Data Ethics, Concluded

https://tinyurl.com/cis545-lecture-10-25-21

Zachary G. Ives
University of Pennsylvania
CIS 545 – Big Data Analytics

Penn Engineering

Ethical Principles around Data

Autonomy

- The right to control your data, possibly via surrogates

Informed consent

- You should explicitly approve use of your data based on understanding

Beneficence

- People using your data should do it for your benefit

Non-maleficence

- Do no harm

https://tinyurl.com/cis545-lecture-10-25-21

Facebook's Mood Manipulation Experiment

In 2012 researchers at Facebook and Cornell University manipulated the newsfeed of selected Facebook users.

- Some users were shown more positive articles.
- Others were shown more negative or sad articles.

People who were shown more positive articles, posted more positive articles themselves on Facebook

People who were shown shown more negative articles, posted more negative articles.

i.e., they demonstrated "emotional contagion"

Proc. of the National Academies of Science (2014): "Experimental Evidence of Massive-Scale Emotional Contagion Through Social Networks"

https://tinyurl.com/cis545-lecture-10-25-21

Data Science and Informed Consent

- Informed consent is often buried in the fine print
- Data is often collected first; the experiment comes later
- How the data, once collected, is going to be used is difficult to control

Most people ignore the terms of usage and just click through!

https://tinyurl.com/cis545-lecture-10-25-21

Intellectual Property

Patents only protect implementations, not ideas

Artistic expression can be *copyrighted*:

- exclusive legal right to print, publish, perform, film or record and authorize others to do the same

Derivative work can be created with permission

- There's also a notion of citation, in which we give credit to the owner
- And many open-source licenses establishing terms

What about data?

- Wikipedia, Yelp, Rotten Tomatoes, TripAdvisor
- A clinical data set, a company's data, your gene sequence, …

https://tinyurl.com/cis545-lecture-10-25-21

Admiral to price car insurance based on Facebook posts

Insurer's algorithm analyses social media usage to identify safe drivers in unprecedented use of customer data

theguardian

https://tinyurl.com/cis545-lecture-10-25-21

Privacy

https://www.panmacmillan.com/blogs/literary/george-orwell-quotes-1984-animal-farm

https://tinyurl.com/cis545-lecture-10-25-21

OKCupid Data Publicly Released

WIRED, Michael Zimmer 5/14/16

May 8, 2016: Danish researchers publicly released a dataset of ~70,000 OKcupid users

- usernames, age, gender, location, what kind of relationship they're interested in
- personality traits, answers to 1000s of profiling questions

Did they attempt to anonymize?

- Researchers' response: "… all the data found in the dataset are or were already publicly available, so releasing this dataset merely presents it in a more useful form."

https://tinyurl.com/cis545-lecture-10-25-21

Was the OKCupid Data "Public"?

- Data acquired by screen scraping – methodology not fully explained
  - Likely from an OkCupid profile researchers created!
- OkCupid users may restrict the visibility of their profiles to *logged-in users only*
  - Likely that the researchers collected—and released—profiles that were intended to *not* be publicly viewable

https://tinyurl.com/cis545-lecture-10-25-21

Privacy is not Simple

Many rules governing use of collected information

- **HIPAA:** Health Insurance Portability and Accountability Act
- **FERPA:** Family Educational Rights and Privacy Act
- **GDPR** General Data Protection Regulation (Europe)

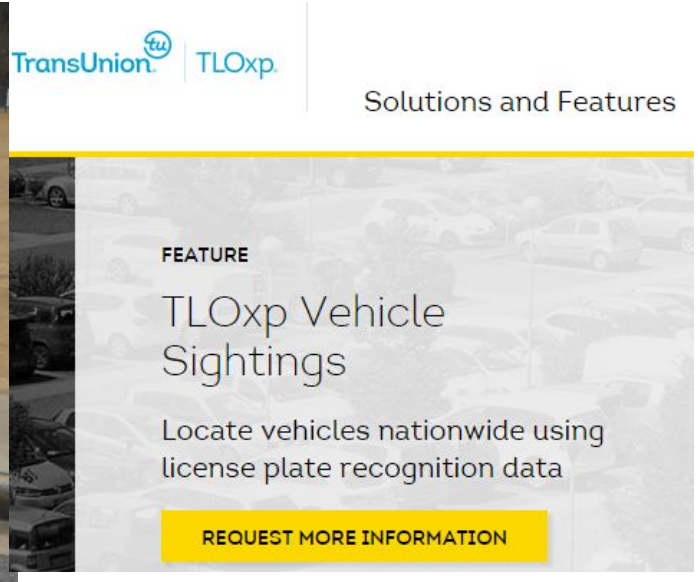
However, "information leakage" can lead to unexpected disclosures

- e.g. smart water meters

"Privacy by trust" versus "privacy by design"

https://tinyurl.com/cis545-lecture-10-25-21

Another Example: License Plate Readers



https://tinyurl.com/cis545-lecture-10-25-21

Can We Make Data Private?
Correlating "De-identified" Data

Netflix Prize Competition: released a de-identified data set with user ID, date, movie name, and the rating given by the user for that movie.

- Researchers were able to link users with IMDb's system where the users were identified, and talked about (some of) the movies they watched.

Problem: "Sparsity" of participation makes it easy to match

- In Netflix data, no two profiles are more than 50% similar.
- If a Netflix profile is more than 50% similar to a profile in IMDB, then there is a high probability that the two profiles are of the same person.

A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets …," Proc. 29th IEEE Symp. Security and Privacy, 2008.

https://tinyurl.com/cis545-lecture-10-25-21

Differential Privacy

When do you feel safe releasing personal information?

- My answers have no impact on the privatized released result?
- With high probability, an attacker looking at the privatized released result cannot learn any new information about me?
- **These are not achievable.**

Differential privacy maximizes accuracy of queries over statistical databases while minimizing the chances of identifying its records

- it adds noise and provides guarantees against a "privacy budget".
- **The privatized released result is nearly the same whether or not I submit my information.**

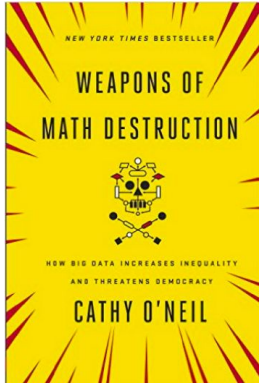
Dwork and Roth, "Algorithmic Foundations of Differential Privacy," Foundations and Trends in Theoretical Computer Science (2014).

