anything. With the course as it stands, I would say that someone smart with a pretty strong math background and decent programming skills (but no ML knowledge) would need between 30 and 40 hours per week to get any kind of lasting, deep understanding for even some (maybe 15-20%) of the material. Everything else will be quickly completely forgotten as is the case with any kind of superficially acquired knowledge.

Is this course challenging?

- Large volume of conceptual information in lectures that you may not grasp the first time through
  - Grasp concepts, approaches and “how” → exam
- Apply the conceptual information and use the state of the art tools
  - Homework: simple then real problems
  - Hands-on experience in multiple domains
  - Final project
- Leverage field literature effectively
  - Read, understand and critique papers
  - Learn new areas of interest in small groups

Tentative Course Topics

- Dimensionality reduction
- Text mining
- Frequent item sets
- Regression
- Time series analysis
- Classification
- Clustering
- Deep learning
- Outlier analysis
- Network analysis
- Recommender systems
- Reinforcement learning

Python vs. R

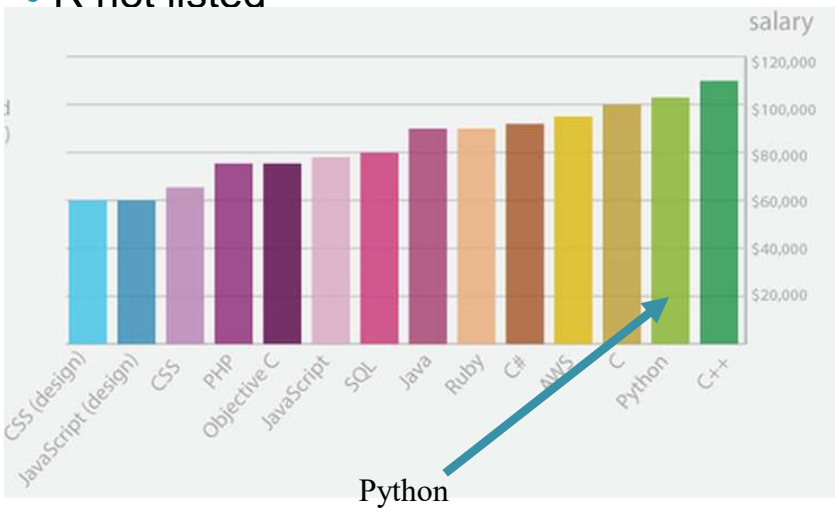
Why Python?

Language Rank	Types	Spectrum Ranking	Spectrum Ranking
1. Java		100.0	100.0
2. C		99.9	99.3
3. C++		99.4	95.5
4. Python		96.5	93.5
5. C#		91.3	92.4
6. R		84.8	84.8
7. PHP		84.5	84.5
8. JavaScript		83.0	78.9
9. Ruby		76.2	74.3
10. Matlab		72.4	72.8

- http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages

Salary by Language

- R not listed



Python vs. R Benchmarks

	Fortran gcc 4.8.2	Julia 0.3.7	Python 2.7.9	R 3.1.3	Matlab R2014a	Octave 3.8.1	Mathematica 10.0	JavaScript V8 3.14.5.9	Go go1.2.1	LuaJIT gsl-shell 2.3.1	Java 1.7.0_75
fib	0.57	2.14	95.45	528.85	4258.12	9211.59	166.64	3.68	2.20	2.02	0.96
parse_int	4.67	1.57	20.48	54.30	1525.88	7568.38	17.70	2.29	3.78	6.09	5.43
quicksort	1.10	1.21	46.70	248.28	55.87	1532.54	48.47	2.91	1.09	2.00	1.65
mandel	0.87	0.87	18.83	58.97	60.09	393.91	6.12	1.86	1.17	0.71	0.68
pi_sum	0.83	1.00	21.07	14.45	1.28	260.28	1.27	2.15	1.23	1.00	1.00
rand_mat_stat	0.99	1.74	22.29	16.88	9.82	30.44	6.20	2.81	8.23	3.71	4.01
rand_mat_mul	4.05	1.09	1.08	1.63	1.12	1.06	1.13	14.58	8.45	1.23	2.35

Figure: benchmark times relative to C (smaller is better, C performance = 1.0).

C compiled by gcc 4.8.2, taking best timing from all optimization levels (-O0 through -O3). C, Fortran and Julia use OpenBLAS v0.2.12. The Python implementations of `rand_mat_stat` and `rand_mat_mul` use NumPy (v1.8.2) functions; the rest are pure Python implementati

Benchmarks can also be seen here as a plot created with Gadfly.

- Low times are better
- http://julialang.org/

Python 2.7.9	R 3.1.3
95.45	528.85
20.48	54.30
46.70	248.28
18.83	58.97
21.07	14.45
22.29	16.88
1.08	1.63

What skills are needed for ML jobs?

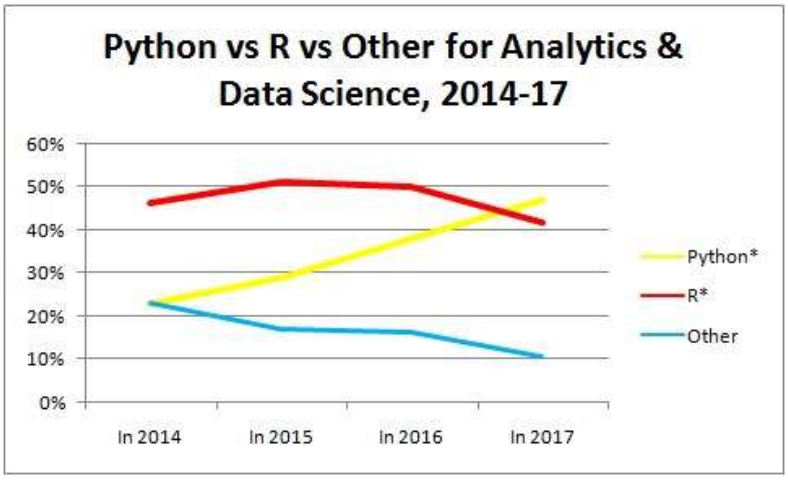
- http://www.quora.com/What-skills-are-needed-for-machine-learning-jobs

1) Python/C++/R/Java
   - Python lib, C++ for speed, Hadoop in Java
2) Probability and Statistics for understanding learning algorithms
3) Applied Math + Algorithms to understand how algorithms work
4) Distributed Computing
5) Unix tools
6) Hadoop sub-projects
7) Advanced signal processing for features
   - Wavelets, Fourier, etc.

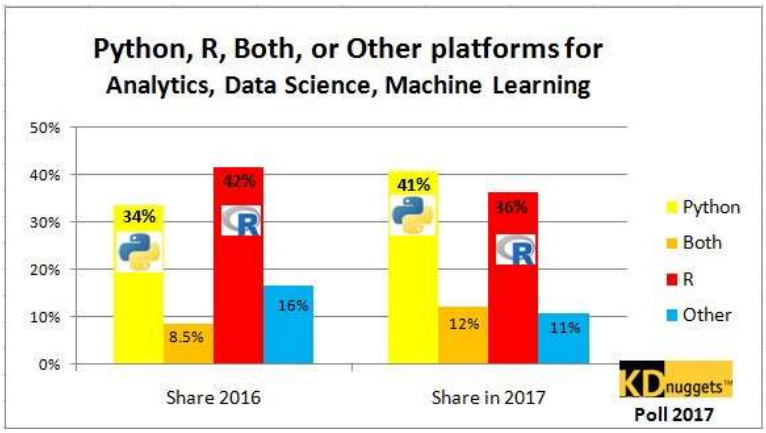
Real choice: Python vs. R

- Python
  - Easy-to-learn scripting language
  - Syntax is almost like pseudocode
  - Object-oriented
  - Good API manuals and IDEs
  - First class libraries for machine learning
  - Ideal of computer science folks
- R
  - Superior libraries with latest publications
  - Object oriented but ugly; e.g. few loops
  - Ideal for statisticians
  - API are cryptic; IDEs getting better

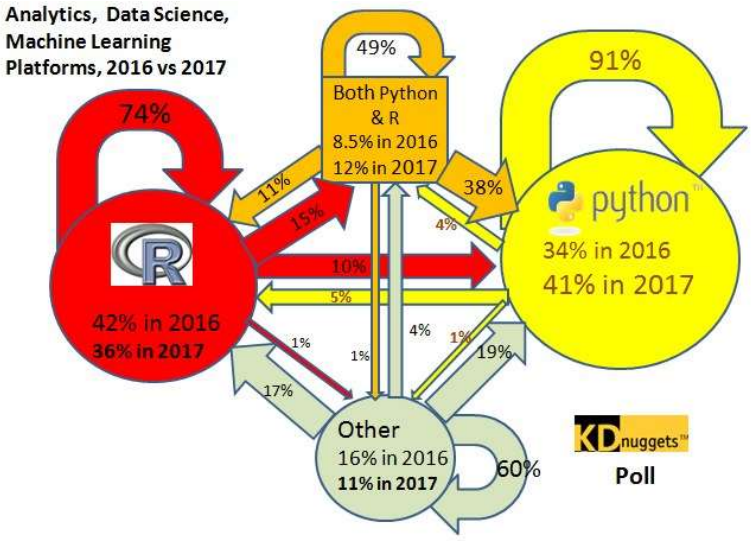
KDNuggets Poll



KDNuggets Poll



KDNuggets Poll



Python API

open(*file*, *mode='r'*, *buffering=-1*, *encoding=None*, *errors=None*, *newline=None*, *closefd=True*, *opener=None*)

Open *file* and return a corresponding *file object*. If the file cannot be opened, an `OSError` is raised.

file is either a string or bytes object giving the pathname (absolute or relative to the current working directory) of the file to be opened or an integer file descriptor of the file to be wrapped. (If a file descriptor is given, it is closed when the returned I/O object is closed, unless *closefd* is set to `False`.)

mode is an optional string that specifies the mode in which the file is opened. It defaults to `'r'` which means open for reading in text mode. Other common values are `'w'` for writing (truncating the file if it already exists), `'x'` for exclusive creation and `'a'` for appending (which on *some* Unix systems, means that *all* writes append to the end of the file regardless of the current seek position). In text mode, if *encoding* is not specified the encoding used is platform dependent: `locale.getpreferredencoding(False)` is called to get the current locale encoding. (For reading and writing raw bytes use binary mode and leave *encoding* unspecified.) The available modes are:

Character	Meaning
`'r'`	open for reading (default)
`'w'`	open for writing, truncating the file first
`'x'`	open for exclusive creation, failing if the file already exists
`'a'`	open for writing, appending to the end of the file if it exists
`'b'`	binary mode
`'t'`	text mode (default)
`'+'`	open a disk file for updating (reading and writing)
`'U'`	*universal newlines* mode (deprecated)

The default mode is `'r'` (open for reading text, synonym of `'rt'`). For binary read-write access, the mode `'w+b'` opens and truncates the file to 0 bytes. `'r+b'` opens the file without truncation.

