10-605/805 – ML for Large Datasets Lecture 3: Principal Component Analysis

Henry Chai

9/6/22

Front Matter

- HW1 released 8/30, due 9/14 at 11:59 PM
  - **For HW1 only, the programming part is optional (but strongly encouraged)**
  - The written part is nominally about PCA but can be solved using pre-requisite knowledge (linear algebra)
- Recitations on Friday, 11:50 – 1:10 (**different from lecture**) in GHC 4401 (**same as lecture**)
  - Recitation 2 on 9/9: Review of linear algebra

Data Pre-processing

- ETL (extract-transfer-load)
- Cleaning data
  - Missing features/labels
  - Duplicated observations
  - Formatting errors
- Understanding data
  - Summarization
  - Exploration
  - Visualization

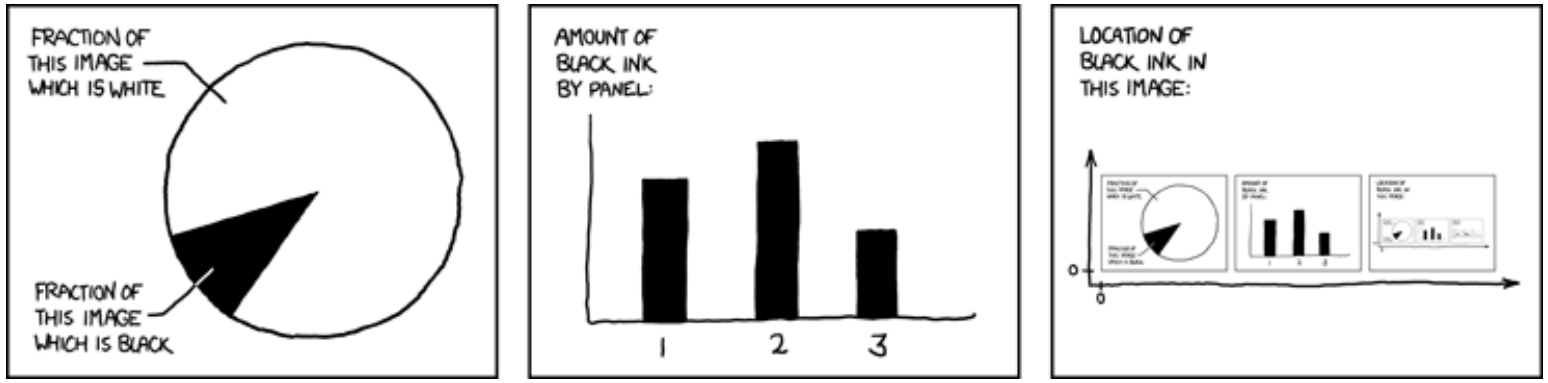
Data Pre-processing

- ETL (extract-transfer-load)
- Cleaning data
  - Missing features/labels
  - Duplicated observations
  - Formatting errors
- **Understanding data**
  - **Summarization**
  - **Exploration**
  - **Visualization**

Given some (labelled) dataset, what questions can you ask to better understand the data?

- How was the data generated?
- How much data is there?
- What do you want to do with the data?
- What values do the data take?
- What patterns exist in the data?
- How do the features relate to the label?

Data Visualization

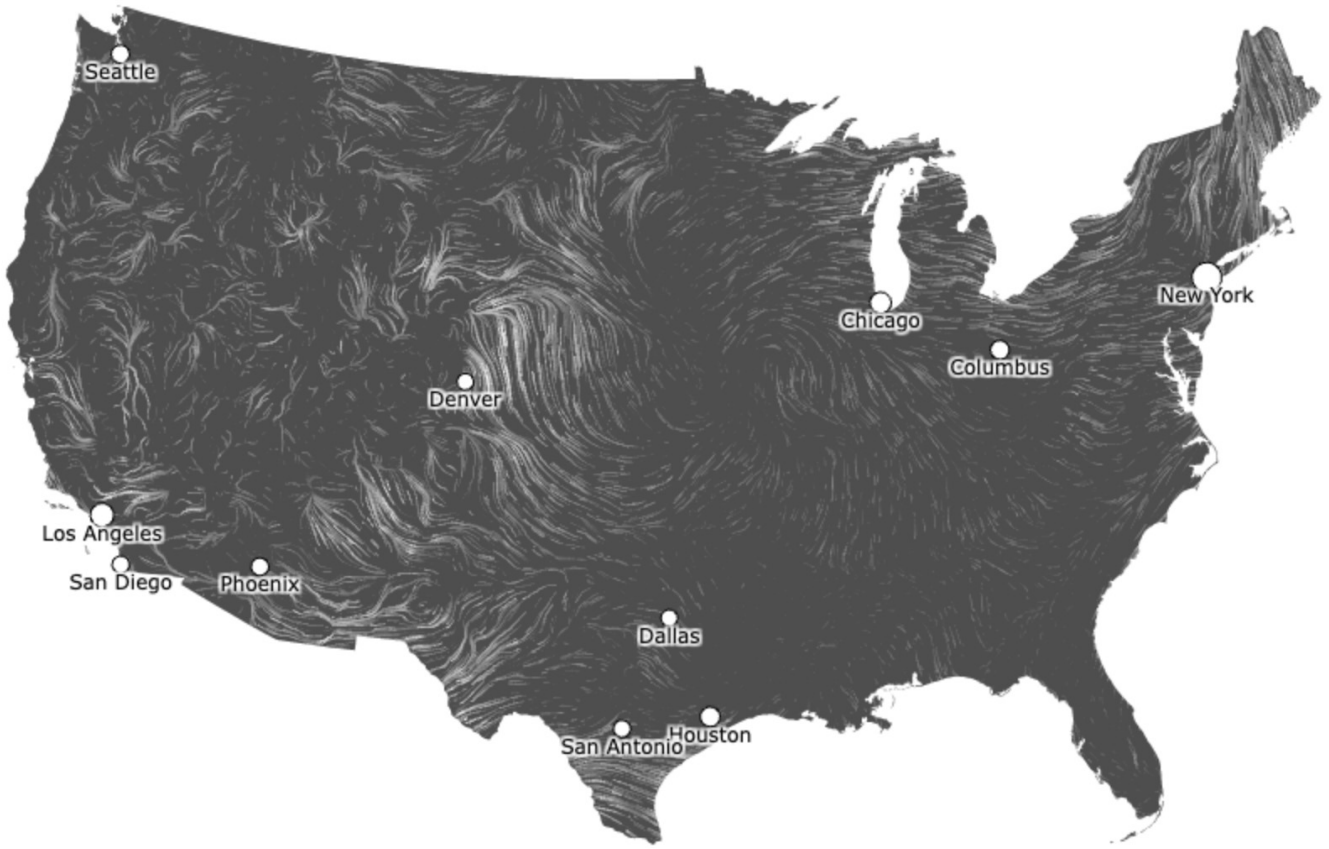


- Visualizations can be used to
  - Provide insight about trends/groups/relationships
  - Reveal systematic errors
  - Aid in model selection
  - Evaluate training (e.g., measure convergence)
  - Interpret/explain predictions

