6.7900: Machine Learning
Lecture 19

Lecture start: Tues/Thurs 2:35pm
Who's speaking today? Prof. Tamara Broderick
Course website: gradml.mit.edu
Questions? Ask here or on piazza.com/mit/fall2024/67900/
Materials: Slides, video, etc linked from gradml.mit.edu after the lecture (but there is no livestream)

Last Time

I. How to proceed when data is missing
II. Dimensionality reduction
III. Principal components analysis (PCA)

Today

I. Another perspective on PCA
II. Successes and challenges of PCA
III. t-SNE

Motivating example for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

Motivating example for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon

Motivating example

for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon

exploratory data analysis

Motivating example

for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.

exploratory data analysis

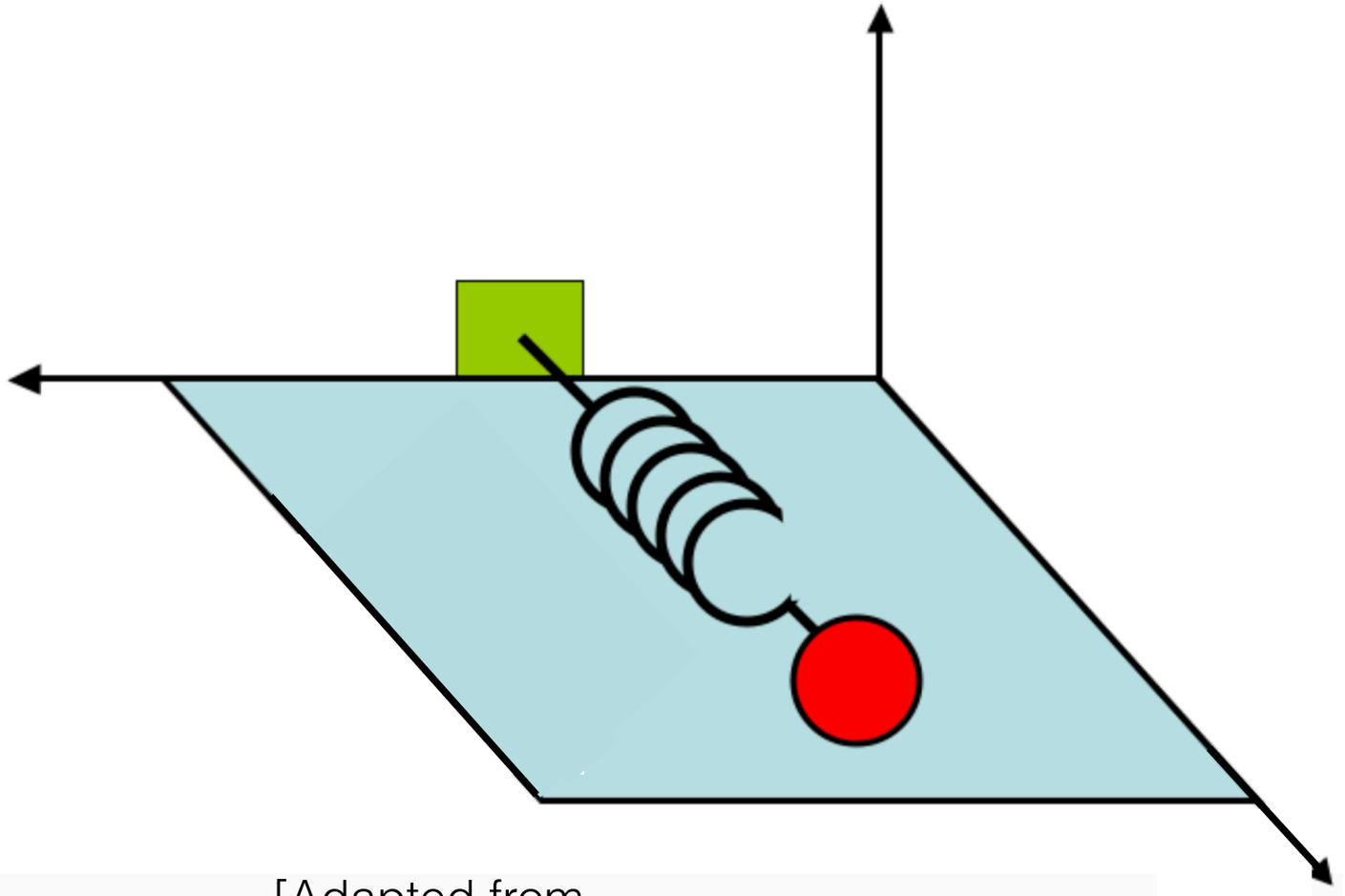
Motivating example

for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.