- https://docs.python.org/3/library/functions.html#open

R API

`combine` *Combining Objects*

Description

Provides the S4 methods to combine several objects based on `itemMatrix` into a single object.

Note, use `union` rather than `c` to combine several mined `itemsets` (or `rules`) into a single set.

Usage

```
## S4 method for signature 'itemMatrix'
c(x, ..., recursive = FALSE)

## S4 method for signature 'transactions'
c(x, ..., recursive = FALSE)

## S4 method for signature 'rules'
c(x, ..., recursive = FALSE)

## S4 method for signature 'itemsets'
c(x, ..., recursive = FALSE)
```

Arguments

`x`	first object.
`...`	further objects of the same class as `x` to be combined.
`recursive`	a logical. If `recursive=TRUE`, the function recursively descends through lists combining all their elements into a vector.

Value

An object of the same class as `x`.

Can we use R/Matlab/Java/*?

- Requiring Python 3.7 for most of the homework assignments
  - Fast for prototyping
  - Currently the standard for data science
  - Libraries are excellent and standardized
- Free to choose any language for
  - Final project
  - Implementation for a paper presentation
  - Individual homework: no
  - Others? Default = no, but might consider…

Offer for a Few Top Students

- Some top past students have taken the course on a mission
  - Professors trying to solve a problem
  - Post-docs augmenting their research
  - Religious scholar interested in Torah
- Independently work with you to focus homework and project around your work
  - More work for these students and me
- Goal is publication of research

Linear Algebra Review

Linear Algebra Basics

- Capital letters for matrices

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

← second row

third column

Crix.org

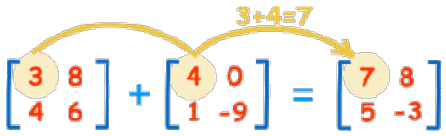
- Lower case for vectors
  - b = [2 3 5]
- Matrices and vectors have dimensions
  - [row, column] format
  - Usually top left corner is (1,1) not (0,0)

Matrix Math

- Addition

http://www.mathsisfun.com/algebra/matrix-introduction.html

  - Both matrices have to have the same size
  - Output has the same size as inputs

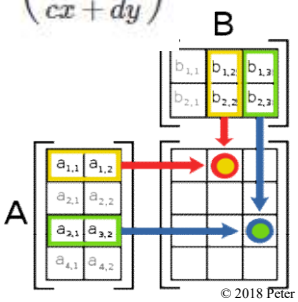


  - The sum at any position is the sum of the same position of the matrix values at that position

Matrix Math

- Multiplication
  - AxB matrix times BxC matrix → AxC matrix
  - If we reverse the matrices...
  - BxC matrix times AxB matrix → ?

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax+by \\ cx+dy \end{pmatrix} = \begin{pmatrix} ax+by \\ cx+dy \end{pmatrix}$$



http://rosalind.info/glossary/matrix-multiplication/

Larger Matrix Multiplication

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix}, \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

$$\mathbf{AB} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{pmatrix}$$

$$\mathbf{BA} = \begin{pmatrix} b_{11}a_{11} + b_{12}a_{21} + b_{13}a_{31} & b_{11}a_{12} + b_{12}a_{22} + b_{13}a_{32} \\ b_{21}a_{11} + b_{22}a_{21} + b_{23}a_{31} & b_{21}a_{12} + b_{22}a_{22} + b_{23}a_{32} \end{pmatrix}$$

Matrix Properties

- Diagonal Matrix
  - Top left down to bottom right (1,1)…(n,n)
  - Does it have to be square?

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Identity Matrix: $I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$
  - Diagonal with 1's

- Upper triangular matrix
  - 0's below the diagonal

$$\begin{bmatrix} 1 & 2 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

Determinant

- Operation that converts a square matrix to a number
- Easy for <= 3 operations else use software
- |A| or det(A)

$$det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

$$det(A) = det\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & k \end{pmatrix} = \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & k \end{vmatrix} = a\begin{vmatrix} e & f \\ h & k \end{vmatrix} - b\begin{vmatrix} d & f \\ g & k \end{vmatrix} + c\begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

https://math.dartmouth.edu/archive/m8s00/public_html/handouts/matrices3/node7.html

Matrix Transpose

- Transpose
  - Matrix A is mxn A' or A^t or A^T will be nxm
  - Every position (i,j) → (j,i)

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$ **Original matrix**

https://namito1358.wordpress.com/2013/04/12/calculations-of-a-33-matrix/

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^T \Rightarrow \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$$

  - Works for vectors too

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}^T = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix}$$

Vector Inner vs. Outer Product

- Outer Product

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^{\mathrm{T}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1 v_1 & u_1 v_2 & u_1 v_3 \\ u_2 v_1 & u_2 v_2 & u_2 v_3 \\ u_3 v_1 & u_3 v_2 & u_3 v_3 \\ u_4 v_1 & u_4 v_2 & u_4 v_3 \end{bmatrix}.$$

- Inner Product or "Dot Product"

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^{\mathrm{T}} \mathbf{v}$$

  - Defined as $\Sigma\, u_i * v_i$
  - What is dot product of u, v example above?

Principal Component Analysis

PCA Goal

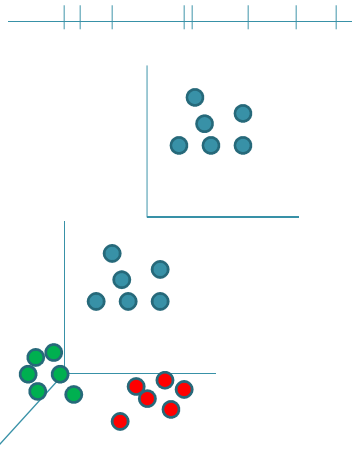
- Find a lower dimensional projection of the original data
- Projection aims to capture or preserve as much of the variance
  - Essentially a data compression

Purpose of PCA

- Unsupervised exploratory method
- Goals:
  - Reduce the full set of variables into a lower dimensional set
  - Visualization of 'proximities'
- Works through a transform from higher dimensional space into a lower one
  - Compression of data

Dimensionality Reduction

- What if you're trying to view k-dim data?
  - 1D:
  - 2D:
  - 3D:
  - 4D and above → need new trick

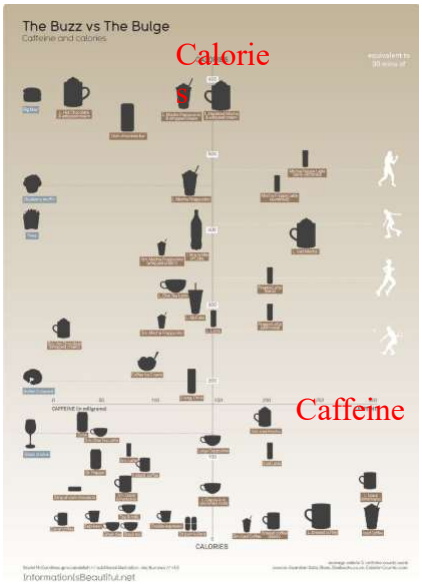


Beverages

- Sweet
- Carbonated
- Bitter
- Sour / acidic
- Watery
- Caffeinated
- Alcohol
- Full-bodied
- Hot/cold
- Tall
- White/creamy

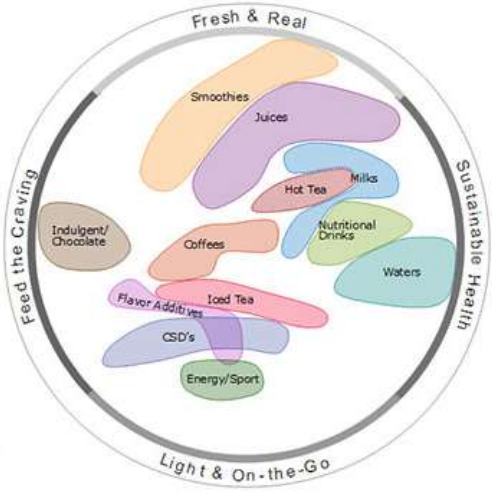
Beverages

- Sweet
- Carbonated
- Bitter
- Sour / acidic
- Watery
- Caffeinated
- Alcohol
- Full-bodied
- Hot/cold
- Tall
- White/creamy



Beverages

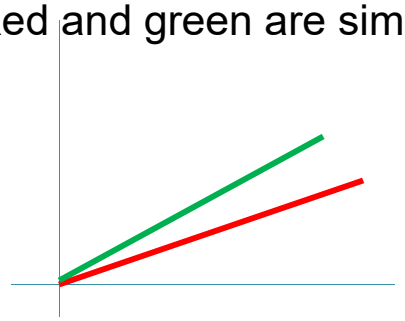
- Sweet
- Carbonated
- Bitter
- Sour / acidic
- Watery
- Caffeinated
- Alcohol
- Full-bodied
- Hot/cold
- Tall
- White/creamy



http://www.tnsglobal.com/us/growth-point/consumption-universe/beverages

Compression

- Red and green are similar in this space



- Using axes, how should we efficiently compress the data?