https://tinyurl.com/cis545-lecture-10-25-21

Ethics Surrounding Algorithms (i.e., around Machine Learning)

https://tinyurl.com/cis545-lecture-10-25-21

Algorithms are not neutral



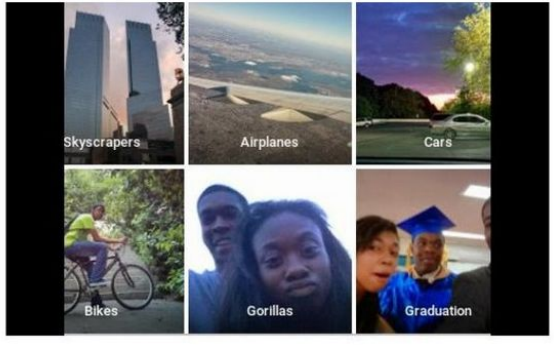
Algorithms encode our biases when:

- Training data set isn't representative
- Past population is not representative of the future population
- Overfitting to underrepresented data is common

Google apologises for Photos app's racist blunder

1 July 2015 | Technology

Share



https://tinyurl.com/cis545-lecture-10-25-21

Amazon's facial recognition matched 28 members of Congress to criminal mugshots

New ACLU test illustrates the limits of Amazon's Rekognition system

By Russell Brandom | @russellbrandom | Jul 26, 2018, 8:02am EDT

SHARE

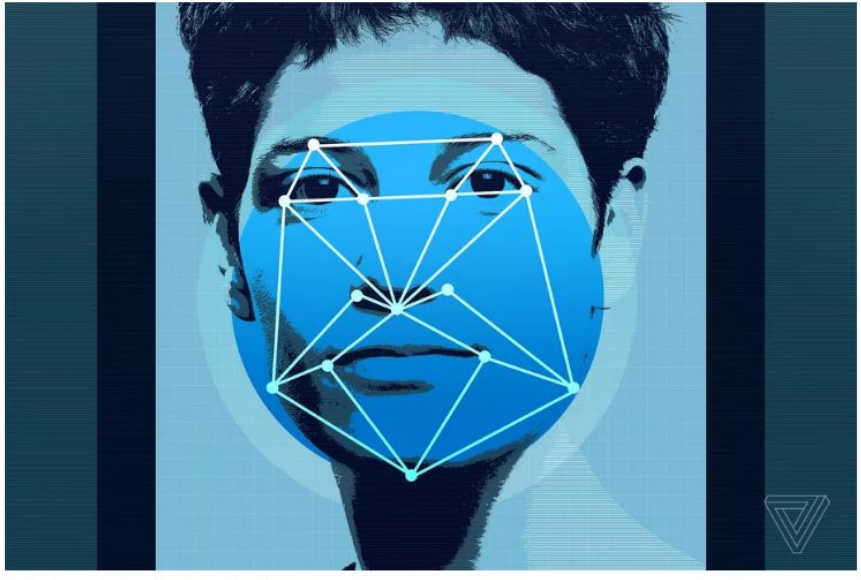


Illustration by James Bareham / The Verge

The American Civil Liberties Union tested Amazon's facial recognition system — and the results were not good. To test the system's accuracy, the ACLU scanned the faces of all 535 members of congress against 25,000 public mugshots, using Amazon's open Rekognition API. None of the members of Congress were in the mugshot lineup, but Amazon's system

The Verge,
https://www.theverge.com/2018/7/26/17615634/amazon-rekognition-aclu-mug-shot-congress-facial-recognition

https://tinyurl.com/cis545-lecture-10-25-21

Fairness

- Fairness has been studied in social choice theory, game theory, economics and law.
- Currently trendy in theoretical computer science
  - **Discrimination of an individual:** An individual from the target group gets treated differently from an otherwise identical individual not from the target group.
  - **Discrimination in aggregate outcome**: the percentage success of the target group compared to that of the general population.

Dwork, Hardt, Pitassi, Reingold and Zemel, "Fairness through Awareness" Proc. 3rd Innovations in Theoretical Computer Science, 2012.

https://tinyurl.com/cis545-lecture-10-25-21

The Inquirer DAILY NEWS philly.com

NEWS SPORTS BUSINESS HEALTH ENTERTAINMENT FOOD OPINION OBITS REAL ESTATE

Already a print subscriber? **Get Access** Never Miss a Sto

153

REPRINTS

News — Crime

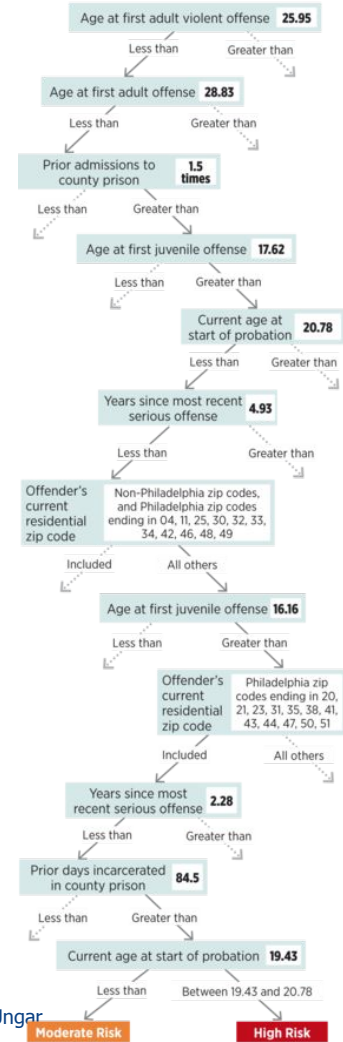
How computers are predicting crime - and potentially impacting your future

Updated: SEPTEMBER 21, 2017 — 12:53 PM EDT

https

How the Model Works

The risk-assessment tool used by the city's probation and parole department has 500 trees. This chart shows a potential path through one of them.



Brief Review

https://canvas.upenn.edu/courses/1606906/quizzes/2741522

"De-identified" data, if not protected by differential privacy:

a. Can safely be shared without revealing private information
b. Can be correlated with other data to reveal information
c. Is sparse
d. Is too large to share

Algorithms aren't neutral, in significant part because:

e. It is not profitable to be fair
f. Datasets are too large to handle
g. No trends can be spotted in a population
h. The training data may not adequately represent the population

https://tinyurl.com/cis545-lecture-10-25-21

Ethics summary

Codes of conduct for research are fairly well understood

- Get IRB approval
- obtain informed consent
- protect the privacy of subjects
- maintain the confidentiality of data collected, minimize harm

Fairness is more subtle

- What is fair treatment of a group: equal accuracy? FP rate?

Key technical aspects:

- differential privacy (bounds amount of information revealed)
- trade-off between optimizing outcomes vs avoiding discrimination against a group

https://tinyurl.com/cis545-lecture-10-25-21

Machine Learning Intro and Unsupervised Learning Overview

Zachary G. Ives
University of Pennsylvania
CIS 545 – Big Data Analytics

Penn Engineering

Data Is Hopefully *Not* Random!