Source: https://xkcd.com/688/

Data Visualization: Examples

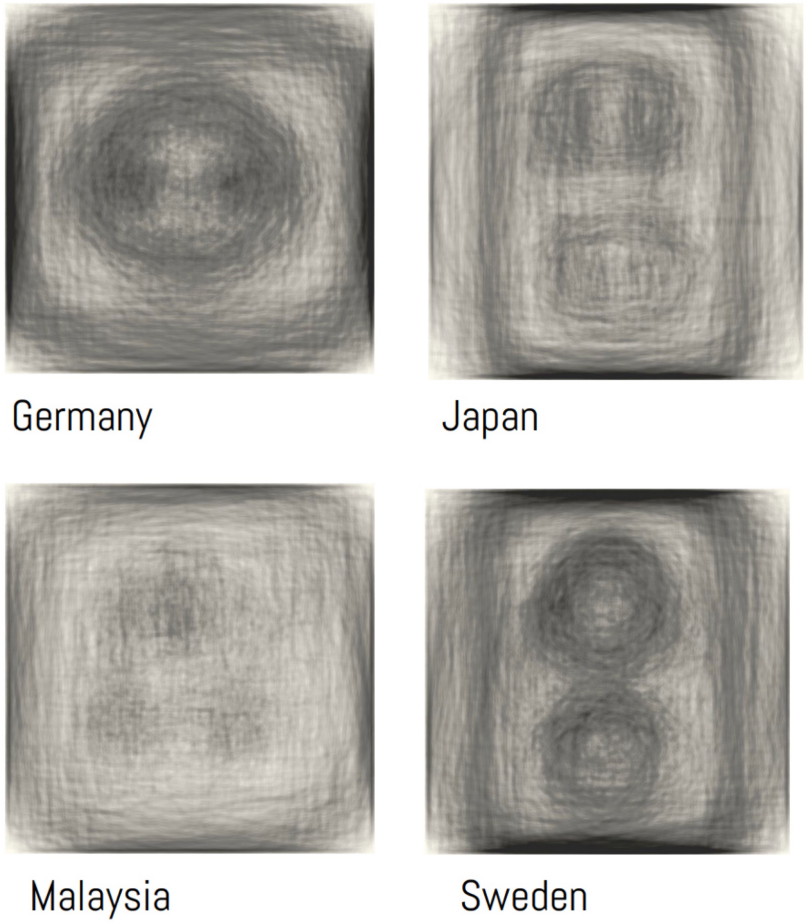
- Understanding scale



Source: http://hint.fm/wind/

Data Visualization: Examples

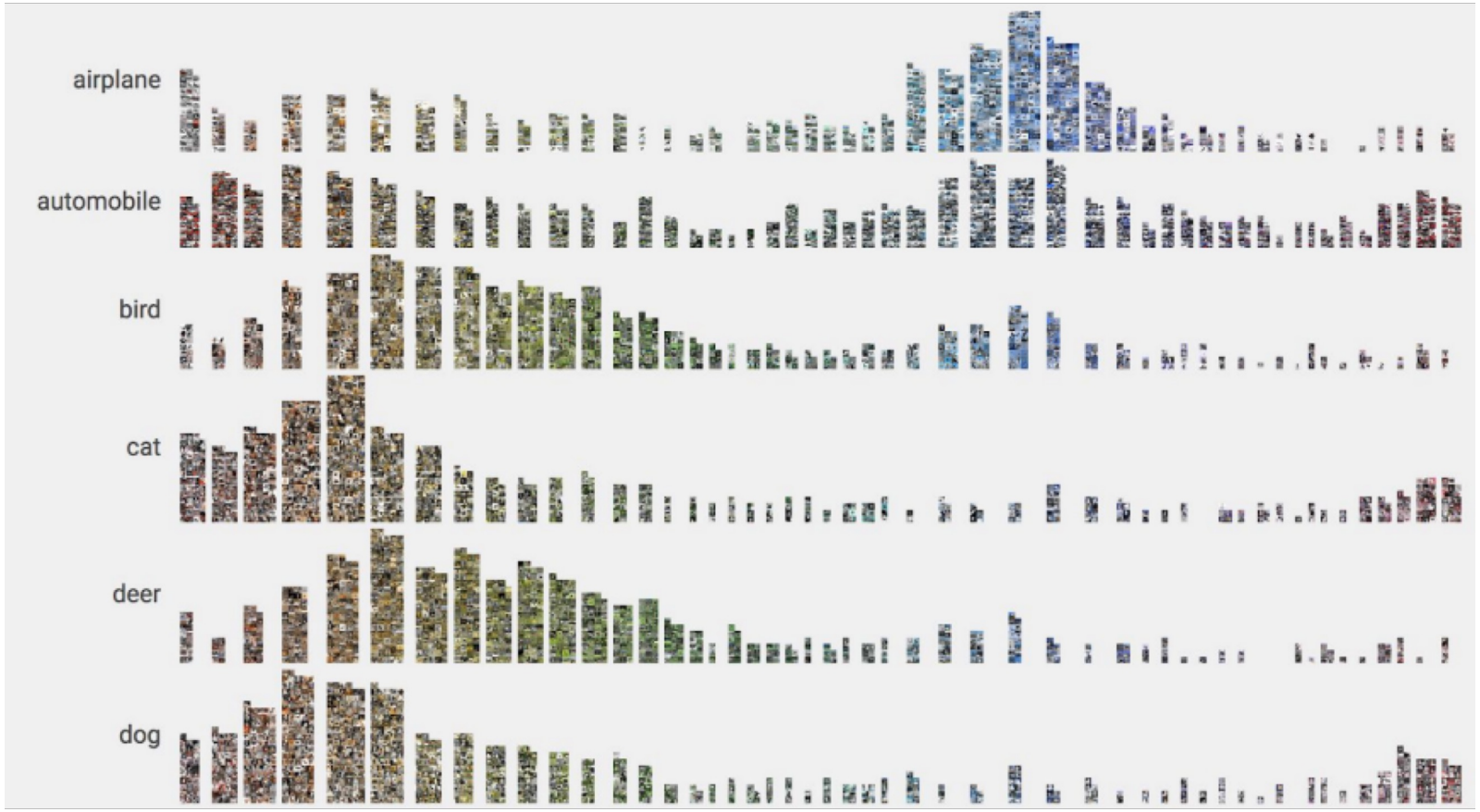
- Understanding clusters



Source: https://quickdraw.withgoogle.com/

Data Visualization: Examples

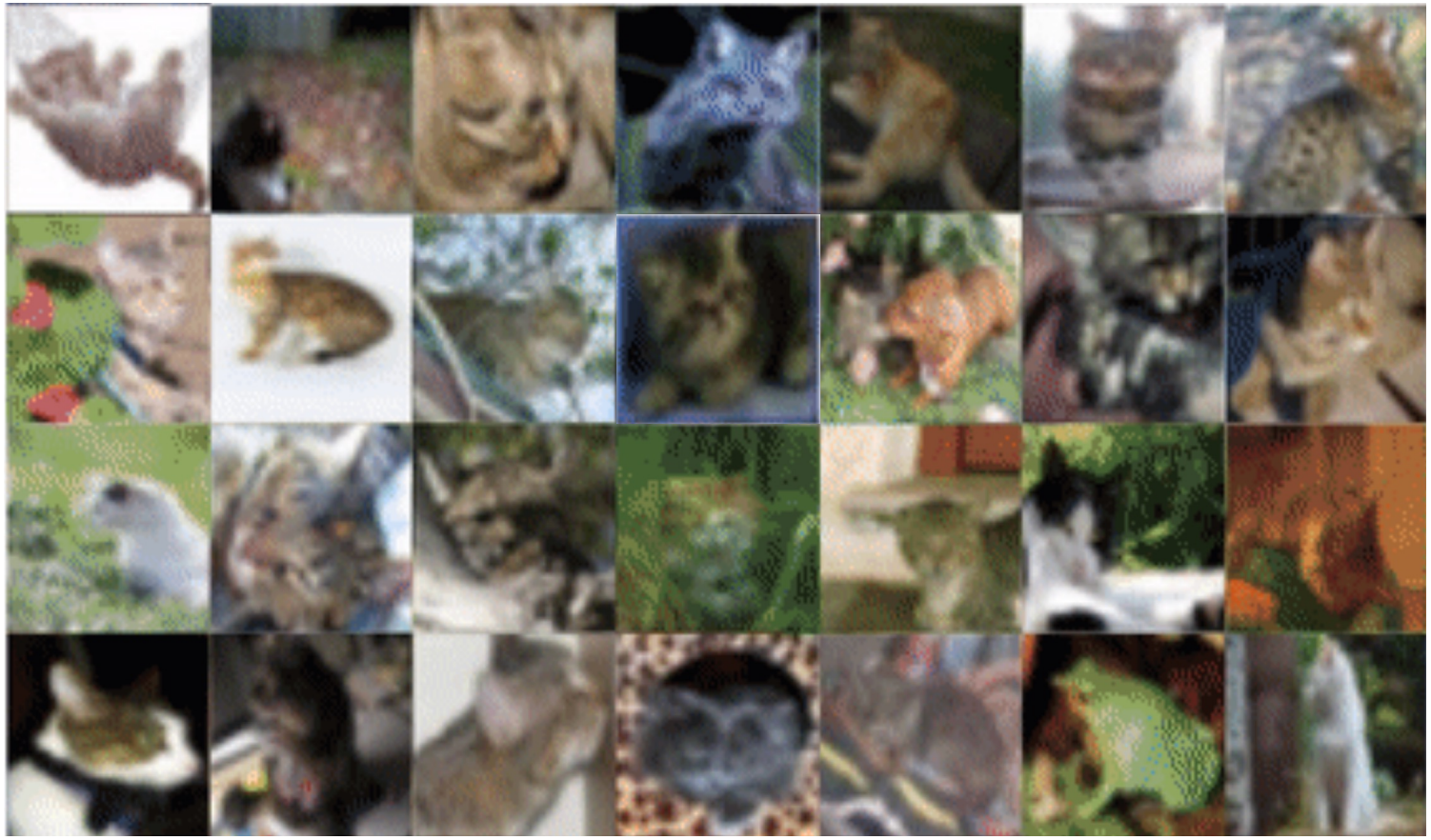
- Understanding variation



Source: https://pair-code.github.io/facets/

Data Visualization: Examples

- Understanding mistakes



Common Data Visualizations

- Summary statistics
- Box plots
- Histograms
- Scatter plots