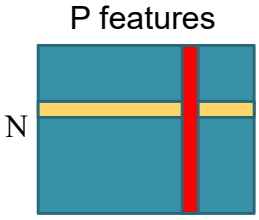
Compression

- Red and green are similar in this space

- Using axes, how should we efficiently compress the data?
- Want to find orthogonal unrelated features (red/blue) that describe different spaces

Data Matrix

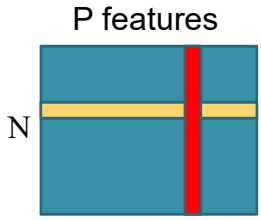
- X is NxP as usual
  - N instances
  - P variables or features

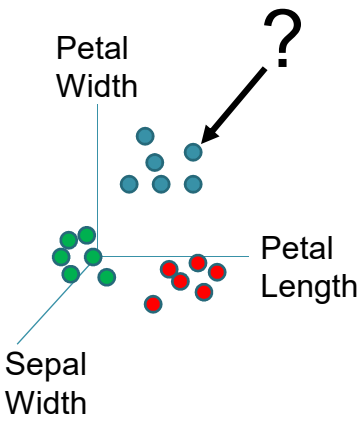


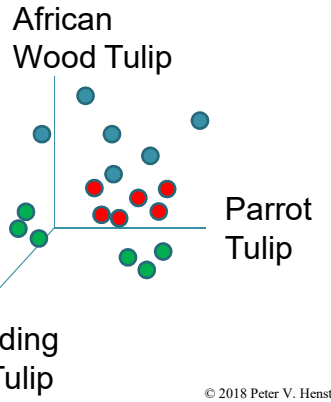
- Two ways of looking at the data:
  - Look at each row observation in P-dimensional space
  - Look at each column in N-dimensional space

Data Matrix

- Tulips x FlowerProperties

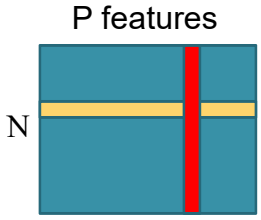


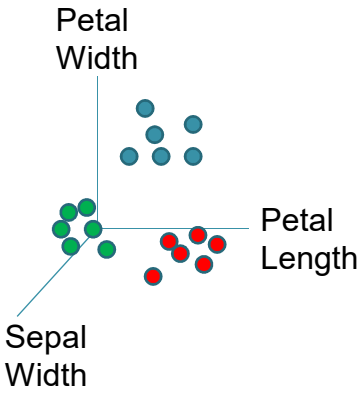


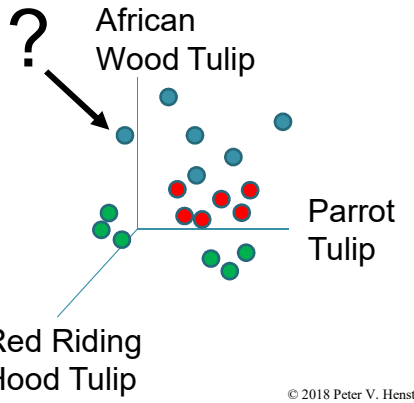


Data Matrix

- Tulips x FlowerProperties







How to choose reconstruction

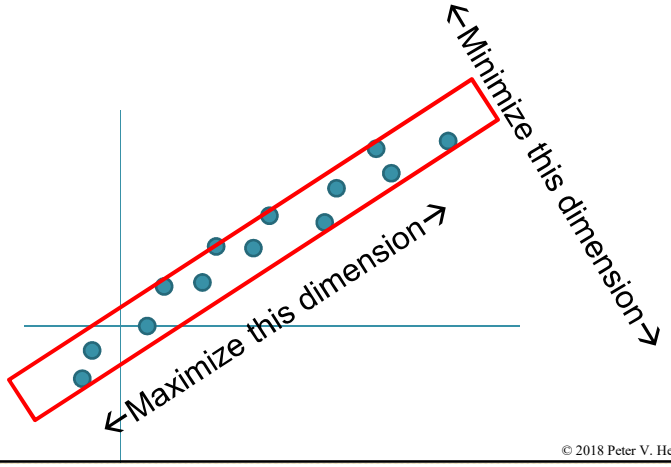
- We're compressing down to a lower dimensional "subspace"
- What criteria would you like to see in a lower dimensional subspace?
- How would you compare one projection to another?

How to choose reconstruction

- Trying to find lower dimensional subspace that will do these equivalent tasks:
  - Minimize error
  - Maximize variance captured
- Trying to make variance of the projection as wide as possible
- Centroid is hinge around which we rotate
- Not the variance of errors but variance of the projected points

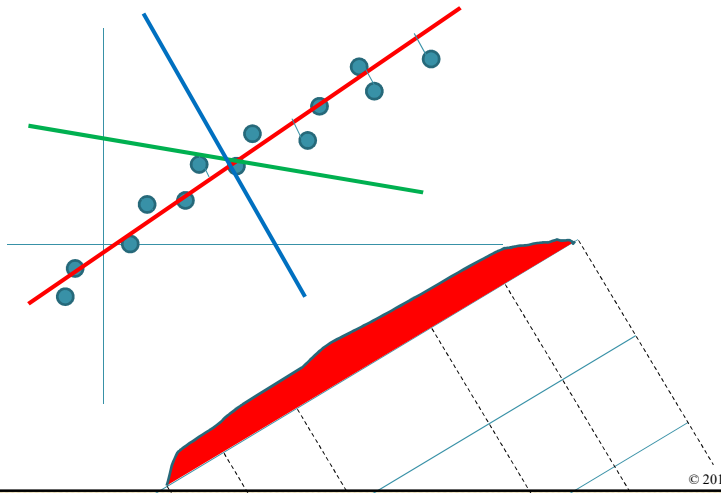
Principal Component Axis

- Preserves the best representation of the data as a single axis
- Minimizes the distortion



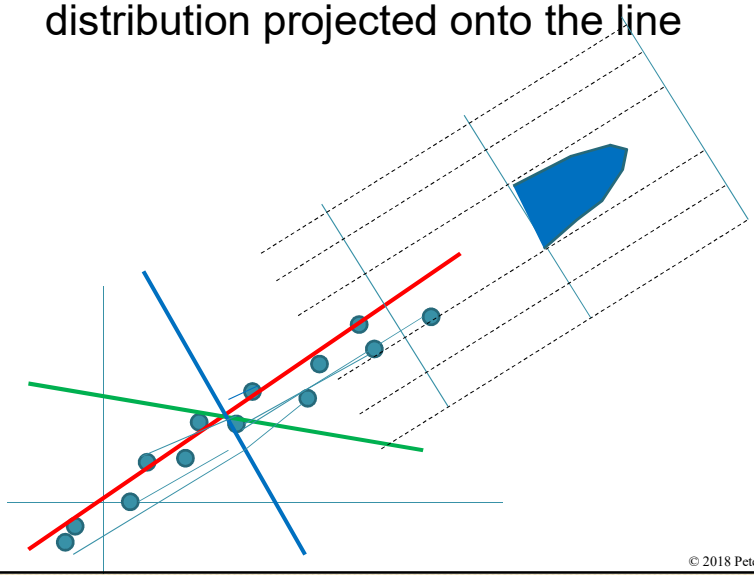
1st Principal Component Axis

- Another Idea: maximize variance of the distribution projected onto the line



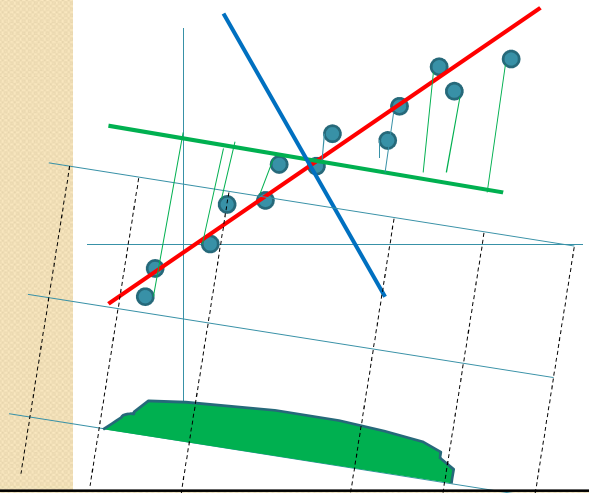
1st Principal Component Axis

- Another Idea: maximize variance of the distribution projected onto the line



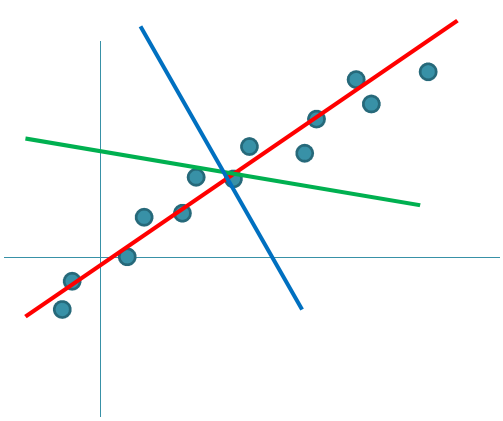
1st Principal Component Axis

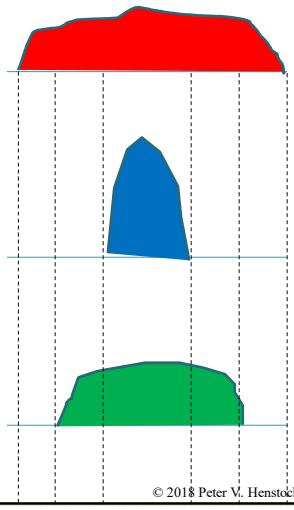
- Another Idea: maximize variance of the distribution projected onto the line



1st Principal Component Axis

- Another Idea: maximize variance of the distribution projected onto the line





Highest Variance Concept v.1

- Data compression tries to represent the information with fewer bits
  - Used for archival
  - Used to transmit fewer bits
- PCA tries to capture most of the information or “variance” by transforming the data
- PCA uses a subset of axes that best represent the data

Principal Components

- Create a weighted sum of features
- Use all the features together in one shot
- Each feature’ = $\Sigma_{featurej}$ weight(j) *feature(j)
- Weights are not equal
  - First several are much stronger than rest
  - Eliminate the lowest ranked transformations