There must be **correlations and structure** in our data values, and between them and **classes or outcomes** – otherwise we just have random phenomena!

https://commons.wikimedia.org/wiki/File:Slot_machine.jpg
CC-SA 2.0

For the next part of this class: we need to find that structure, aka the underlying **model**!

https://tinyurl.com/cis545-lecture-10-25-21

Machine Learning Basics

Data is typically comprised of **features** – values that might be useful in predicting the output

- Red, green, blue values of pixels in an image
- Keywords in a document
- Purchases of a customer

These are used in machine learning, and as we saw they are in a matrix!

"Not all features are created equal"

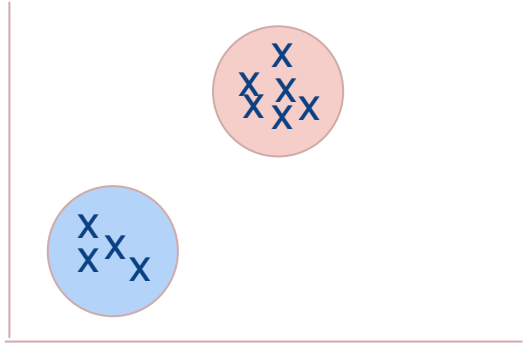
- Some are easier to learn from
- Some are correlated with others!

https://tinyurl.com/cis545-lecture-10-25-21

Two Flavors of Machine Learning

1. Find the *structure* within the data (*unsupervised)*

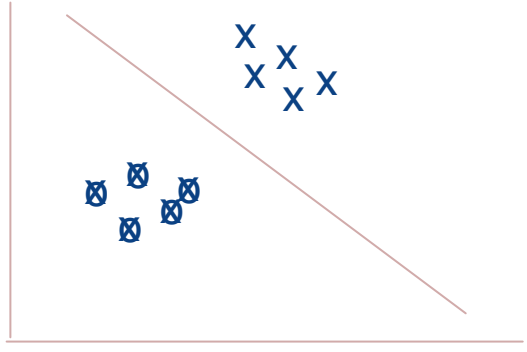
   Input data: $x_1, x_2, x_3, \ldots, x_n$



2. Find a *function* mapping from data features to classes (*supervised)*

   Input data: $x_1, x_2, x_3, \ldots, x_n$

   + Labels: $y_1, y_2, y_3, \ldots, y_n$



https://tinyurl.com/cis545-lecture-10-25-21

A Workflow for ML: Unsupervised □ Supervised Learning

Sometimes data has MANY fields/features, each of which *might* be a good feature

- But which features actually matter the most?
- Are they in the right representation?

Often, we **start with unsupervised learning**, which can:

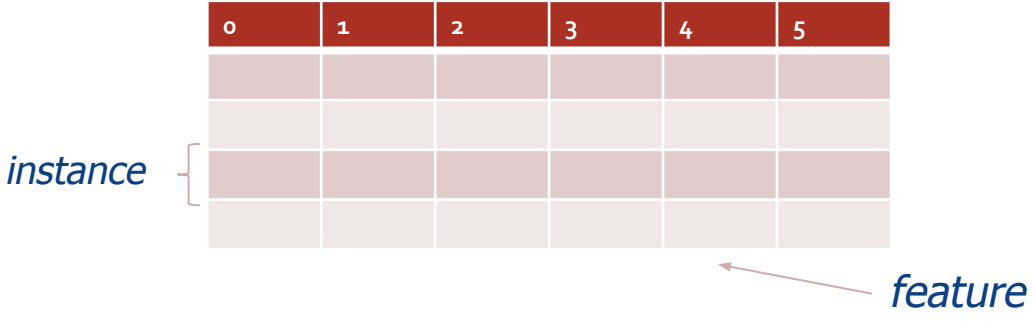
- Reduce number of dimensions, generate more "natural" features
- Provide insights into "natural clusters"

Then we'll run **supervised** learning methods (see later modules)!

https://tinyurl.com/cis545-lecture-10-25-21

Recall the Inputs to ML Problems

Data will generally be in matrices – either integer or floating-point



Each row is typically a single observation

If we have non-numeric data we may need to **transform** or **one-hot encode** it

https://tinyurl.com/cis545-lecture-10-25-21

Sparsity

Sometimes (especially with one-hot encoding) we cause an explosion of columns for each item!

Alabama	Alaska	…	Wyoming
0	0	…	1
1	0	…	0

For efficiency we may want to use a **sparse matrix** data structure to hold the matrix…

https://tinyurl.com/cis545-lecture-10-25-21

Conceptual Storage of a Normal Matrix vs a Sparse Matrix

By default, we essentially have a list of lists of ints – one int per cell

```
[
  [ 0, 1, 0, 0, 0, …],
  [ 1, 0, 0, 0, 0, …],
…]
```

numpy array() or ndarray()

A sparse matrix stores a map from cell coordinates to nonzero items (this is one of several forms)

```
{(0, 1) □ 1,
 (1, 0) -> 1,
…}
```

scipy.sparse.csr_matrix*(array)*

https://tinyurl.com/cis545-lecture-10-25-21

Brief Review

https://canvas.upenn.edu/courses/1606906/quizzes/2688466

Unsupervised machine learning

a. Predicts class membership
b. Requires TensorFlow
c. Finds structure in the values of the features
d. Develops machine learning models without needing human input

To store values, a sparse matrix uses

e. a list of arrays
f. an array of arrays
g. an array of lists
h. a dictionary

https://tinyurl.com/cis545-lecture-10-25-21

This Module: Finding Structure in Our Matrix

We'll start our discussion off by looking at how to reduce the number of features – dimensionality reduction

- Principal Components Analysis (PCA)
- Scaling PCA to big data
- PCA alternatives

https://tinyurl.com/cis545-lecture-10-25-21

Principal Components Analysis

https://tinyurl.com/cis545-011

Zachary G. Ives

University of Pennsylvania

CIS 545 – Big Data Analytics

Penn Engineering

Unsupervised Machine Learning

Technique 1: **Dimensionality Reduction**

Principal Component Analysis (PCA)

- Reduce high number of correlated **features** to a lower number of uncorrelated features
- New features are weighted combinations of originals

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Technique 2: Clustering

Find groupings in the data

https://tinyurl.com/cis545-lecture-10-25-21

What Is Principal Component Analysis?

An unsupervised method that takes X from *p* dimensions down to *k* dimensions

- Each dimension is orthogonal to the rest

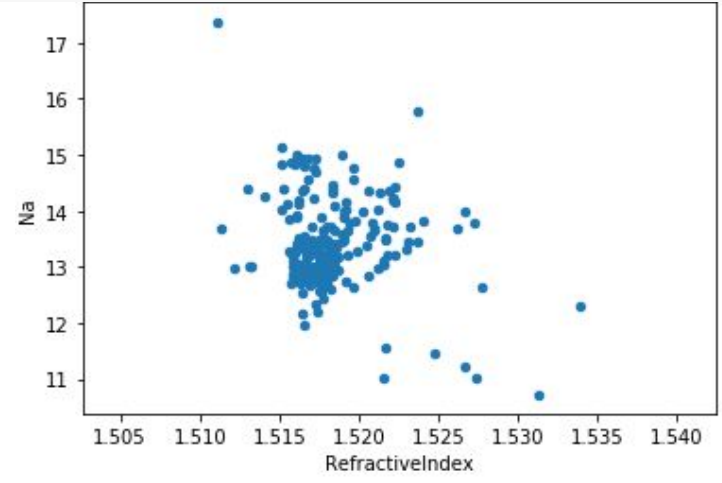
Why is this helpful?