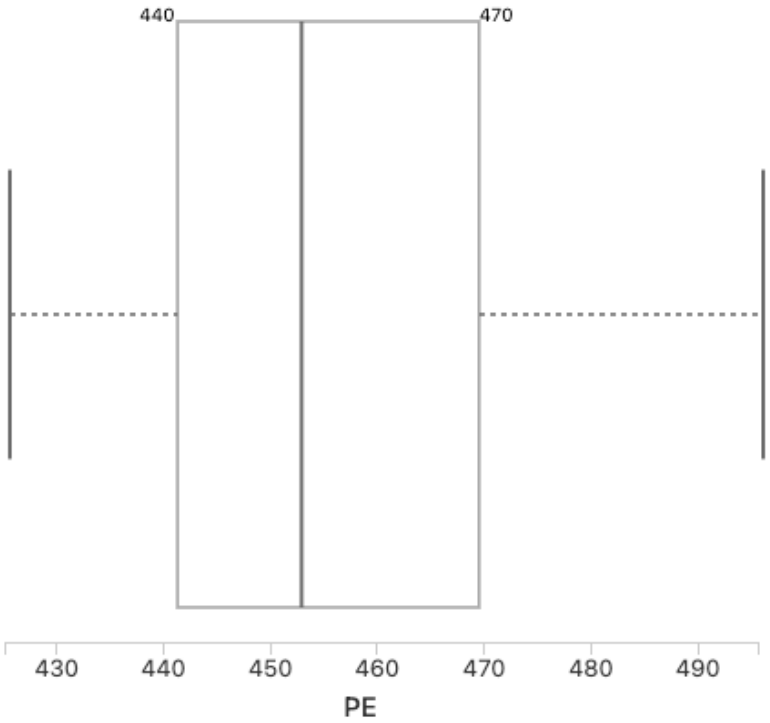
Common Data Visualizations

- Summary statistics
- Box plots
- Histograms
- Scatter plots

	AT	PE
Mean	19.65	454.37
Stddev	7.45	17.07
Min	1.81	420.26
Max	37.11	495.76

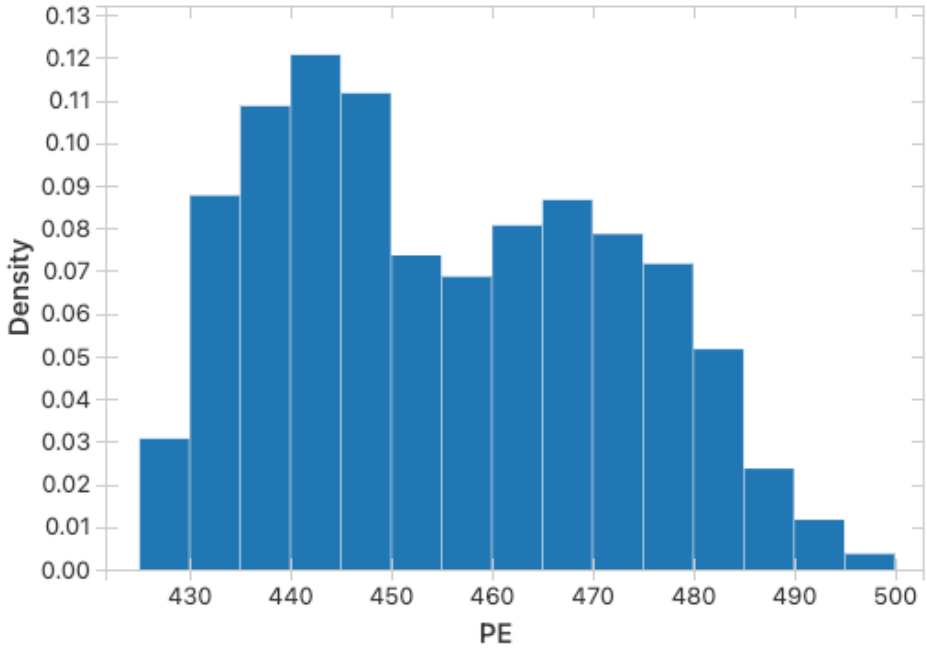
Common Data Visualizations

- Summary statistics
- Box plots
- Histograms
- Scatter plots



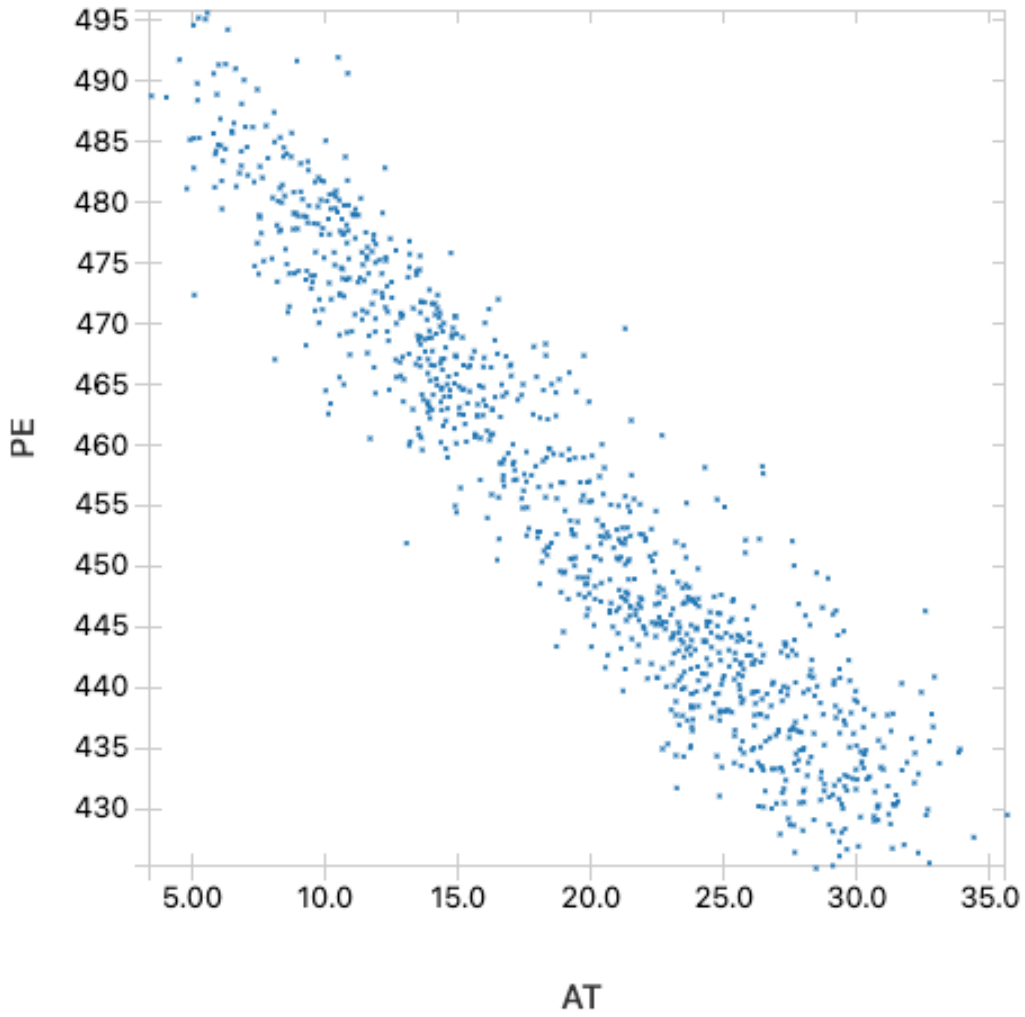
Common Data Visualizations

- Summary statistics
- Box plots
- Histograms
- Scatter plots



Common Data Visualizations

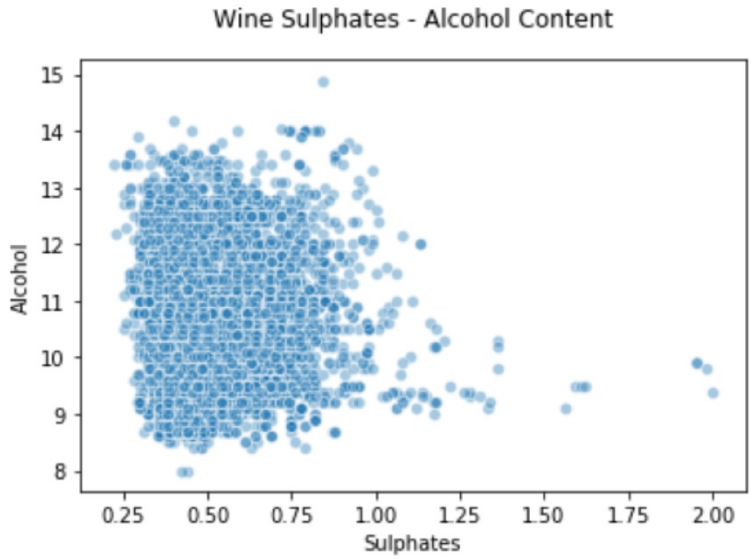
- Summary statistics
- Box plots
- Histograms
- Scatter plots