PCA Goal

- Choose an axis for the first principal projection such that the difference between each point and projection is minimized
- Minimize Σ(Point – Projection)^2

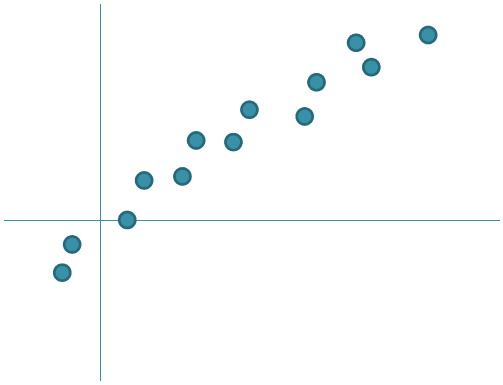
- Need to then choose subsequent projections that are orthogonal to original principal projection

PCA

- Create new vectors by projecting points onto a subspace spanned by them

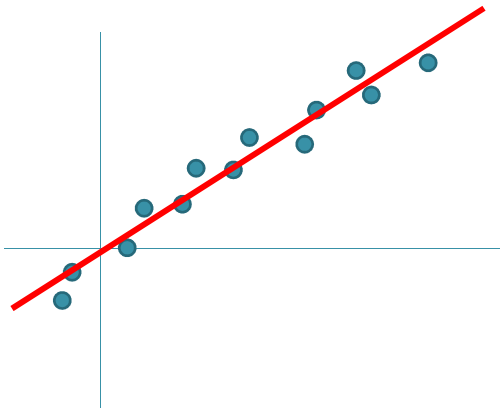
- X is the original data
- Y = A(x-μ) and A is a matrix

PCA onto 1 dimension



- Which line captures most of the variance in the data?

PCA onto 1 dimension



- How does the new line relate to the original X and Y axes?

PCA

- How can we find the dominant axis that captures most of the variation?
- How can we find an “orthogonal basis”?

PCA

- How can we find the dominant axis that captures most of the variation?
- How can we find an “orthogonal basis”?
  - How have we characterized the relationship between variables that allows us to describe them as similar?

PCA

- Use linear algebra to figure out the dominant axis and other orthogonal axes
- Use the covariance matrices
- Compute sample covariance matrix Σ
  - $\Sigma_{nxn} = \frac{1}{n-1}\Sigma_j(x_j - \mu)^T(x_j - \mu)$
- Compute the mean vector $\mu = \frac{1}{n}\Sigma_j x_j$

PCA Summary

- Start with our X matrix
- X has n instances and p columns
- De-mean (preferably $(x_i-\mu_i)/\sigma_i$ for all columns $i \in$ [1 to p]
- Find the dominant axis that explains most of the variance
- Sequentially add next dominant axis
  - Require next axis be orthogonal to all previous axes
- Assess how many axes we need

PCA Summary

- Start with our X matrix
- X has n instances and p columns
- De-mean (preferably $(x_i-\mu_i)/\sigma_i$ for all columns $i \in$ [1 to p]
- Find the dominant axis that explains most of the variance: hinged on centroid
- Sequentially add next dominant axis
  - Require next axis be orthogonal to all previous axes: hinged on centroid
- Assess how many axes we need

Linear Algebra solution

- Find a projection space
  - Find the dominant axis
  - Find subsequent orthogonal axes
- Figure out how many axes we need

PCA Steps

- Compute the mean of data
- Compute the covariance matrix of data
- Compute eigenvectors and eigenvalues of the covariance matrix
- Choose only the top k eigenvalues
- Project points onto the new subspace
  - $z = A(x-\mu)$ where x is not the x-axis but the 2-dimensional original data points

Eigenvalues & Eigenvector

- A is an nxn matrix
  - Covariance matrix
- x is a non-zero vector called eigenvector
- λ is called an eigenvalue if $Ax = \lambda x$
- Eigenvalues λ will help us choose how many dimensions we need
- Eigenvectors create the new transform space

Eigenvalues

1) Multiply n x n Identity matrix by λ
2) Subtract 1) from matrix
3) Compute the determinant of 2)
4) Solve for λ that satisfy det(A- λI) = 0
5) Solve for vectors associated with each eigenvalue λ

Eigenvalues & Eigenvectors

- $A = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix}$

1) Multiply n x n Identity matrix by λ

- $\lambda I = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$

2) Subtract 1) from matrix

- $A - \lambda I = \begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix}$

Eigenvalues & Eigenvectors

3) Compute the determinant of 2)

- $\det\begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix} =$
- $= (7-\lambda)(\text{-}1\text{-}\lambda) - 3(3) =$
- $=-7-7\lambda+\lambda^2-9$
- $=\lambda^2-6\lambda+-16=(\lambda\text{ -}8)(\lambda\text{ +}2)=0$
-

4) Solve for λ that satisfy det(A- λI) = 0

- Eigenvalues are 8 and -2

Compute Eigenvectors

5) Solve for vectors associated with each eigenvalue λ that are 8 and -2)

- $A-\lambda I=\begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix}$
- $A-8I=\begin{bmatrix} 7-8 & 3 \\ 3 & -1-8 \end{bmatrix}=\begin{bmatrix} -1 & 3 \\ 3 & -9 \end{bmatrix}$
- Solve means set $(A-\lambda I)x=0$
  - $\begin{bmatrix} -1 & 3 \\ 3 & -9 \end{bmatrix}x=\begin{bmatrix} -1 & 3 \\ 3 & -9 \end{bmatrix}\begin{bmatrix} x1 \\ x2 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
  - Add Row1 + 1/3Row2
  - $\begin{bmatrix} -1 & 3 \\ 0 & -3 \end{bmatrix}\begin{bmatrix} x1 \\ x2 \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ → $-x_1+3x_2=0$
  - $x_1=3x_2$ and we can take $x_2=1$, $x_1=3$ → $\begin{bmatrix} 3 \\ 1 \end{bmatrix}$

Compute Eigenvectors

5) Solve for vectors associated with each eigenvalue λ that are 8 and -2)

- $A - \lambda I = \begin{bmatrix} 7-\lambda & 3 \\ 3 & -1-\lambda \end{bmatrix}$
- $A + 2I = \begin{bmatrix} 7+2 & 3 \\ 3 & -1+2 \end{bmatrix} = \begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix}$
- Solve means set $(A - \lambda I)x = 0$
  - $\begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix} x = \begin{bmatrix} 9 & 3 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
  - Add Row1 - 3Row2
  - $\begin{bmatrix} 9 & 3 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ → $9x_1+3x_2 = 0$
  - x_1=-x_2/3; we can take x_2=1, x_1=-3 → $\begin{bmatrix} 1 \\ -3 \end{bmatrix}$

Checking eigenvectors

- Ax = λx
- $A = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix}$ $x = \begin{bmatrix} 3 & 1 \\ 1 & -3 \end{bmatrix}$
- $Ax = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & -3 \end{bmatrix} = \lambda x = \begin{bmatrix} 8 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 1 & -3 \end{bmatrix}$
- $Ax = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \lambda x$ $\begin{bmatrix} 24 \\ 8 \end{bmatrix} = 8 \begin{bmatrix} 3 \\ 1 \end{bmatrix}$
- $Ax = \begin{bmatrix} 7 & 3 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -3 \end{bmatrix} = \lambda x$ $\begin{bmatrix} -2 \\ 6 \end{bmatrix} = -2 \begin{bmatrix} 1 \\ -3 \end{bmatrix}$

Properties of Eigenvalues

- Singular matrices have some $\lambda = 0$
- Eigenvectors of real symmetric matrices with distinct eigenvalues are orthogonal
- If A is square matrix & λ is eigenvalue of A
  - λ is an eigenvalue of A^T
  - $\alpha\lambda$ is an eigenvalue of α A
  - λ^k is an eigenvalue of A^k for k integer ≥ 0
  - λ^{-1} is an eigenvalue of A^{-1}

Computing Eigenvalues & Eigenvectors

- Jacobi
- QR Decomposition
- Single Value Decomposition
- Power iteration method
  - Lanczos iterative approach

SVD = Single Value Decomposition

- A = covariance decomposes into USV'
- U = columns are eigenvectors
- S = diagonal matrix of eigenvalues
- V = rows are eigenvectors
- U'U = I_{nxn} V'V = I_{pxp}
- For symmetric matrix, U = V'

Process

X vectors with 4 features

```
0.0344   0.4387   0.3816   0.7655
0.7952   0.1869   0.4898   0.4456
0.6463   0.7094   0.7547   0.2760
0.6797   0.6551   0.1626   0.1190
0.4984   0.9597   0.3404   0.5853
0.2238   0.7513   0.2551   0.5060
0.6991   0.8909   0.9593   0.5472
0.1386   0.1493   0.2575   0.8407
0.2543   0.8143   0.2435   0.9293
0.3500   0.1966   0.2511   0.6160
```

Covariance matrix A

```
 0.0712   0.0165   0.0350  -0.0485
 0.0165   0.0949   0.0227  -0.0126
 0.0350   0.0227   0.0660  -0.0130
-0.0485  -0.0126  -0.0130   0.0614
```

SVD of (cov(A)) → U,S,V

U columns are eigenvectors

```
-0.5930   0.3879  -0.0341  -0.7048
-0.4779  -0.8362  -0.2654  -0.0453
-0.4527  -0.0221   0.8286   0.3287
 0.4638  -0.3871   0.4918  -0.6271
```

S diagonal are eigenvalues in decreasing order

```
0.1491        0        0        0
     0   0.0820        0        0
     0        0   0.0496        0
     0        0        0   0.0128
```

V columns are eigenvectors

```
-0.5930   0.3879  -0.0341  -0.7048
-0.4779  -0.8362  -0.2654  -0.0453
-0.4527  -0.0221   0.8286   0.3287
 0.4638  -0.3871   0.4918  -0.6271
```

Check: USV'

```
 0.0712   0.0165   0.0350  -0.0485
 0.0165   0.0949   0.0227  -0.0126
 0.0350   0.0227   0.0660  -0.0130
-0.0485  -0.0126  -0.0130   0.0614
```