- Reduces noise in the data for supervised learning
- Reduces data
- Simpler to visualize data (though dimensions may be unintuitive!)

https://tinyurl.com/cis545-lecture-10-25-21

The Intuition

ID	RefractiveIndex	Na	Mg	Al	Si	K	Ca
1	1.52101	13.64	4.49	1.10	71.78	0.06	8.75
2	1.51761	13.89	3.60	1.36	72.73	0.48	7.83
3	1.51618	13.53	3.55	1.54	72.99	0.39	7.78



Take a data set as a matrix **X** of *features*

- Some have little variance, some have a lot
- Some are correlated, some are not

Scale each dimension so they are comparable

Consider the *glass* dataset, with 2D to the right

- **Sodium** and **refractive index** might in fact be correlated!

- Find the vector that **maximizes variance** – the **principal eigenvector**
- Repeat for the next orthogonal vector, etc.

https://tinyurl.com/cis545-lecture-10-25-21

How Do We Formalize This?

https://tinyurl.com/cis545-lecture-10-25-21

The Covariance Matrix

$$\begin{bmatrix} E[(X_1 - E[X_1])(X_1 - E[X_1]) & \cdots & E[(X_1 - E[X_1])(X_p - E[p]) \\ \vdots & \ddots & \vdots \\ E[(X_p - E[X_p])(X_1 - E[X_1]) & \cdots & E[(X_p - E[X_p])(X_p - E[X_p]) \end{bmatrix}$$

For 1 <= *a* <= *p*

For 1 <= *b* <= *p*

Let $\Sigma[b,a] = \mathrm{cov}(\mathrm{X}^{(a)}, X^{(b)})$

- $$\Sigma = E\left[\left(X - \bar{X}\right)^T \left(X - \bar{X}\right)\right]$$

And if X is zero-centered (and *n* is large):

$$\Sigma = \frac{X^T X}{n-1} \approx \frac{X^T X}{n}$$

https://tinyurl.com/cis545-lecture-10-25-21

From Covariance Matrix, Computing the Principal Components

-

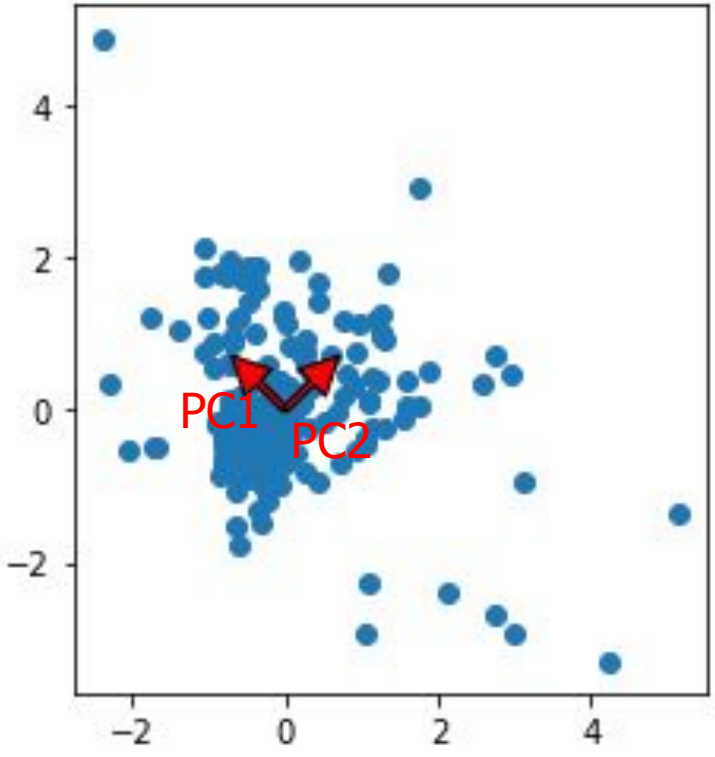
https://tinyurl.com/cis545-lecture-10-25-21

PCA, Visualized

Transform to new coordinate system:

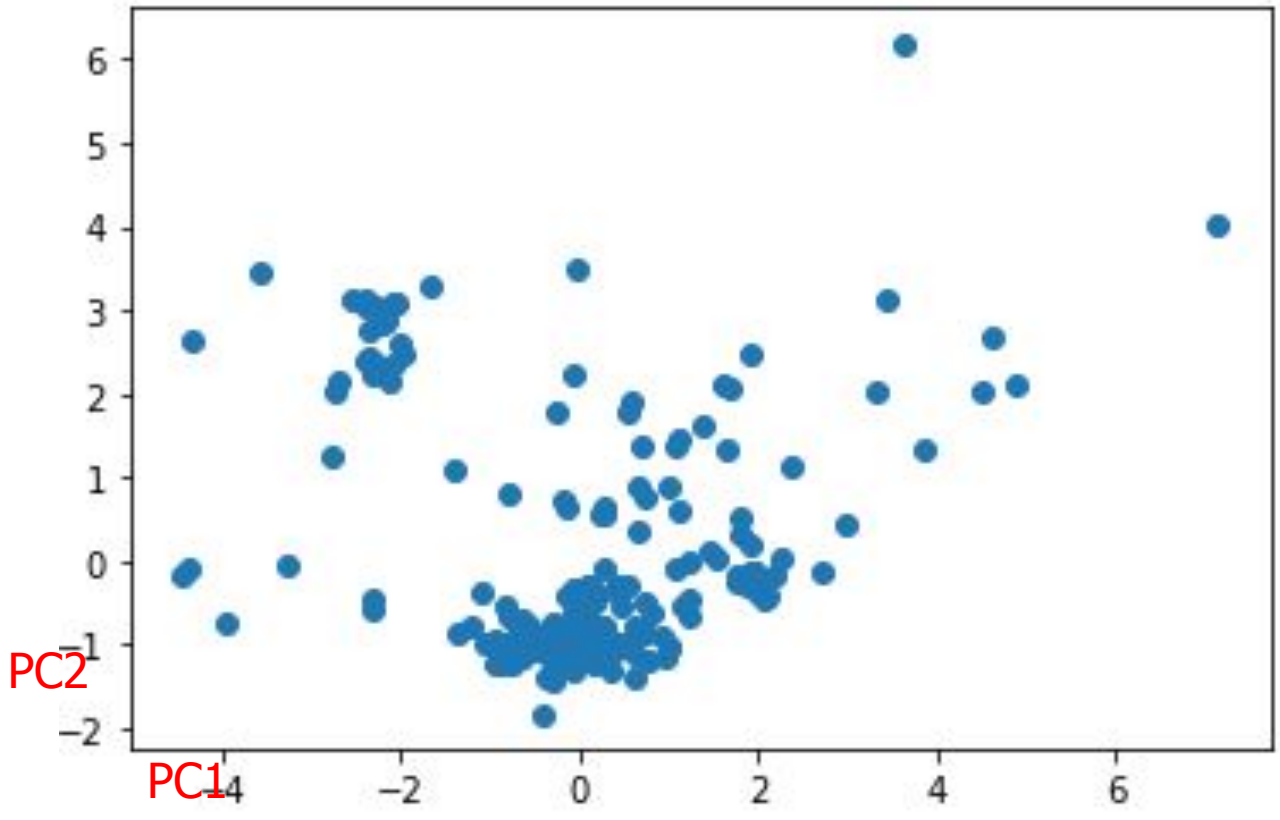
1. Find directions of maximum variation (covariance)