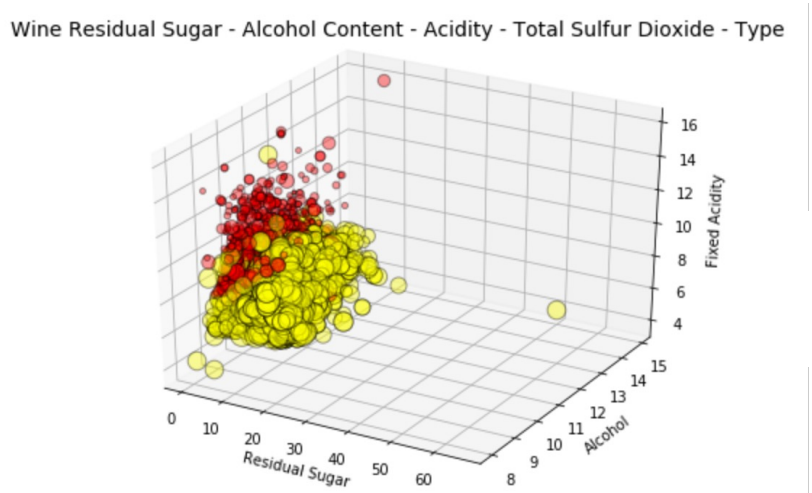


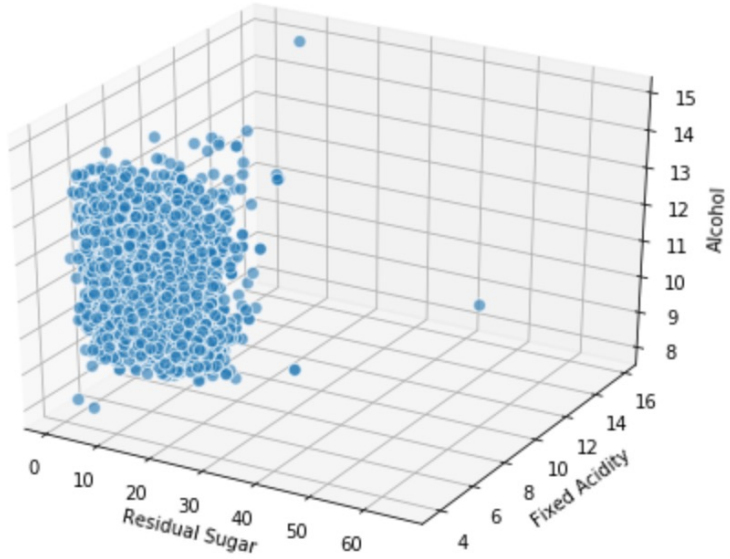
Big Data Visualizations

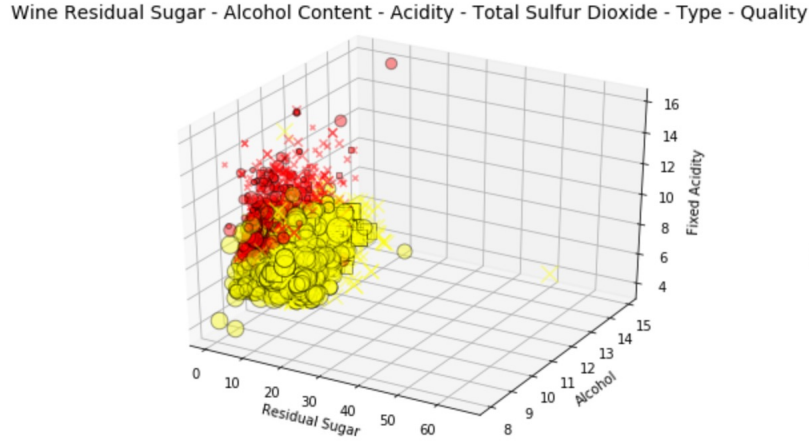
- Large n
  - Computationally expensive to render
  - Dense/complex
  - Address via subsampling or parallelization
- Large k
  - Difficult to represent more than a few dimensions

Big Data Visualizations



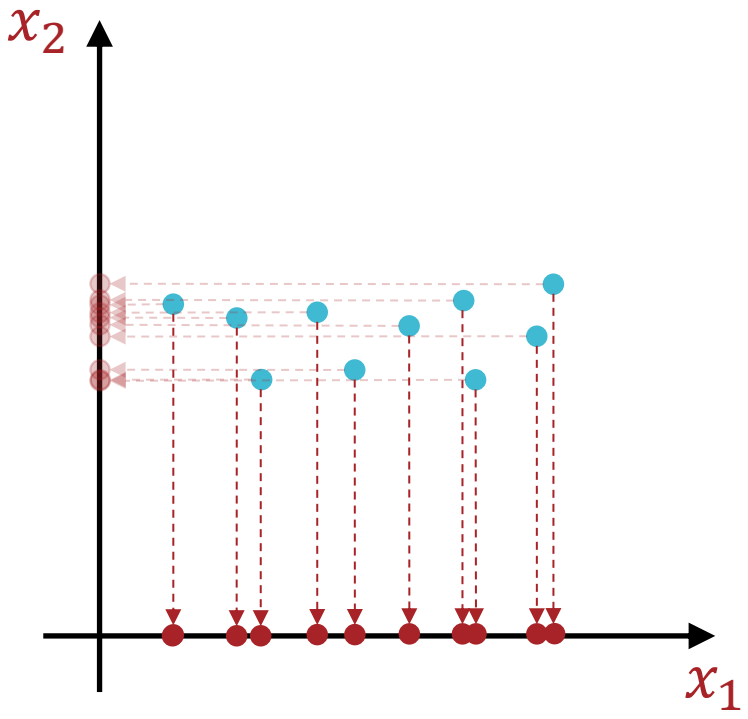


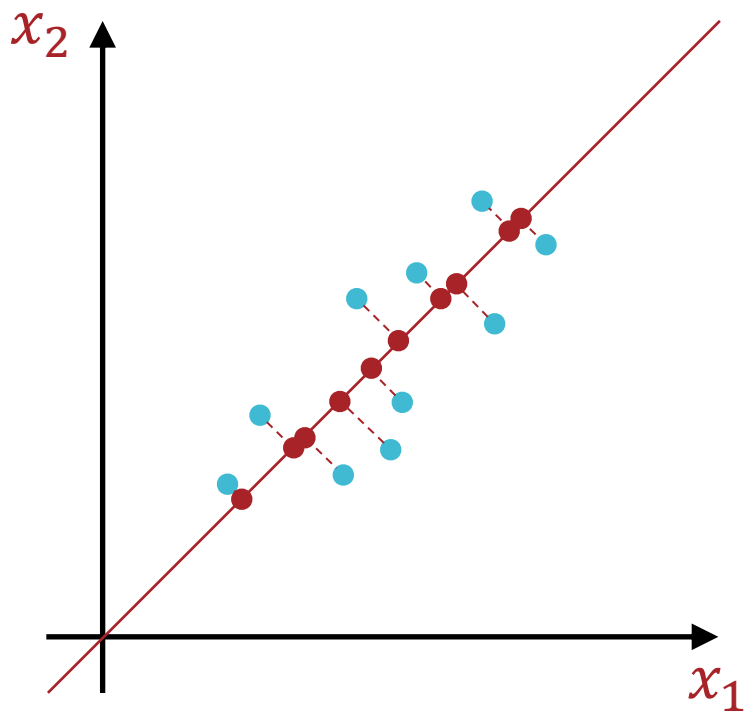




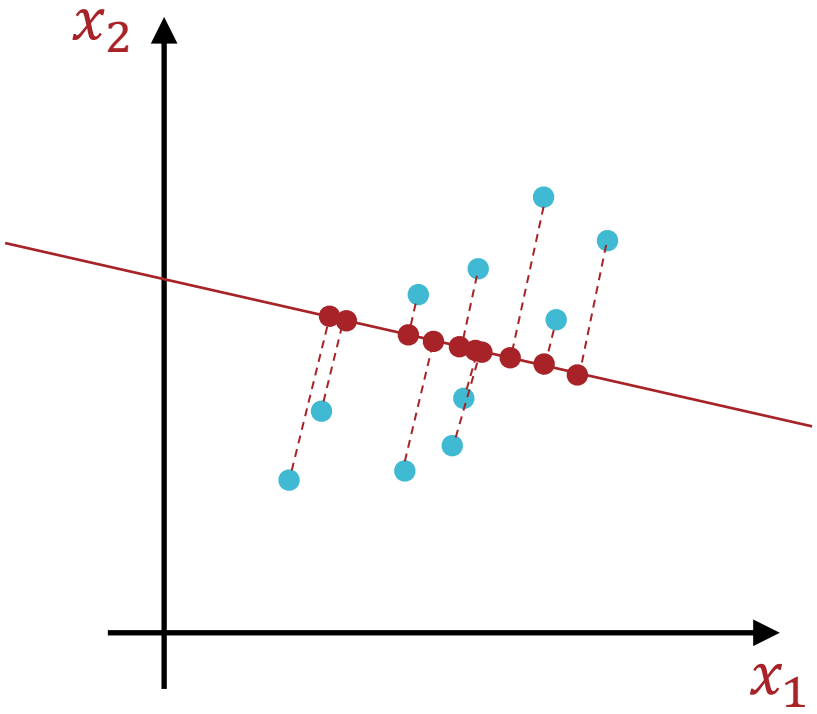
Big Data Visualizations

- Large n
  - Computationally expensive to render
  - Dense/complex
  - Address via subsampling or parallelization
- Large k
  - Difficult to represent more than a few dimensions
  - Address via dimensionality reduction = learning a latent (typically lower-dimensional) representation