Purge Eigenvalues

U

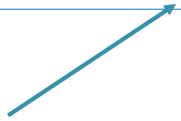
-0.5930	0.3879	-0.0341	-0.7048
-0.4779	-0.8362	-0.2654	-0.0453
-0.4527	-0.0221	0.8286	0.3287
0.4638	-0.3871	0.4918	-0.6271

S

0.1491	0	0	0
0	0.0820	0	0
0	0	0.0496	0
0	0	0	0.0128

V'

-0.5930	-0.4779	-0.4527	0.4638
0.3879	-0.8362	-0.0221	-0.3871
-0.0341	-0.2654	0.8286	0.4918
-0.7048	-0.0453	0.3287	-0.6271



- What columns/rows will be affected if we zero the S[4,4] of 0.0128?
- Note A = USV'

Purge Eigenvalues

-0.5930	0.3879	-0.0341	-0.7048
-0.4779	-0.8362	-0.2654	-0.0453
-0.4527	-0.0221	0.8286	0.3287
0.4638	-0.3871	0.4918	-0.6271

S

0.1491	0	0	0
0	0.0820	0	0
0	0	0.0496	0
0	0	0	0.0128

V'

-0.5930	-0.4779	-0.4527	0.4638
0.3879	-0.8362	-0.0221	-0.3871
-0.0341	-0.2654	0.8286	0.4918
-0.7048	-0.0453	0.3287	-0.6271

- What columns/rows will be affected if we zero the S[4,4] of 0.0128?

US

US*V'

Purge Eigenvalues

U

-0.5930	0.3879	-0.0341	-0.7048
-0.4779	-0.8362	-0.2654	-0.0453
-0.4527	-0.0221	0.8286	0.3287
0.4638	-0.3871	0.4918	-0.6271

S

0.1491	0	0	0
0	0.0820	0	0
0	0	0.0496	0
0	0	0	0.0128

V'

-0.5930	-0.4779	-0.4527	0.4638
0.3879	-0.8362	-0.0221	-0.3871
-0.0341	-0.2654	0.8286	0.4918
-0.7048	-0.0453	0.3287	-0.6271

- What if we zero the lowest eigenvalue?

0.0648	0.0161	0.0379	-0.0541
0.0161	0.0949	0.0229	-0.0130
0.0379	0.0229	0.0646	-0.0104
-0.0541	-0.0130	-0.0104	0.0563

Cov approximation with 3 eigenvalues

0.0712	0.0165	0.0350	-0.0485
0.0165	0.0949	0.0227	-0.0126
0.0350	0.0227	0.0660	-0.0130
-0.0485	-0.0126	-0.0130	0.0614

Original Cov

Purge Eigenvalues

U

-0.5930	0.3879	-0.0341	-0.7048
-0.4779	-0.8362	-0.2654	-0.0453
-0.4527	-0.0221	0.8286	0.3287
0.4638	-0.3871	0.4918	-0.6271

S

0.1491	0	0	0
0	0.0820	0	0
0	0	0.0496	0
0	0	0	0.0128

V'

-0.5930	-0.4779	-0.4527	0.4638
0.3879	-0.8362	-0.0221	-0.3871
-0.0341	-0.2654	0.8286	0.4918
-0.7048	-0.0453	0.3287	-0.6271

- What if we zero 2 lowest eigenvalues?

0.0648	0.0156	0.0393	-0.0533
0.0156	0.0914	0.0338	-0.0065
0.0393	0.0338	0.0306	-0.0306
-0.0533	-0.0065	-0.0306	0.0444

Cov approximation with 2 eigenvalues

0.0524	0.0422	0.0400	-0.0410
0.0422	0.0340	0.0322	-0.0330
0.0400	0.0322	0.0305	-0.0313
-0.0410	-0.0330	-0.0313	0.0321

Cov with 1 eigenvalues

Purge Eigenvalues

U

-0.5930	0.3879	-0.0341	-0.7048
-0.4779	-0.8362	-0.2654	-0.0453
-0.4527	-0.0221	0.8286	0.3287
0.4638	-0.3871	0.4918	-0.6271

S

0.1491	0	0	0
0	0.0820	0	0
0	0	0.0496	0
0	0	0	0.0128

V'

-0.5930	-0.4779	-0.4527	0.4638
0.3879	-0.8362	-0.0221	-0.3871
-0.0341	-0.2654	0.8286	0.4918
-0.7048	-0.0453	0.3287	-0.6271

0.0712	0.0165	0.0350	-0.0485
0.0165	0.0949	0.0227	-0.0126
0.0350	0.0227	0.0660	-0.0130
-0.0485	-0.0126	-0.0130	0.0614

Original Cov

0.0648	0.0161	0.0379	-0.0541
0.0161	0.0949	0.0229	-0.0130
0.0379	0.0229	0.0646	-0.0104
-0.0541	-0.0130	-0.0104	0.0563

Cov Approximation 1 eigenvalues

0.0648	0.0156	0.0393	-0.0533
0.0156	0.0914	0.0338	-0.0065
0.0393	0.0338	0.0306	-0.0306
-0.0533	-0.0065	-0.0306	0.0444

Cov with 2 eigenvalues

0.0524	0.0422	0.0400	-0.0410
0.0422	0.0340	0.0322	-0.0330
0.0400	0.0322	0.0305	-0.0313
-0.0410	-0.0330	-0.0313	0.0321

Cov with 3 eigenvalues

Projection into new space

- A = USV' from the SVD equation
- We assume that we A is already zero-mean
- Columns of U corresponding to the top eigenvalues
- X*U[,1:k] → projection into PCA space
- Each new projection is weighted sum (by U column) of the original X's features

Process

X*U[,1:k] → projection
If k = 2, what dimensionality will we have?

X vectors with 4 features

0.0344	0.4387	0.3816	0.7655
0.7952	0.1869	0.4898	0.4456
0.6463	0.7094	0.7547	0.2760
0.6797	0.6551	0.1626	0.1190
0.4984	0.9597	0.3404	0.5853
0.2238	0.7513	0.2551	0.5060
0.6991	0.8909	0.9593	0.5472
0.1386	0.1493	0.2575	0.8407
0.2543	0.8143	0.2435	0.9293
0.3500	0.1966	0.2511	0.6160

Process Check

X*U[,1:k] → projection
If k = 2, what dimensionality will we have?

X vectors with 4 features

0.0344	0.4387	0.3816	0.7655
0.7952	0.1869	0.4898	0.4456
0.6463	0.7094	0.7547	0.2760
0.6797	0.6551	0.1626	0.1190
0.4984	0.9597	0.3404	0.5853
0.2238	0.7513	0.2551	0.5060
0.6991	0.8909	0.9593	0.5472
0.1386	0.1493	0.2575	0.8407
0.2543	0.8143	0.2435	0.9293
0.3500	0.1966	0.2511	0.6160

SVD of (cov(A)) → U,S,V
U columns are eigenvectors

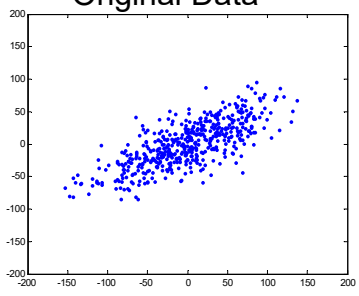
-0.5930	0.3879	-0.0341	-0.7048
-0.4779	-0.8362	-0.2654	-0.0453
-0.4527	-0.0221	0.8286	0.3287
0.4638	-0.3871	0.4918	-0.6271

X matrix 10 x 4 * U matrix is 4x2 after reduction

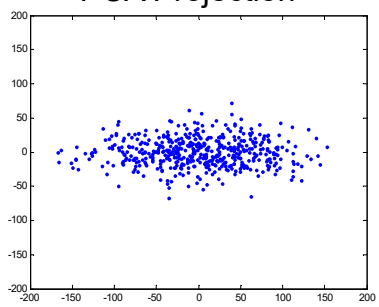
Projection

- Generated a random blob and rotated it by $\pi/6$ (Left)
- 1st PC X-axis; 2nd PC is Y-axis
- Dominant PC axis captures the variation

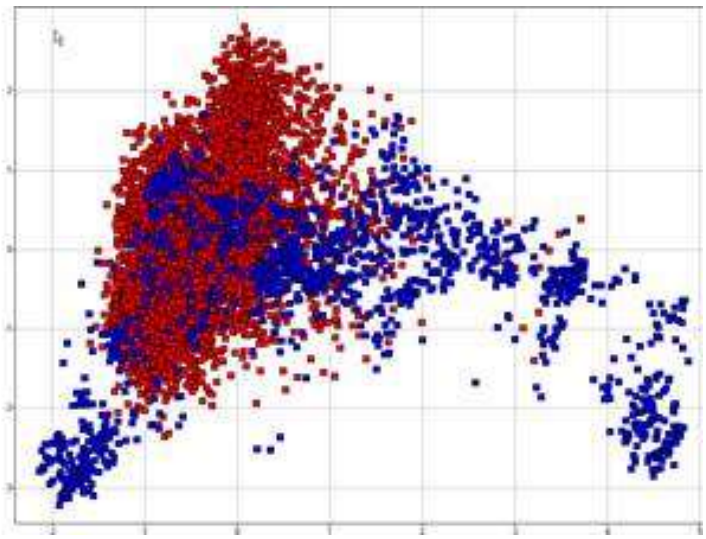
Original Data



PCA Projection



2000 dimensional space → 2



How many dimensions?