2. Minimize reconstruction error



https://tinyurl.com/cis545-lecture-10-25-21

PCA Projection



https://tinyurl.com/cis545-lecture-10-25-21

Observations

- PCA is *not* scale-invariant as it finds the direction of maximum variation!

- We can look at *explained variance* for each component – the ratio of the principal component vs the total variance across all components
  - First *k* principal components explain the **most variance any *k* variables can explain**

https://tinyurl.com/cis545-lecture-10-25-21

Brief Review

https://canvas.upenn.edu/courses/1606906/quizzes/2688519

The first component in PCA is based on the vector that

a. minimizes variance
b. minimizes entropy
c. maximizes variance
d. is orthogonal to the original features

(If we are not using SVD), PCA uses the eigenvectors and eigenvalues of

e. the weight transfer matrix
f. the weight matrix
g. the covariance matrix
h. the variance matrix

https://tinyurl.com/cis545-lecture-10-25-21

PCA Recap

- Finds the directions of maximum variation

  by computing the principal eigenvectors of the covariance matrix

- Returns a projection matrix and a feature subspace that minimizes reconstruction error!

- Next: how do we use this?

https://tinyurl.com/cis545-lecture-10-25-21

Formalizing PCA

For *i*th instance $\boldsymbol{x_i}$, find *k* scores t_{ij} by projecting $\boldsymbol{x_i}$ onto each of *k* **loading vectors** w_j

$$t_{ij} = \boldsymbol{x}_i \cdot \boldsymbol{w}_j$$

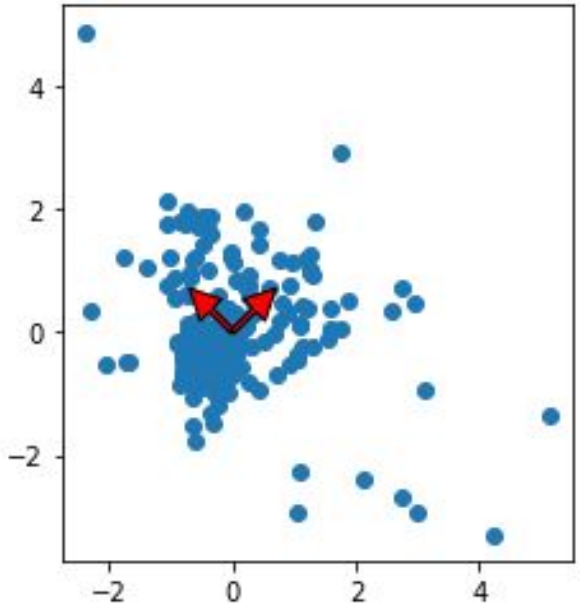
Given data matrix $\boldsymbol{X}$ with *n* rows and *p* columns, we define the T matrix:

$$\boldsymbol{T}_{\boldsymbol{n},\boldsymbol{k}} = \boldsymbol{X}_{n,p}\boldsymbol{W}_{p,k}$$

We want to invert this:

$$\boldsymbol{T}_{n,k}\boldsymbol{w}_{k,p}^{-1} = \boldsymbol{X}_{n,p}$$

We can't do this exactly but can approximate it!



https://tinyurl.com/cis545-lecture-10-25-21

Applying PCA

Zachary G. Ives
University of Pennsylvania
CIS 545 – Big Data Analytics

Penn Engineering

A PCA Example

Glass Identification Data Set from UC Irvine ML Repository

https://archive.ics.uci.edu/ml/datasets/glass+identification

The task:

Forensic scientists need to identify the type of glass based on its chemical composition!

Can we take a dataset of glass instances and their chemical compositions and:

1. Reduce dimensionality (by default 8 elements + refractive index are given)
2. Predict the type of glass: building, vehicle, container, tableware, headlight [this part uses **supervised machine learning**]

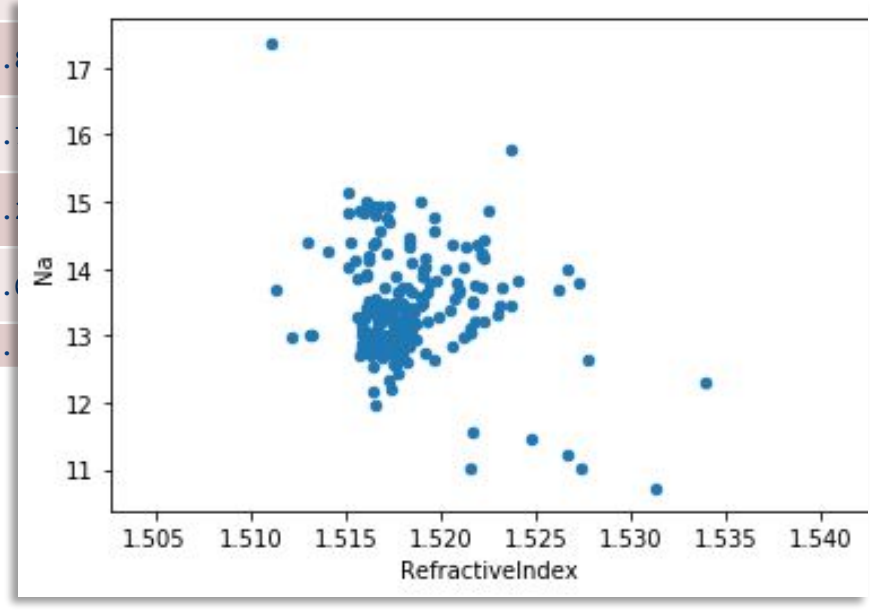
https://tinyurl.com/cis545-lecture-10-25-21