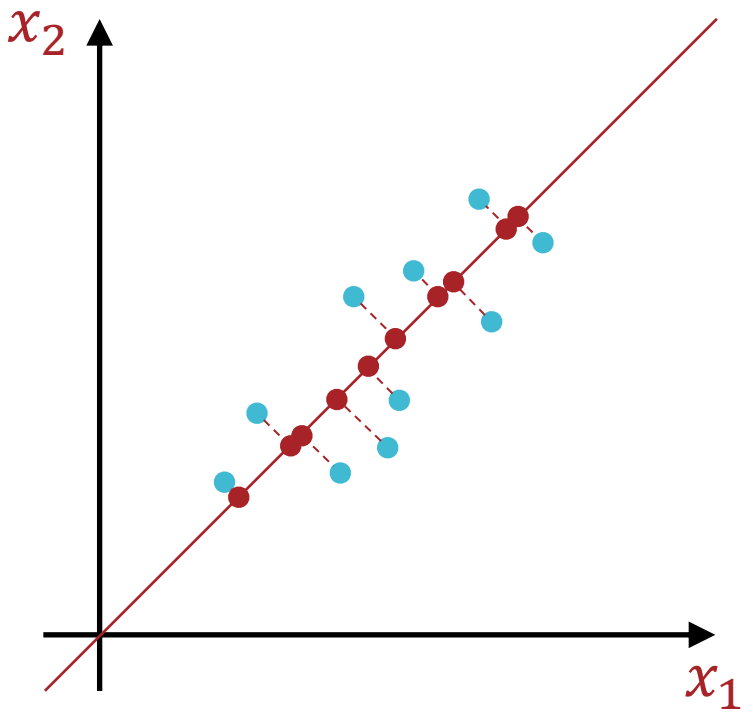




Feature Elimination

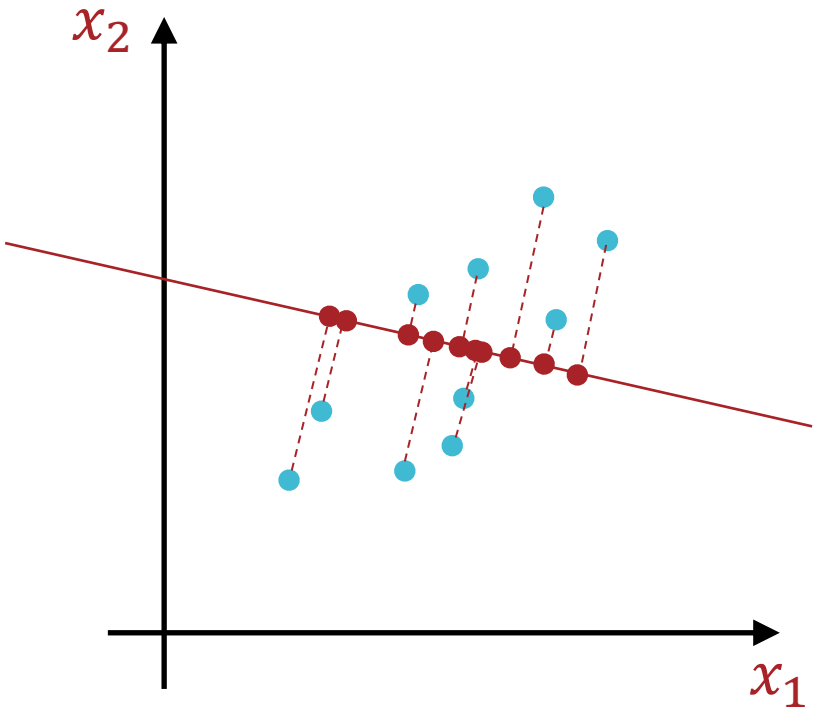


Option A

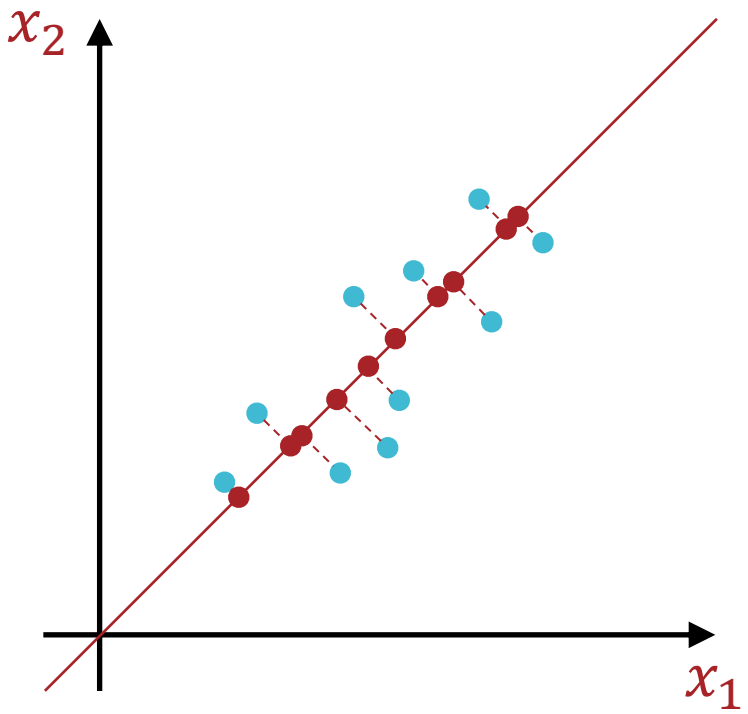


Option B

Dimensionality Reduction

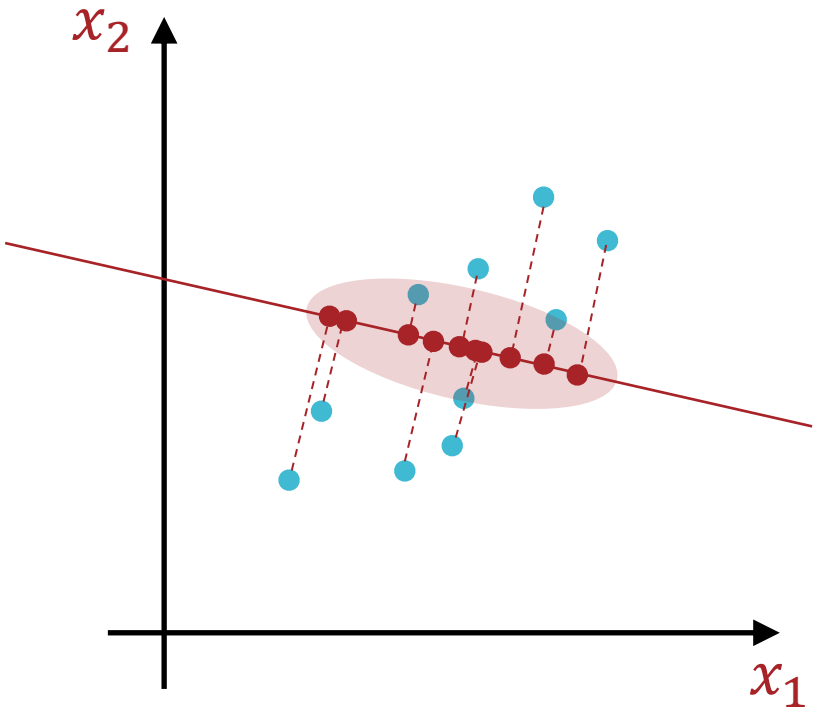


Option A

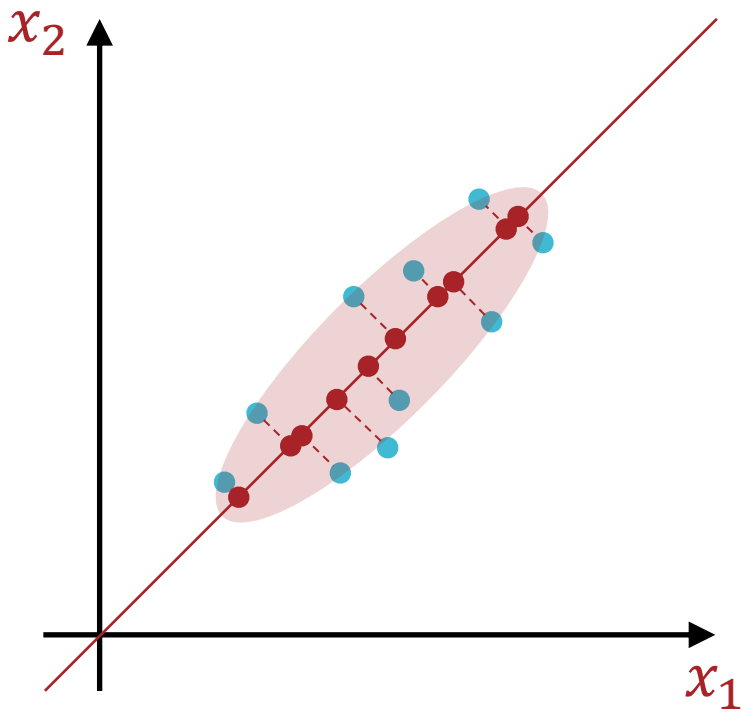


Option B

Goal: minimize the reconstruction error



Option A



Option B

Goal: maximize the variance of the projections

Centering the Data

- To be consistent, we will constrain principal components to be *orthonormal vectors* (orthogonal unit vectors) that begin at the origin
- Preprocess data to be centered around the origin:

1. $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}^{(i)}$
2. $\tilde{\boldsymbol{x}}^{(i)} = \boldsymbol{x}^{(i)} - \boldsymbol{\mu}\ \forall\ i$
3. $X = \begin{bmatrix} \tilde{\boldsymbol{x}}^{(1)^T} \\ \tilde{\boldsymbol{x}}^{(2)^T} \\ \vdots \\ \tilde{\boldsymbol{x}}^{(n)^T} \end{bmatrix} \in \mathbb{R}^{n \times k}$

Reconstruction Error

- The projection of $\tilde{\boldsymbol{x}}^{(i)}$ onto a vector $\boldsymbol{v}$ is

$$\boldsymbol{z}^{(i)} = \left(\frac{\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}}{\|\boldsymbol{v}\|_2} \right) \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2}$$