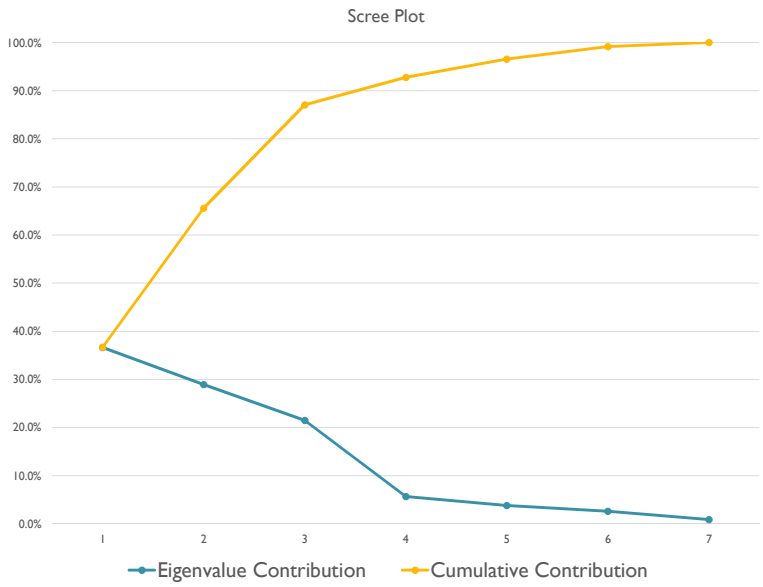
- For all symmetric matrices
  - Eigenvalues are real and never complex
- Covariance is real & positive semi-definite
- For positive semi-definite matrices
  - Eigenvalues are ≥ 0
- We can easily order the real eigenvalues for covariance matrices
  - Order eigenvalues in decreasing order
- Fraction of variance for eigenvalue k:
  - $\lambda_k / \Sigma_i \lambda_i$

Choosing number of eigenvalues

- Reduce P dimensions to K where P>K

Number	Eigenvalue	Percent	Cumulative Percent
1	21.3	36.7%	36.7%
2	16.8	28.9%	65.6%
3	12.5	21.5%	87.1%
4	3.3	5.7%	92.8%
5	2.2	3.8%	96.6%
6	1.5	2.6%	99.1%
7	0.5	0.9%	100.0%

Scree Plot



Scree Plot

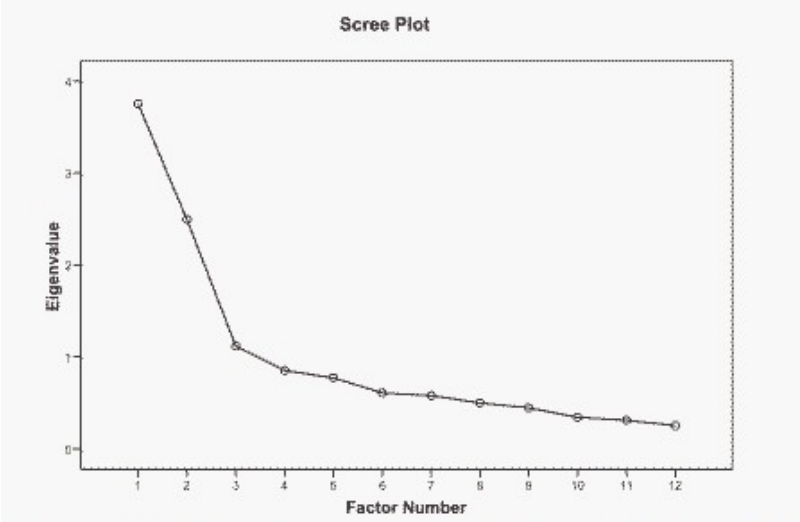


Figure 1. Scree Plot for the EFA for the 12 Y/GPI Scales Administered in Brazil

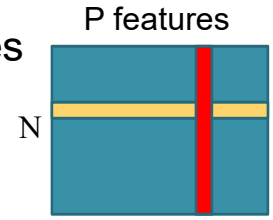
- http://jalt.org/test/bro_30.htm

PCA References

- Good tutorial that I've found
  - http://arxiv.org/pdf/1404.1100.pdf
- Numerous videos on YouTube
- Many articles in every field

Correlation Circle

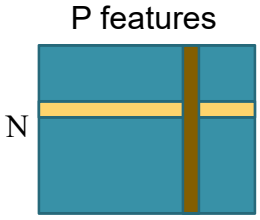
- Take each of the P features
- Take PC1 and PC2



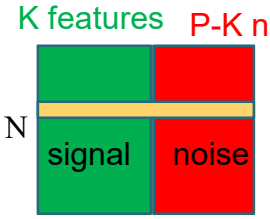
- X-axis = correlation(X:,p) row and PC1
- Y-axis = correlation(X:,p) row and PC2
- Provides an interpretation of PC space
- Shows correlations of features

Correlation Circle

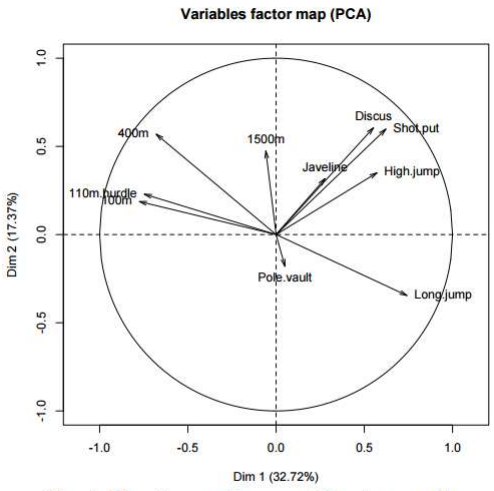
- Data matrix X is N x P



- PCA maps X into new space that has a full set of P features but



Correlation Circle



Variables' graph considering only active variables

Heptathlon data

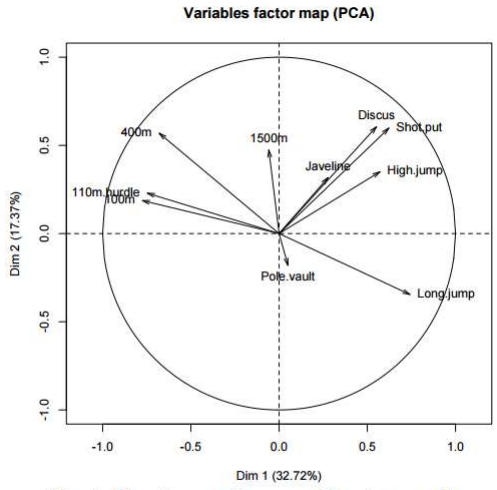
x.k = Event Individual

z.i = PCA component i

The correlation circle describes the correlation between the vector x.k and z.1 on the first axis and the correlation between x.k and z.2 on the second axis. The angle between two arrows represent the correlation of the respective variables. There is no linear dependence if the angle is 90 degrees.

http://www.statistik.tuwien.ac.at/public/filz/students/seminar/ws1011/hoffmann_ausarbeitung.pdf

Correlation Circle

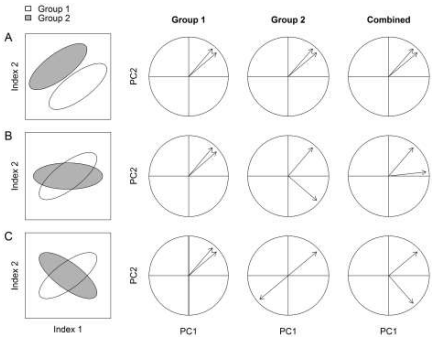


Variables' graph considering only active variables

Projecting the arrows onto the first dimension we can see that Long.jump, 100m and 110m.hurdle are most important for the first principal component. The arrows of 100m and Long.jump are nearly on a straight line, so they are negatively correlated, i.e. high scores in Long.jump are strongly correlate to short time in 100m.

http://www.statistik.tuwien.ac.at/public/filz/students/seminar/ws1011/hoffmann_ausarbeitung.pdf

More on Correlation Circles



Simplified scenarios illustrating how correlation circles can be interpreted when relationships between variables among groups are similar (a), dissimilar (b), or opposite (c).

- The first vertical panel consists of a scatterplot showing the correlation between two indices in two groups (e.g. species, individuals or months). The next two panels show the resulting correlation circles for the two groups separately. The fourth panel shows the relationship given when the two groups are combined, and highlights how important patterns may be diluted (b), or missed (c). In the correlation circles, vectors are loadings for the two indices of immune function. The angle between two vectors gives the degree of correlation (adjacent=highly correlated, orthogonal (90°)=uncorrelated, and opposite (180°)=negatively correlated). http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3079723/figure/pone-0018592-g001

Correlations with angles

- Dot product $\langle x_j, x_k \rangle = ||x_j||\ ||x_k||\ \cos(\theta_{jk})$
- If the variables are centered, then the cosine distance corresponds to the correlation between x_j and x_k

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- We're saying that centered variables will have the xbar = 0 and ybar =0

Continuing the logic…

- Like to find sequentially a set of new variables that maximizes the
- $argmax \sum_{k=0}^{P} r(vector, x_k)^2$
- Maximizing over all vectors
- This essentially defines the best principal components

PCA for original feature selection?

- We have lots of features
- We can use a PCA to map to a lower dimensional space

- Can we use PCA for feature selection?
- Can we select high-value features?
- Can we purge low-value features?

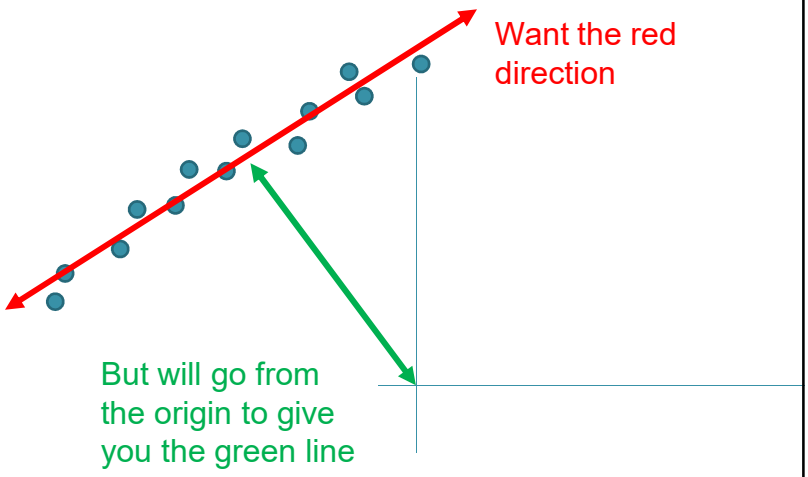
PCA for original feature selection?