Glass Data in a DataFrame

	ID	Refractive Index	Na	Mg	Al	Si	K	Ca	Ba	Fe	Type
0	1	1.52101	13.64	4.49	1.10	71.78	0.06	8.75	0.00	0.0	1
1	2	1.51761	13.89	3.60	1.36	72.73	0.48	7.[illegible]			
2	3	1.51618	13.53	3.55	1.54	72.99	0.39	7.[illegible]			
3	4	1.51766	13.21	3.69	1.29	72.61	0.57	8.[illegible]			
4	5	1.51742	13.27	3.62	1.24	73.08	0.55	8.[illegible]			

```
# Remove the ID and the Type labels for
# the training data set X
X = glass_df.drop([ 'ID','Type'], axis=1)

# Labels are in a separate y vector
y = glass_df[ 'Type']
```



https://tinyurl.com/cis545-lecture-10-25-21

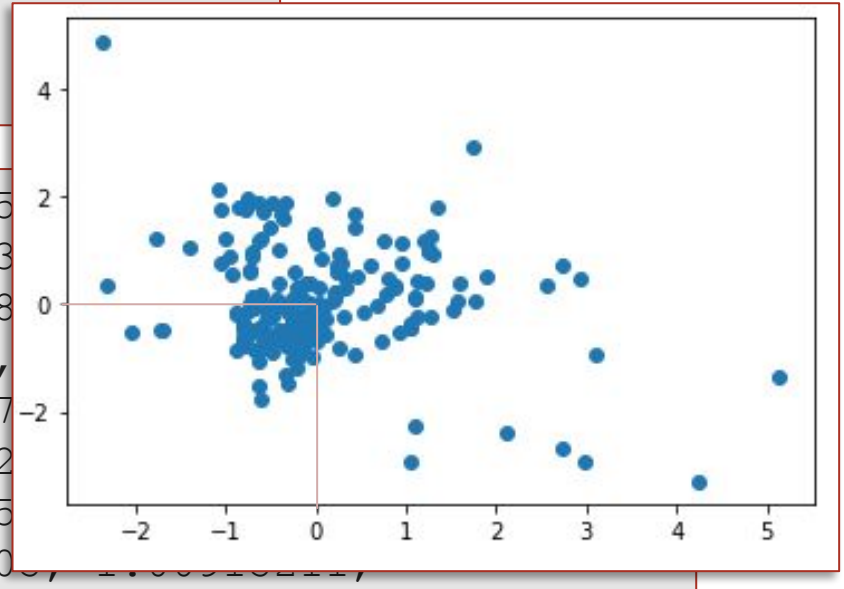
Best Practice: Scaling and Centering

```
# Standardize features by removing the mean and scaling to unit variance
from sklearn.preprocessing import StandardScaler

# Standardizing the features between 0 and 1
X = StandardScaler().fit_transform(X)

X
```

```
array([[ 0.87286765, 0.28495326, 1.25
-0.35287683, -0.5864509 ], [-0.249333
0.63616803, ..., -0.79373376, -0.3528
[-0.72131806, 0.14993314, 0.60142249,
-0.35287683, -0.5864509 ], ..., [ 0.7
-1.86551055, ..., -0.36410319, 2.9532
[-0.61239854, 1.19327046, -1.86551055
2.81208731, -0.5864509 ], [-0.4143630
-1.86551055, ..., -0.23732695, 3.01367739, -0.5864509 ]])
```

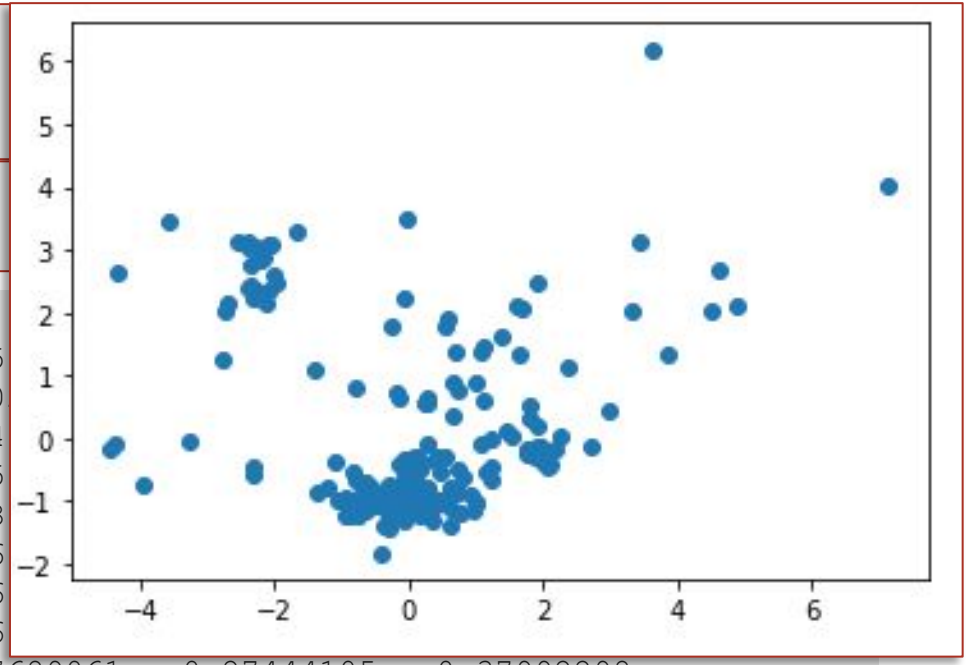


https://tinyurl.com/cis545-lecture-10-25-21

PCA in SciKit-Learn

```
from sklearn.decomposition import PCA
pca = PCA(n_components=9)
X2 = pca.fit_transform(X)
```

```
# Let's see the components
pca.components_
```




```
array([[ 0.54517662, -0.2581256 , 0.11088095,
0.49230609, -0.25037512, 0.18584154], [ 0.285
-0.15509891, -0.15397013, 0.3453798 , 0.48470
0.00841796, 0.32923712, -0.45870884, 0.662574
0.14738099, 0.49124204, 0.37878577, -0.137505
0.13317545, -0.23049202], [-0.0735427 , 0.153
-0.30703984, -0.18818774, 0.25133426, 0.87326
-0.01885731, 0.08609797, -0.24363237, -0.1486
0.14858006, -0.20604537, -0.69923557, 0.21606
0.07372136], [-0.7522159 , -0.12769315, -0.07689061, -0.27444105, -0.37992298,
-0.10981168, 0.39870468, 0.14493235, -0.01627141], [ 0.02573194, -0.31193718, -0.57727335,
-0.19222686, -0.29807321, -0.26050863, -0.57932321, -0.1982282 , -0.01466944]])
```