Length of projection

Direction of projection

Reconstruction Error

- The projection of $\tilde{\boldsymbol{x}}^{(i)}$ onto a unit vector $\boldsymbol{v}$ is

$$\boldsymbol{z}^{(i)} = \left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)\boldsymbol{v}$$

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1}{\operatorname{argmin}} \sum_{i=1}^{n} \left\|\tilde{\boldsymbol{x}}^{(i)} - \left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)\boldsymbol{v}\right\|_2^2$$

$$\left\|\tilde{\boldsymbol{x}}^{(i)} - \left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)\boldsymbol{v}\right\|_2^2$$

$$= \tilde{\boldsymbol{x}}^{(i)^T}\tilde{\boldsymbol{x}}^{(i)} - 2\left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)\boldsymbol{v}^T\tilde{\boldsymbol{x}}^{(i)} + \left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)\left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)\boldsymbol{v}^T\boldsymbol{v}$$

$$= \tilde{\boldsymbol{x}}^{(i)^T}\tilde{\boldsymbol{x}}^{(i)} - \left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)\boldsymbol{v}^T\tilde{\boldsymbol{x}}^{(i)}$$

$$= \left\|\tilde{\boldsymbol{x}}^{(i)}\right\|_2^2 - \left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)^2$$

Minimizing the Reconstruction Error

⇕

Maximizing the Variance

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1}{\operatorname{argmin}} \sum_{i=1}^{n} \left\|\tilde{\boldsymbol{x}}^{(i)}\right\|_2^2 - \left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)^2$$

$$= \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1}{\operatorname{argmax}} \sum_{i=1}^{n} \left(\boldsymbol{v}^T \tilde{\boldsymbol{x}}^{(i)}\right)^2 \longleftarrow \text{Variance of projections } (\tilde{\boldsymbol{x}}^{(i)} \text{ are centered})$$

$$= \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1}{\operatorname{argmax}} \boldsymbol{v}^T \left(\sum_{i=1}^{n} \tilde{\boldsymbol{x}}^{(i)} \tilde{\boldsymbol{x}}^{(i)^T}\right) \boldsymbol{v}$$

$$= \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1}{\operatorname{argmax}} \boldsymbol{v}^T (X^T X) \boldsymbol{v}$$

$$= \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1}{\operatorname{argmax}} \boldsymbol{v}^T C_X \boldsymbol{v} \quad \text{where } C_X = X^T X \text{ is the } \textit{covariance matrix}$$

Maximizing the Variance

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1}{\operatorname{argmax}} \boldsymbol{v}^T C_X \boldsymbol{v}$$

$$\begin{aligned}\mathcal{L}(\boldsymbol{v}, \lambda) &= \boldsymbol{v}^T C_X \boldsymbol{v} - \lambda(\|\boldsymbol{v}\|_2^2 - 1) \\ &= \boldsymbol{v}^T C_X \boldsymbol{v} - \lambda(\boldsymbol{v}^T \boldsymbol{v} - 1)\end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}} = C_X \boldsymbol{v} - \lambda \boldsymbol{v}$$

$$\rightarrow C_X \hat{\boldsymbol{v}} - \lambda \hat{\boldsymbol{v}} = 0 \rightarrow C_X \hat{\boldsymbol{v}} = \lambda \hat{\boldsymbol{v}}$$

- $\hat{\boldsymbol{v}}$ is an eigenvector of C_X and λ is the corresponding eigenvalue! But which one?

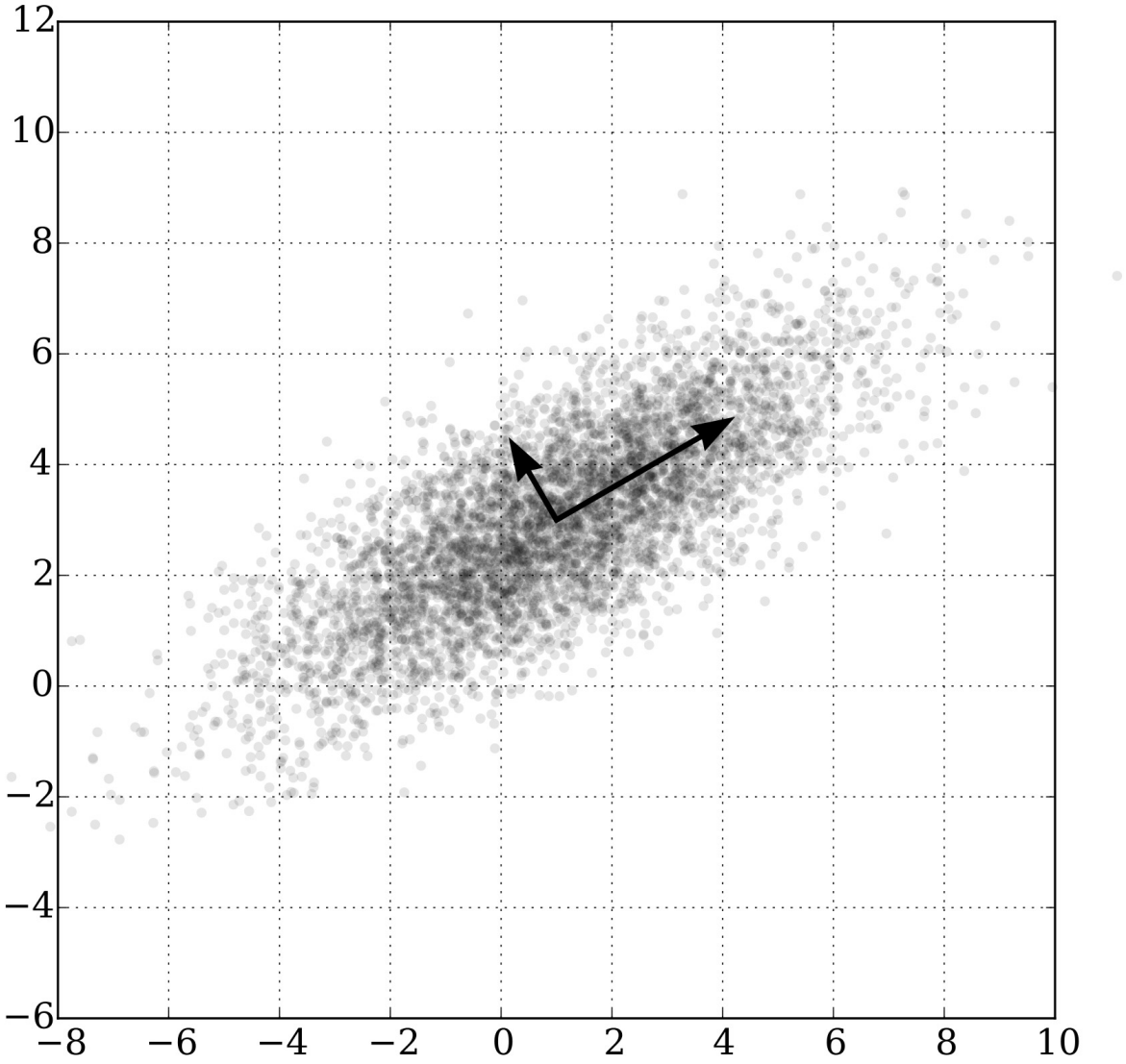
Maximizing the Variance

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1}{\operatorname{argmax}} \boldsymbol{v}^T C_X \boldsymbol{v}$$

$$C_X\hat{\boldsymbol{v}} = \lambda\hat{\boldsymbol{v}} \rightarrow \hat{\boldsymbol{v}}^T C_X\hat{\boldsymbol{v}} = \lambda\hat{\boldsymbol{v}}^T\hat{\boldsymbol{v}} = \lambda$$

- The first principal component is the eigenvector $\hat{\boldsymbol{v}}_1$ that corresponds to the largest eigenvalue λ_1
- The second principal component is the eigenvector $\hat{\boldsymbol{v}}_2$ that corresponds to the second largest eigenvalue λ_1
  - $\hat{\boldsymbol{v}}_1$ and $\hat{\boldsymbol{v}}_2$ are orthonormal!
- Etc …
- λ_i is a measure of how much variance falls along $\hat{\boldsymbol{v}}_i$

Principal Components: Example



Source: https://en.wikipedia.org/wiki/Principal_component_analysis#/media/File:GaussianScatterPCA.svg

Aside: Sparse PCA

$$\hat{\boldsymbol{v}} = \underset{\boldsymbol{v}:\|\boldsymbol{v}\|_2^2=1 \text{ and } \|\boldsymbol{v}\|_0 \leq s}{\operatorname{argmax}} \boldsymbol{v}^T C_X \boldsymbol{v}$$

- L0-norm constraint: $\boldsymbol{v}$ can contain at most s non-zero elements
- Reduces to standard PCA if $s = k$
- Sparse principal components may be easier to interpret and can also reduce data pre-processing needs

PCA Algorithm

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}\right)\right\}_{i=1}^{n}, r$

1. Center the data
   A. Optionally, normalize the data by features so that all features are of the same scale
2. Compute the covariance matrix $C_X = X^T X$
3. Collect the top r eigenvectors (corresponding to the r largest eigenvalues), $P \in \mathbb{R}^{k \times r}$
4. Project the data into the space defined by P, $Z = XP$

- Output: Z, the latent representation ("PCA scores")

How many PCs should we use?