- We have lots of features
- We can use a PCA to map to a lower dimensional space

- Can we infer which of the original features are less meaningful?

- PCA1 = w11*feat1 + w12*feat2 + …
- PCA2 = w21*feat1 + w22*feat2 + ..
- PCA3 = w31*feat1 + w32*feat2 + …

Why do you subtract the mean?



T-SNE

"Dandelions" & "Train Wrecks"

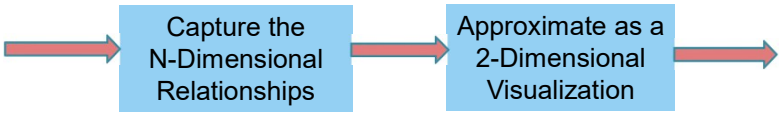
Preserves local distances

Preserves global distances

Facebook Connections

kipedia.org/wiki/Taraxacum
.roundsconsulting.com/2016/10/12/choosing-train-wreck-dumpster-fire/
.net/2014/04/facebook-data-collection-and-photo-network-visualization-with-gephi-and-r/#iLightbox[79713ef84a664b9d4dd]/0
http://kateto.net/2014/04/facebook-data-collection-and-photo-network-visualization-with-gephi-and-r/#iLightbox[d4dd84cf3af192994c5]/0

How t-SNE class of solutions works



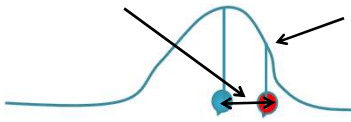
Distance matrix

Optimization

How t-SNE Works

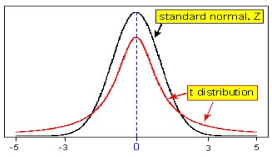
- Represent ND data with pairwise distances
  - Standardize distances to a normal distribution

N-dim distance on X axis



Standardized distance on Y. One point is always at center of distribution

- Approximate as 2D view
  - Moves points with attraction/repulsion idea
    - Similar points attracted; distant points repulsed
    - Stochastic optimization using gradient descent
  - T-distribution avoids train-wreck

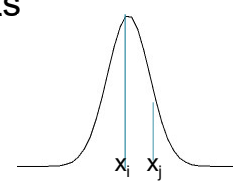


How t-SNE works

- Compute all pairwise distances using a Gaussian distribution → p
- Randomly arrange points in lower dimensional space and compute distances using a t-distribution
- Set up an optimization criterion which is KL-Divergence
- Gradient descent optimization

t-SNE

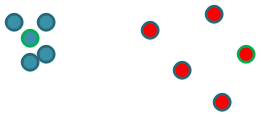
- T-Dist. Stochastic Neighbor Embedding
- Strategy:
  - Focus on local neighborhoods
  - Use normalized Gaussian probabilities of similarity between local points
  - $p_{ij} = \frac{\exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})}{\sum_k \sum_{l \neq k} \exp(-\frac{\|x_k - x_l\|^2}{2\sigma^2})}$



  - Note that the variance σ^2 is global here

t-SNE

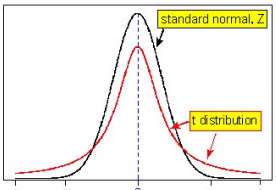
- $p_{ij} = \frac{\exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})}{\sum_k \sum_{l \neq k} \exp(-\frac{\|x_k - x_l\|^2}{2\sigma^2})}$
- Replace global variance with local variance
- $p_{j|i} = \frac{\exp(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2})}{\sum_{k \neq i} \exp(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2})}$



- $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$ kluge to restore symmetry
- $q_{ij} = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1+\|y_k - y_l\|^2)^{-1}}$ in low dim space
- Gradient descent optimize Kullback-Leibler distance KL(P|Q) = $\sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$

P & Q have different similarities?

- $p_{j|i} = \frac{\exp(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2})}{\sum_{k \neq i} \exp(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2})}$: Gaussian
- $q_{ij} = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1+\|y_k - y_l\|^2)^{-1}}$: Student-T



http://election.princeton.edu/2014/09/18/how-our-predictions-work-contin

- Heavy tails allow for distant points to be further apart as will usually happen

Optimization of t-SNE

- Kullback-Leibler distance is standard entropy measure
  - KL(P|Q) = $\sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$
- If i-j are close, p_{ij} is large or near 1.0
  - KL ensures that q_{ij} is large to avoid penalty
- If i-j are far, who cares--small anyway
- Therefore t-SNE optimizes local

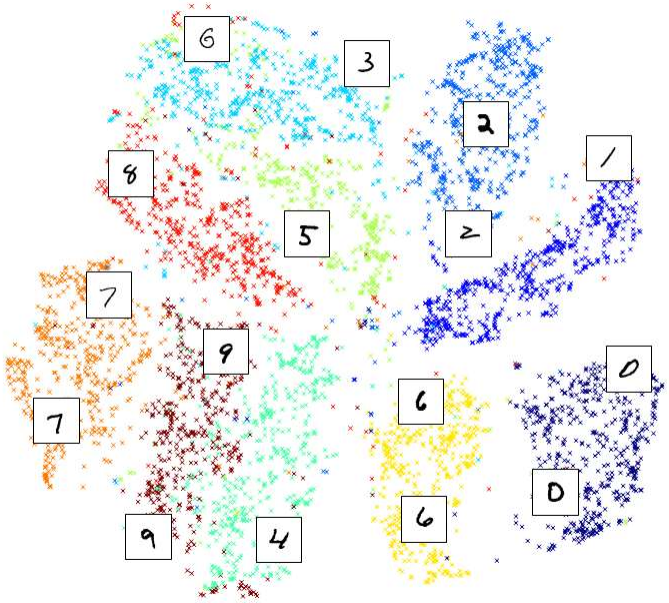
Comparison on MNIST

- http://colah.github.io/posts/2014-10-Visualizing-MNIST/
- 10K or 60K handwritten numbers