https://tinyurl.com/cis545-lecture-10-25-21

Computing PCA via Singular Value Decomposition

A great tutorial: https://arxiv.org/pdf/1404.1100.pdf

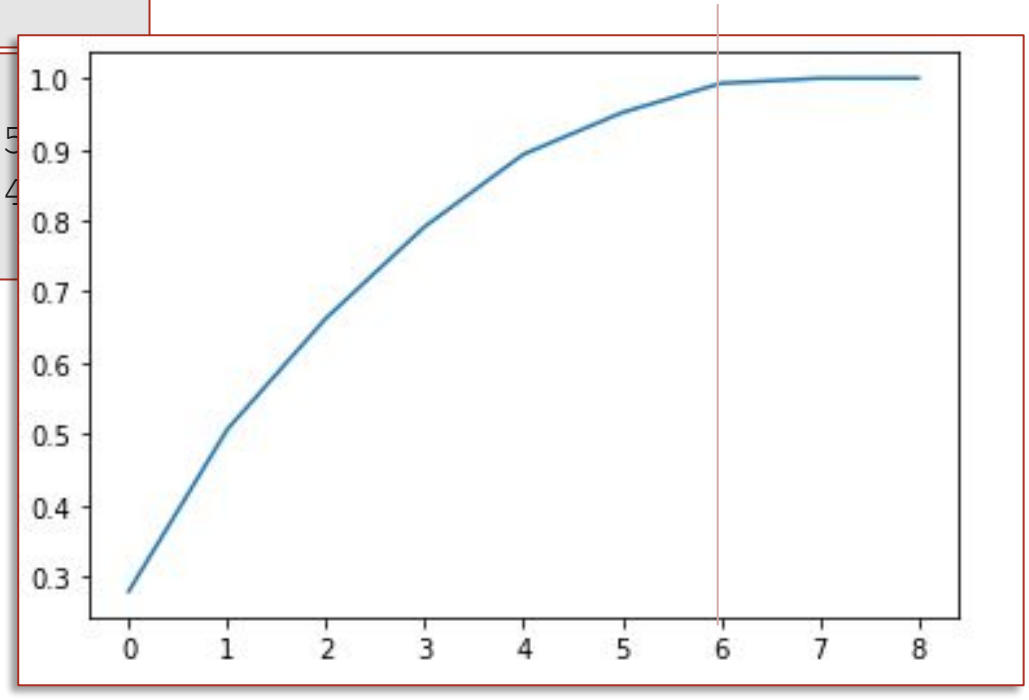
- Subtract off the mean from each column
- Calculate the SVD of the matrix
- Take the *k* principal eigenvectors for the matrix **W**

https://tinyurl.com/cis545-lecture-10-25-21

Principal Components vs Variance (Showing with 9 possible components)

```
np.set_printoptions(suppress=True)
pca.explained_variance_ratio_
```

```
array([0.27901819, 0.2277858 ,
0.15609378, 0.12865138, 0.101555
0.05862613, 0.04099538, 0.007094
0.00017876])
```



We should ideally set *k* to the smallest value where we see things flatten out

https://tinyurl.com/cis545-lecture-10-25-21

PCA Often Feeds into Supervised Machine Learning

```
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split

# Split 80% of data to train the supervised classifer
# and 20% to test on
X_train, X_test, y_train, y_test = train_test_split(\
  X, y, test_size=0.20, random_state=42)

# Fit the PCA on the training data
pca = PCA(n_components=6)
X_train_2 = pca.fit_transform(X_train)

# Train a simple linearclassifier (details
# aren't important yet) -- tries to find the best
# weighted linear combination to match the output)
regr = linear_model.LinearRegression()
regr.fit(X_train_2, y_train)
```

```
X_test_2 = pca.transform(X_tes
t)

regr.predict(X_test_2)

regr.score(X_test_2, y_test)
```

```
0.8739872270917841
```

https://tinyurl.com/cis545-lecture-10-25-21

Brief Review

https://canvas.upenn.edu/courses/1606906/quizzes/2688428

Before applying the SciKit-Learn PCA algorithm, it is a good idea to run `fit_transform` from the

a. `StandardScaler`
b. `LinearRegression` module
c. machine learning classifier
d. `Pipeline`

To pick the number of dimensions, we look at the

e. explained variance ratio
f. KL divergence
g. average value
h. dimensionality reduction constant

https://tinyurl.com/cis545-lecture-10-25-21

PCA So Far

We've seen that:

- We can easily do it via SciKit-Learn over local matrices
- We should scale and center
- We can pick the number of components via the explained variance ratio

What about if we have truly big data?

https://tinyurl.com/cis545-lecture-10-25-21

Scaling PCA to Big Data

Zachary G. Ives
University of Pennsylvania
CIS 545 – Big Data Analytics

Penn Engineering

Loading Big Data into Spark

```
# ID,RefractiveIndex,Na,Mg,Al,Si,K,Ca,Ba,Fe,Type
schema = StructType([
        StructField("ID", IntegerType(), True),
        StructField("RefractiveIndex", DoubleType(), True),
…

glas
'htt
s.da
```

```
+---+------------------+-----+----+----+-----+----+----+---+----+----+
| ID|   RefractiveIndex|   Na|  Mg|  Al|   Si|   K|  Ca| Ba|  Fe|Type|
+---+------------------+-----+----+----+-----+----+----+---+----+----+
|  2|1.5176100000000001|13.89| 3.6|1.36|72.73|0.48|7.83|0.0| 0.0|   1|
|  3|1.5161799999999999|13.53|3.55|1.54|72.99|0.39|7.78|0.0| 0.0|   1|
|  4|           1.51766|13.21|3.69|1.29|72.61|0.57|8.22|0.0| 0.0|   1|
|  5|           1.51742|13.27|3.62|1.24|73.08|0.55|8.07|0.0| 0.0|   1|
|  6|           1.51596|12.79|3.61|1.62|72.97|0.64|8.07|0.0|0.26|   1|
+---+------------------+-----+----+----+-----+----+----+---+----+----+
only showing top 5 rows
```

https://tinyurl.com/cis545-lecture-10-25-21

PCA for Big Data: Converting to a Matrix & PCA

```
from pyspark.mllib.linalg import Vectors
from pyspark.mllib.linalg.distributed import RowMatrix

# Each SDF row gets processed as a
# and converted into a Vector
M = RowMatrix(glass_sdf.select('Ref
  'Mg','Al','Si','K','Ca','Ba','Fe'
  lambda row: Vectors.dense(list(rc
)

pc = M.computePrincipalComponents(6)

projected = M.multiply(pc)

projected.rows.collect()
```

```
[DenseVector([-2.2414, -13.0845,
41.5609, -45.2941, 13.1599, -2.874]),
DenseVector([-2.2496, -13.0844,
41.9784, -45.3113, 12.9424, -2.735]),
DenseVector([-2.4224, -12.4907,
41.8856, -44.9697, 12.9633, -2.9404]),
…
```

https://tinyurl.com/cis545-lecture-10-25-21

Brief Review

https://canvas.upenn.edu/courses/1606906/quizzes/2688506

For machine learning matrices in Spark, do we need to consider which rows are on which machine?