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}\right)\right\}_{i=1}^{n}, r$

1. Center the data
   A. Optionally, normalize the data by features so that all features are of the same scale
2. Compute the covariance matrix $C_X = X^T X$
3. Collect the top r eigenvectors (corresponding to the r largest eigenvalues), $P \in \mathbb{R}^{k \times r}$
4. Project the data into the space defined by P, $Z = XP$

- Output: Z, the latent representation ("PCA scores")

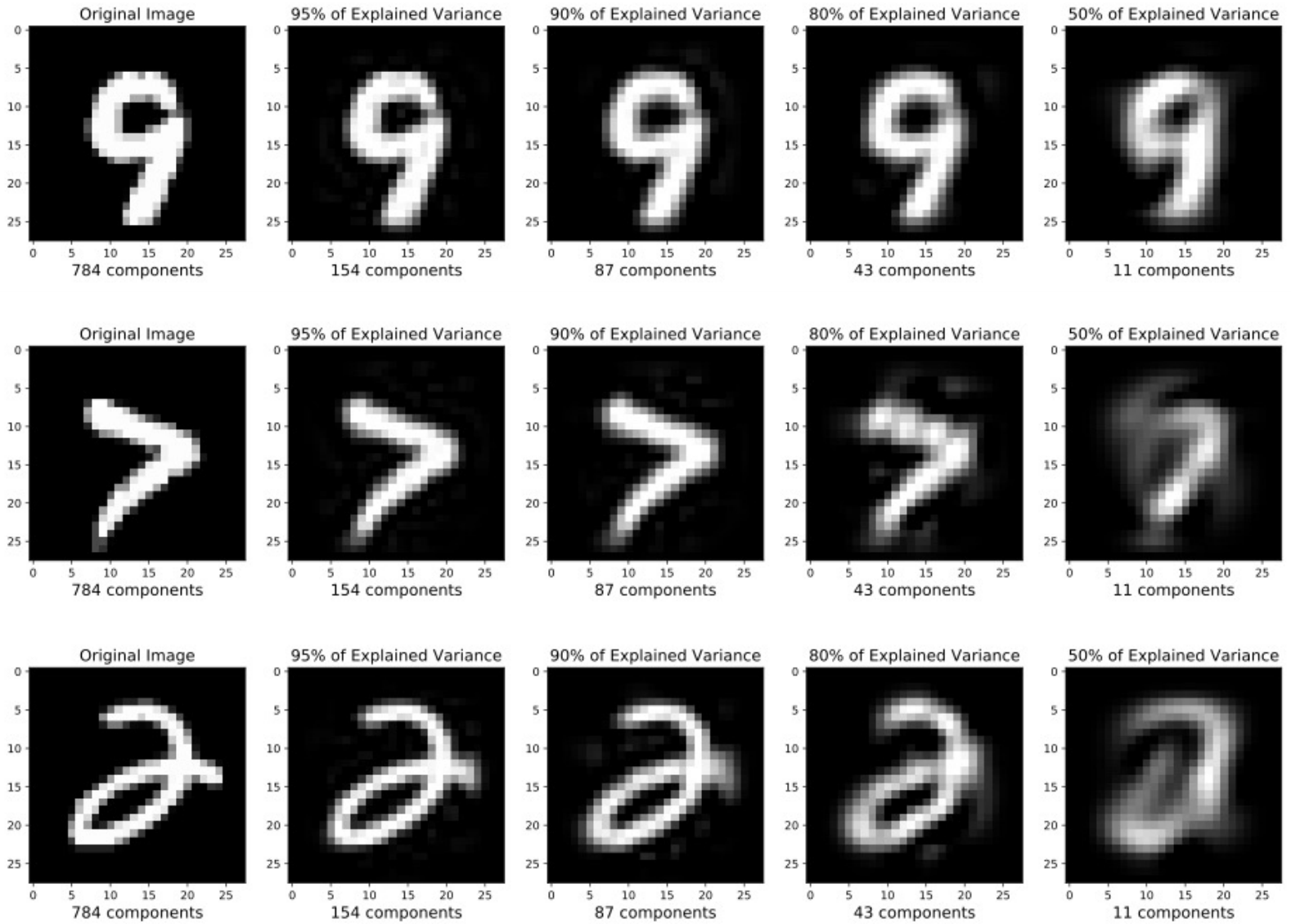
Choosing the number of PCs

- Define a percentage of explained variance for the i^{th} PC:

$$\lambda_i \Big/ \sum \lambda_j$$

- Select all PCs above some threshold of explained variance, e.g., 5%
- Keep selecting PCs until the total explained variance exceeds some threshold, e.g., 90%
- Evaluate on some downstream metric