PCA MDS Sammon's

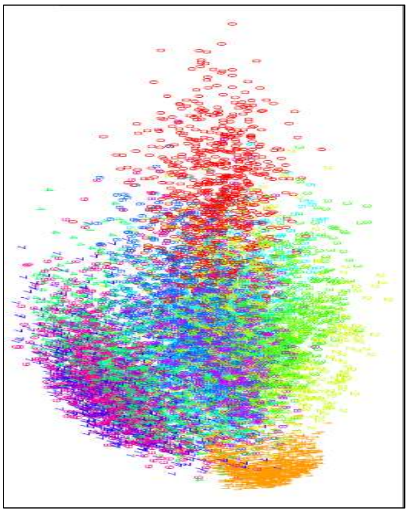
T-SNE on MNIST from

- http://alexanderfabisch.github.io/t-sne-in-scikit-learn.html

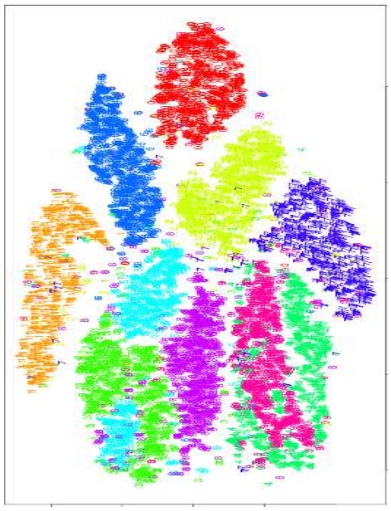


MNIST Handwritten Digits Data Set

PCA



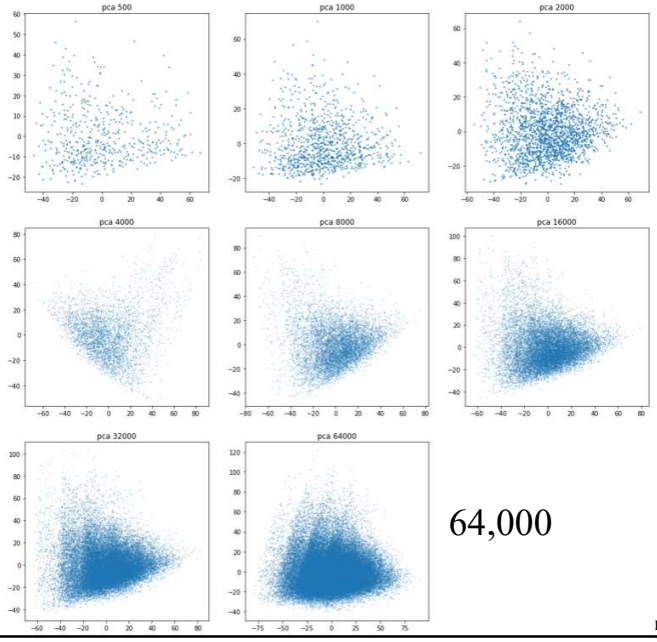
t-SNE



https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/

Ugi Data Set using PCA

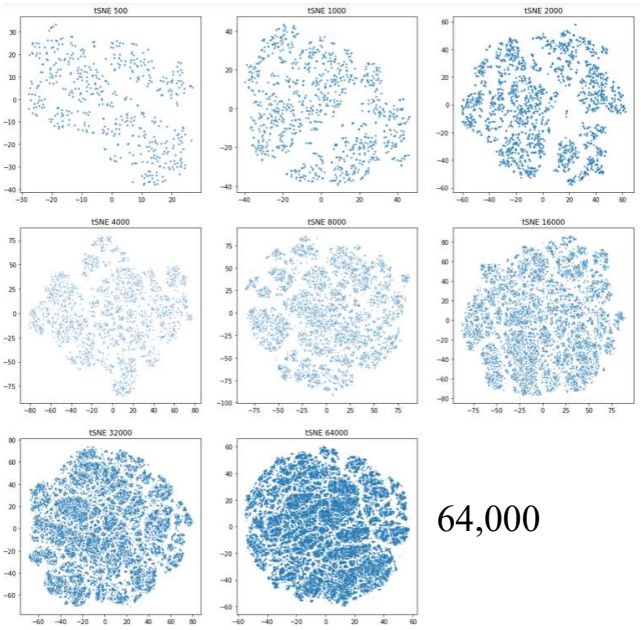
500



64,000

Ugi Data Set using t-SNE

500



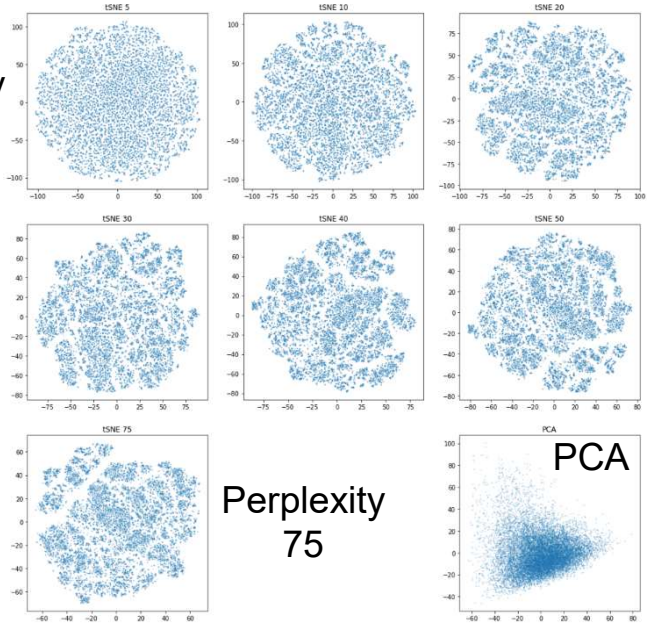
64,000

210

Peter V. Henstock

Ugi Series Data – t-SNE Perplexity

Perplexity 5

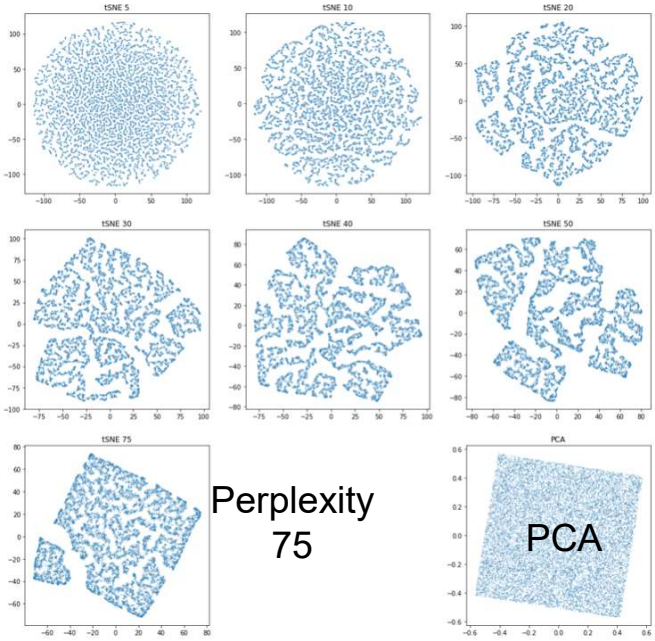


Perplexity 75

PCA

Random Data – t-SNE Perplexity

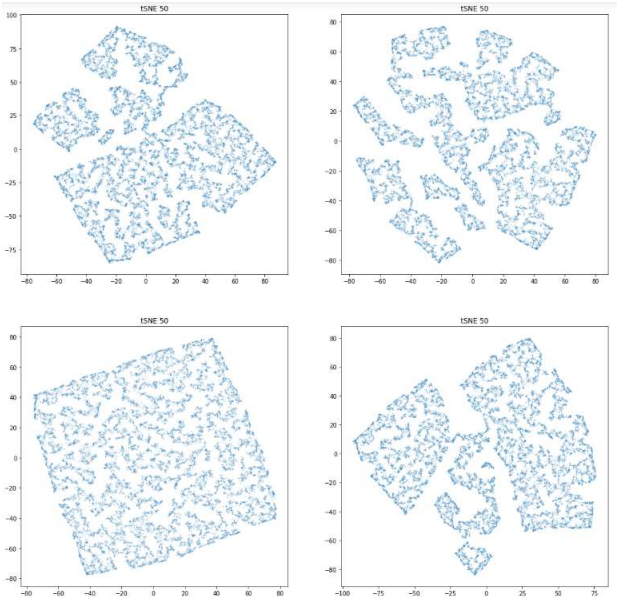
Perplexity 5



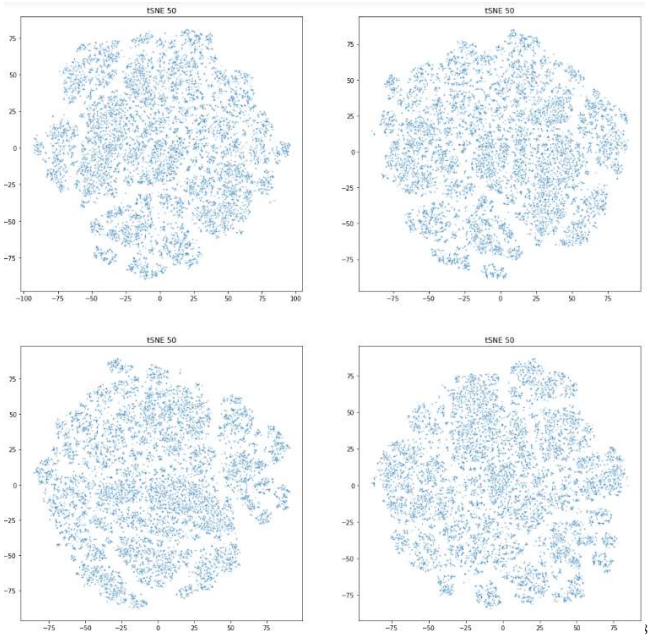
Perplexity 75

PCA

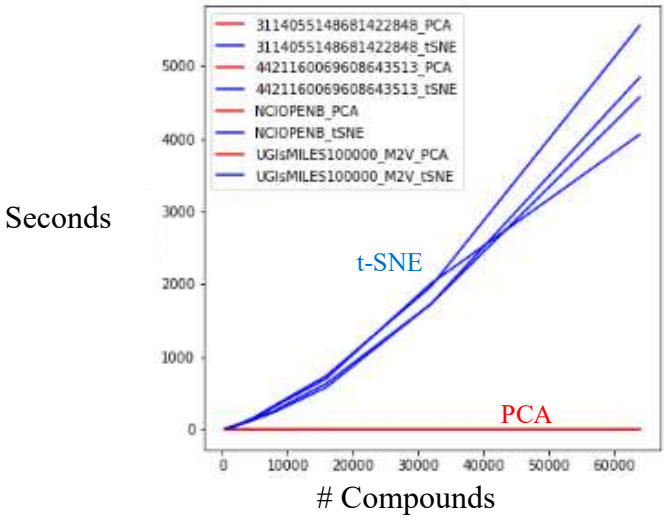
Repeatability for t-SNE (Random)



Repeatability for t-SNE (Ugi)



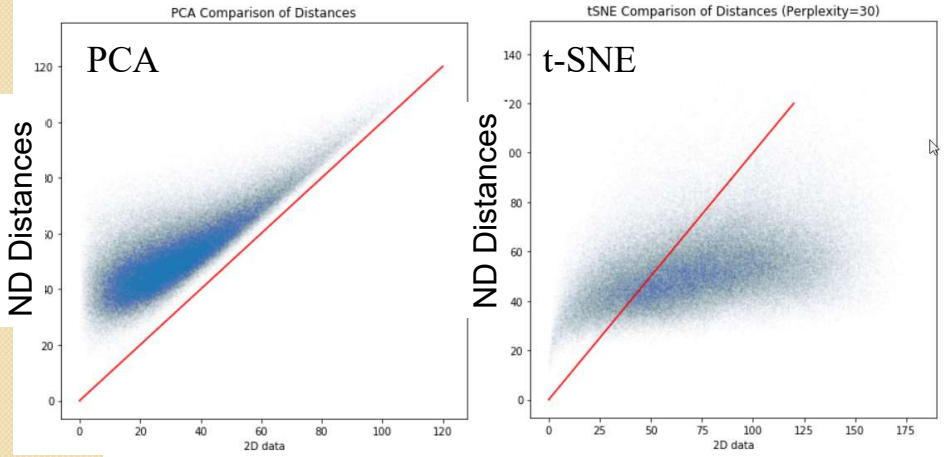
Computational Speed t-SNE vs PCA



Seconds

Compounds

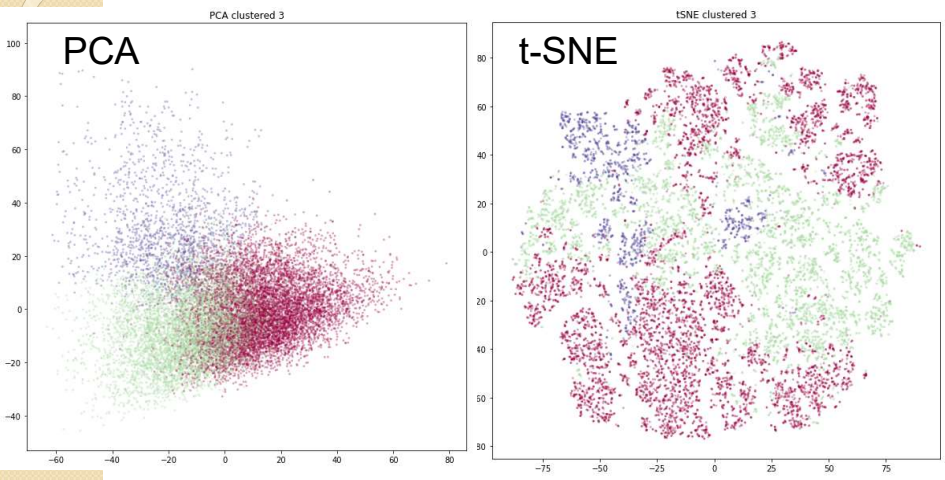
Comparing Distances: ND vs. 2D



2D Distances

2D Distances

Clustering Data



t-SNE with10 clusters