a. it depends on the classifier
b. it depends on the values
c. yes
d. no

https://tinyurl.com/cis545-lecture-10-25-21

Summary: Principal Component Analysis

- Unsupervised method to reduce dimensionality, often before running supervised machine learning
  - Finds directions of maximum variance in high-dimensional data
  - Projects onto a smaller (or equal) subspace
  - Uses SVD (or eigenvectors) under the covers

- Assumes linearity
- Sensitive to data scaling

https://tinyurl.com/cis545-lecture-10-25-21

Dimensionality Reduction for Visualization

Zachary G. Ives
University of Pennsylvania
CIS 545 – Big Data Analytics

Penn Engineering

A Common Alternative for Visualization: t-SNE
t-Distributed Stochastic Neighbor Embedding

For visualizing high-dimensional data

Makes similar data points close, maintains "neighbor" relationships

- Converts similarities between data points to joint probabilities
- Minimizes **Kullback-Leibler divergence** between joint probabilities of low-dimensional embedding and high-dimensional data

Non-convex cost function

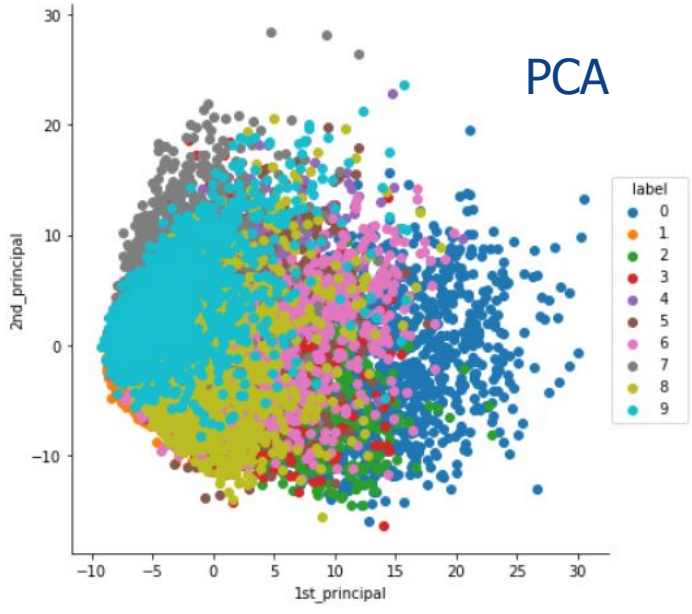
- Different initializations yield different results
- Can be slow, may want to do PCA first!

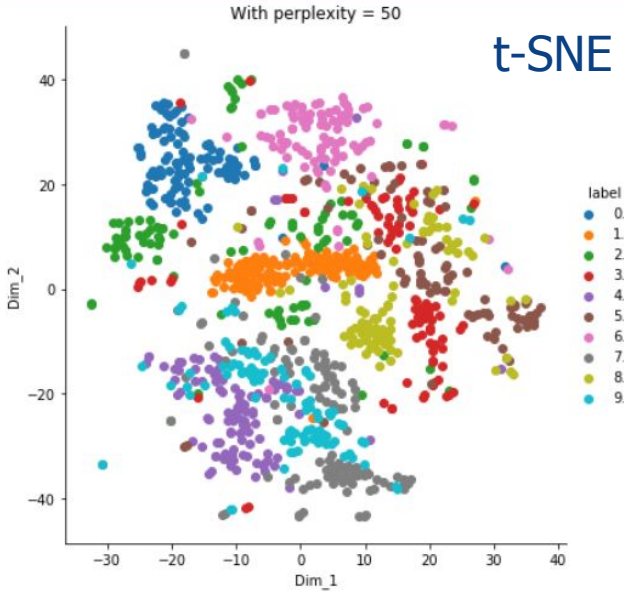
https://tinyurl.com/cis545-lecture-10-25-21

t-SNE vs PCA

https://medium.com/analytics-vidhya/pca-vs-t-sne-17bcd882bf3d

For numeric digit detection (MNIST), visualizations of 1st 2 components



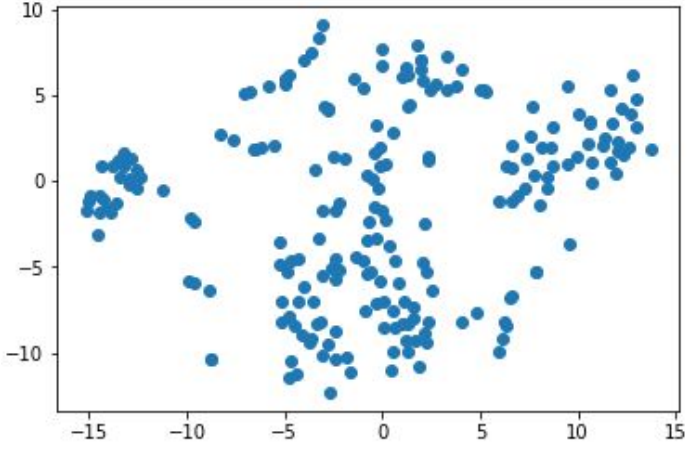


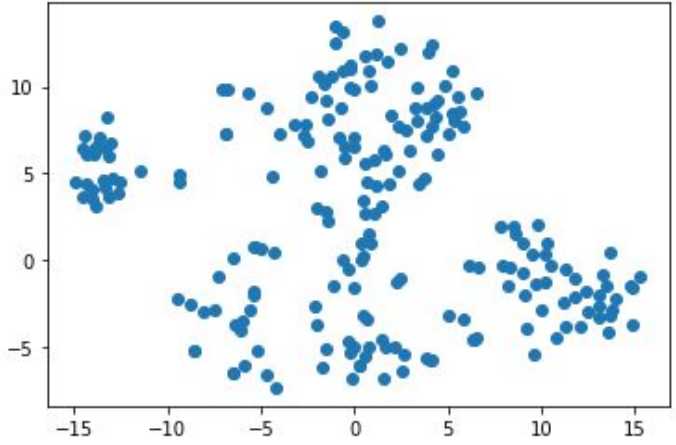
https://tinyurl.com/cis545-lecture-10-25-21

t-SNE in Python

```
from sklearn.manifold import TSNE

X_embedded = TSNE(n_components=2).fit_transform(X)
plt.scatter(X_embedded[:,0],X_embedded[:,1])
```





2nd run is different!

https://tinyurl.com/cis545-lecture-10-25-21

Brief Review

https://canvas.upenn.edu/courses/1606906/quizzes/2688448

Which algorithm focuses on preserving neighbor relationships?

a. t-SNE
b. SVD
c. PIK
d. PCA

https://tinyurl.com/cis545-lecture-10-25-21

Dimensionality Reduction Wrap-up

Two main methods for reducing dimensionality

- PCA – assumes linearity, sensitive to scaling, but broadly effective
- t-SNE – helpful in visualizing highly dimensional data, doesn't scale as well

Generally these are used as an intermediate step towards some broader task, whether visualizing + understanding data or doing supervised machine learning

https://tinyurl.com/cis545-lecture-10-25-21