PCA Example: MNIST Digits



Figures courtesy of Matt Gormley

PCA Example: MNIST Digits

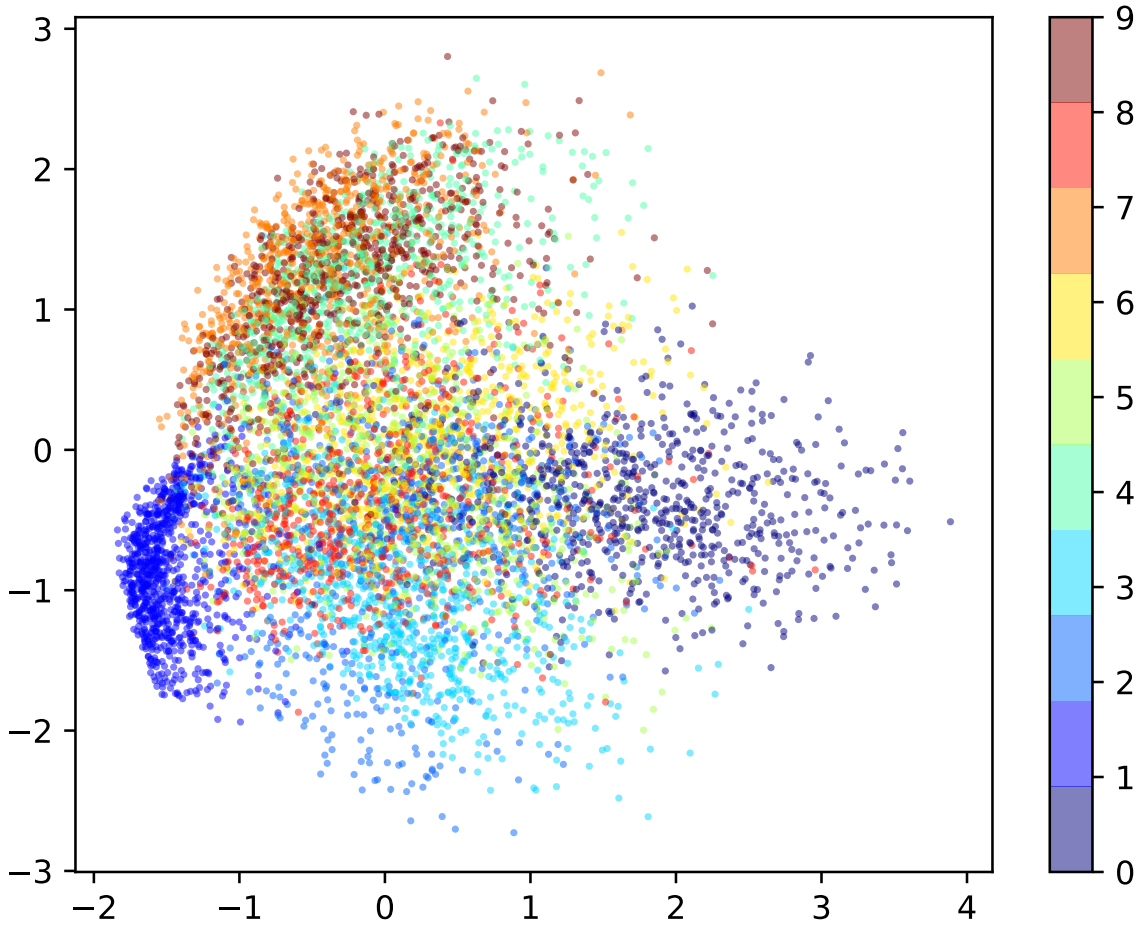


Figure courtesy of Matt Gormley

PCA Example: MNIST Digits

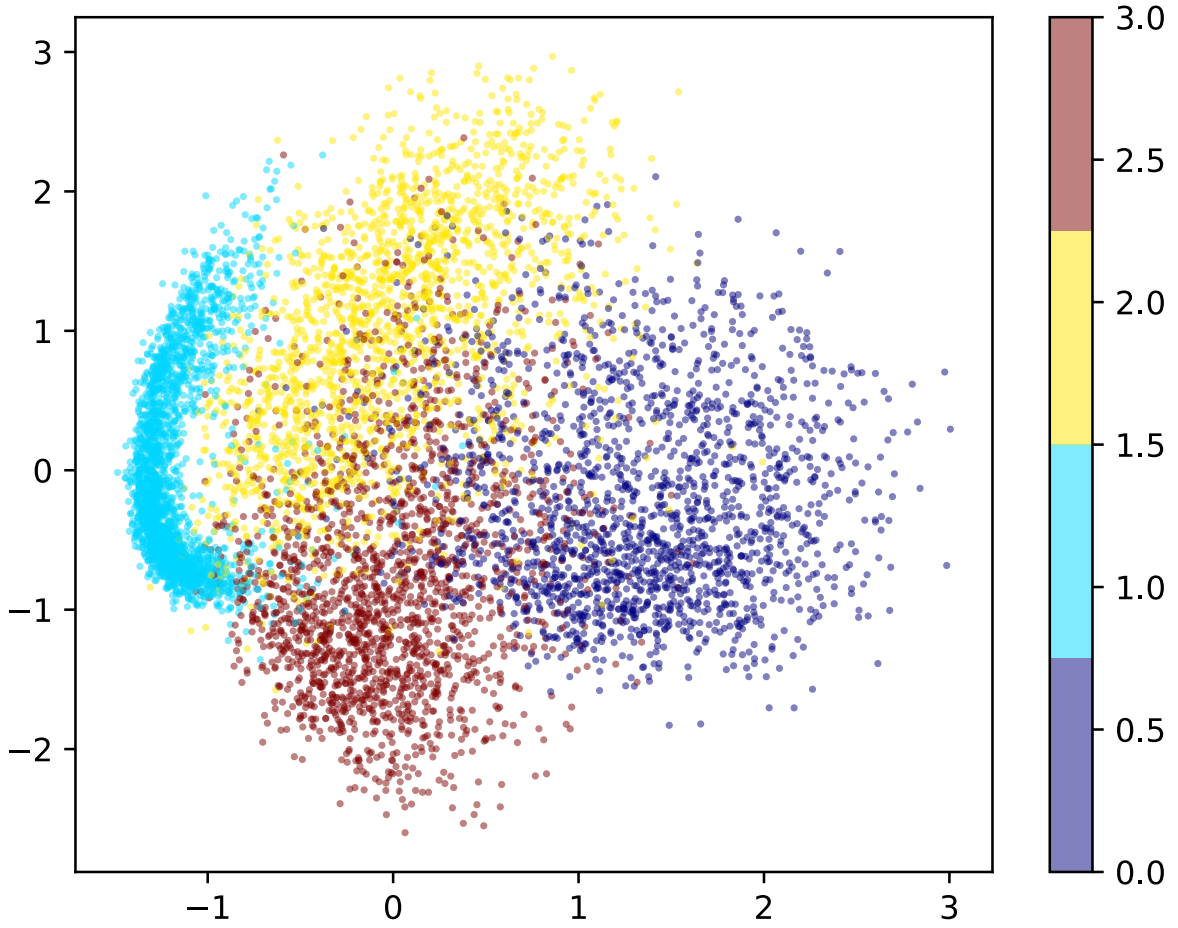
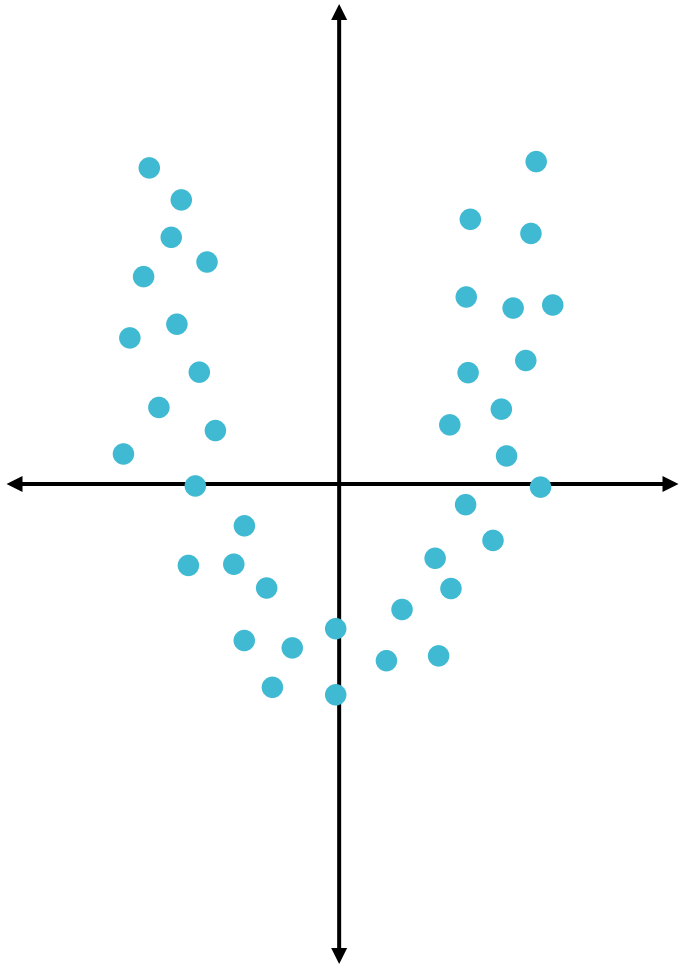


Figure courtesy of Matt Gormley

Shortcomings of PCA

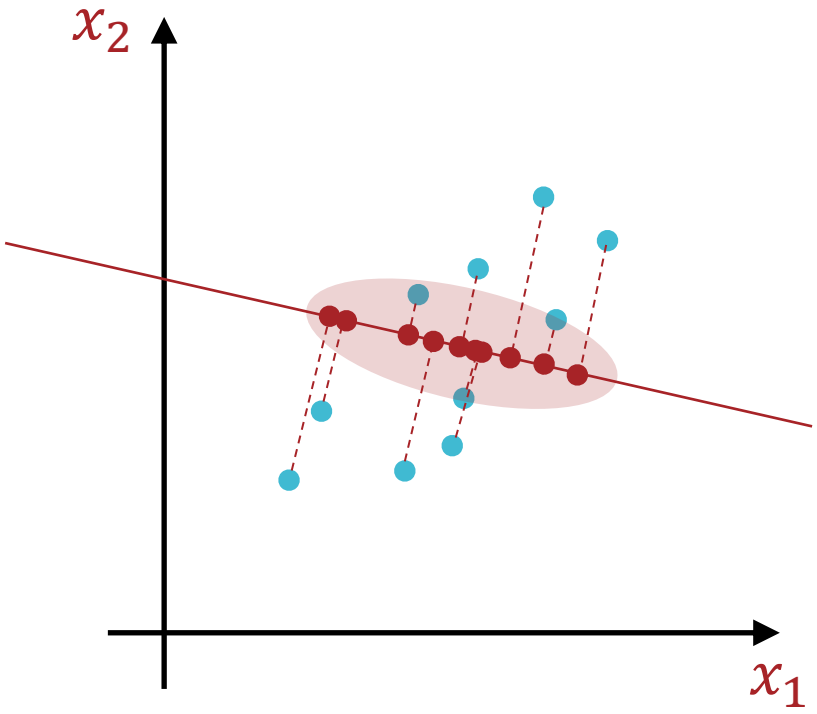


- Principal components are orthonormal
- Principal components are linear combinations of the features
- Principal components are expensive to compute

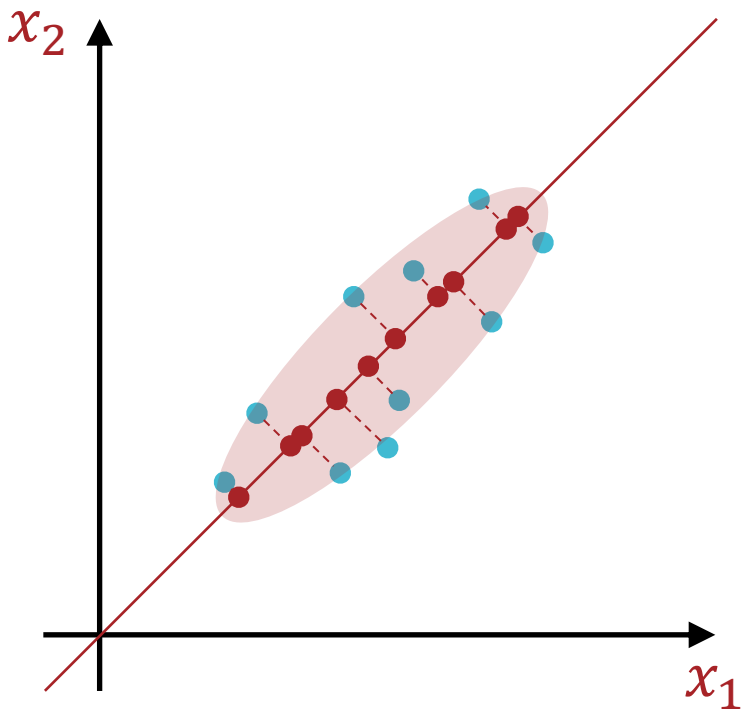
PCA Algorithm: Computational Cost

- Input: $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}\right)\right\}_{i=1}^{n}, r$

1. Center the data
   A. Optionally, normalize the data by features so that all features are of the same scale
2. Compute the covariance matrix $C_X = X^T X$ $(O(nk^2))$
3. Collect the top r eigenvectors (corresponding to the r largest eigenvalues), $P \in \mathbb{R}^{k \times r}$ $(O(k^3))$
4. Project the data into the space defined by P, $Z = XP$ $(O(nkr))$



Option A



Option B

Maybe Option A isn't so bad?

Key Takeaways

- Visualization is a key component of data pre-processing
  - Visualizing big data presents unique challenges
- PCA is dimensionality reduction technique that finds an orthonormal latent representation
  - Minimizes reconstruction error ↔ maximizing the projection variance
  - Constrained to find projections that are linear combinations of the existing features
  - Computationally expensive (cubic in the number of features)