exploratory data analysis



[Adapted from Schlens 2014]

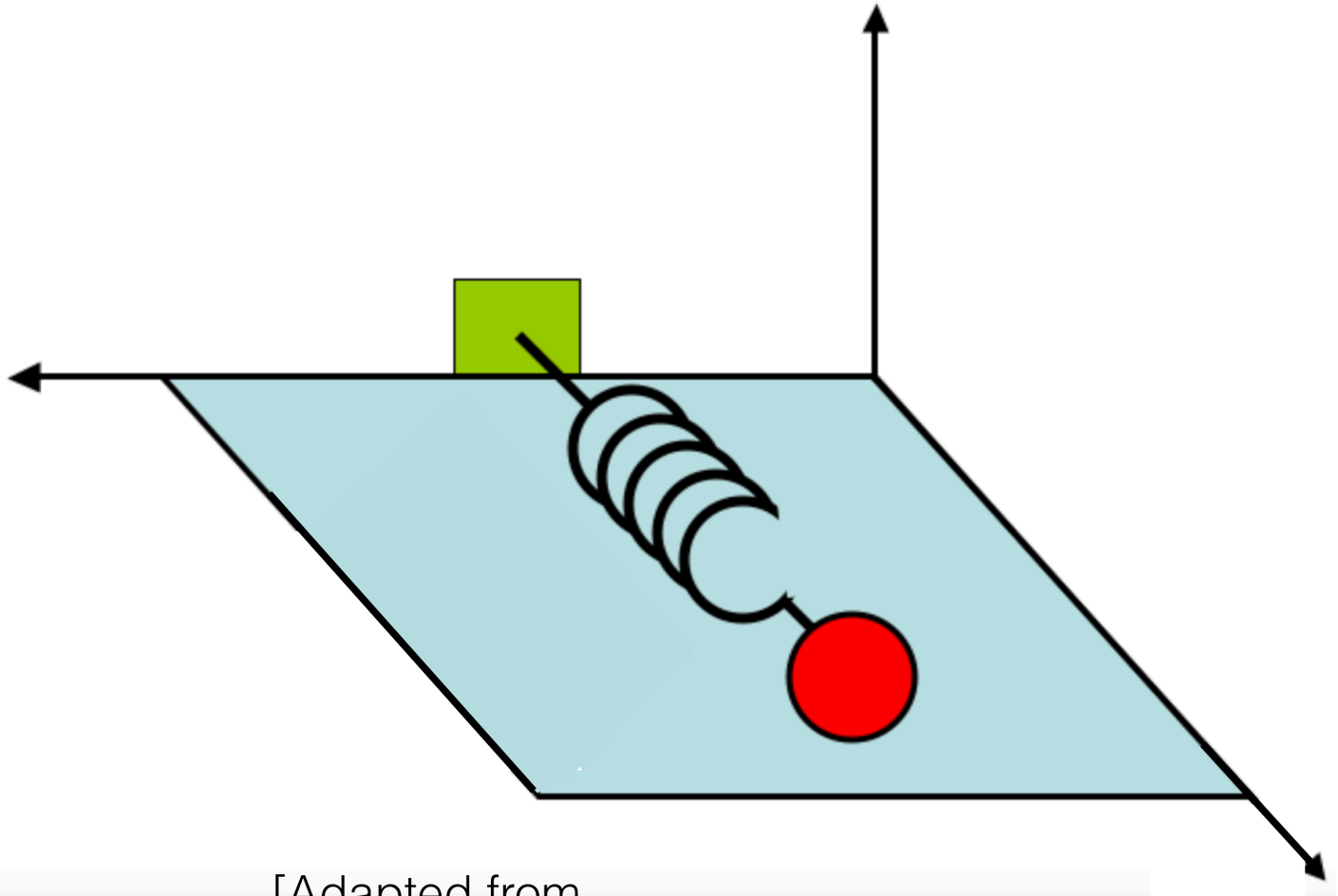
Motivating example

for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.
  - There are some true, unknown axes w_1, w_2, w_3 such that the ball and spring movement are very largely in w_1.

exploratory data analysis



[Adapted from Schlens 2014]

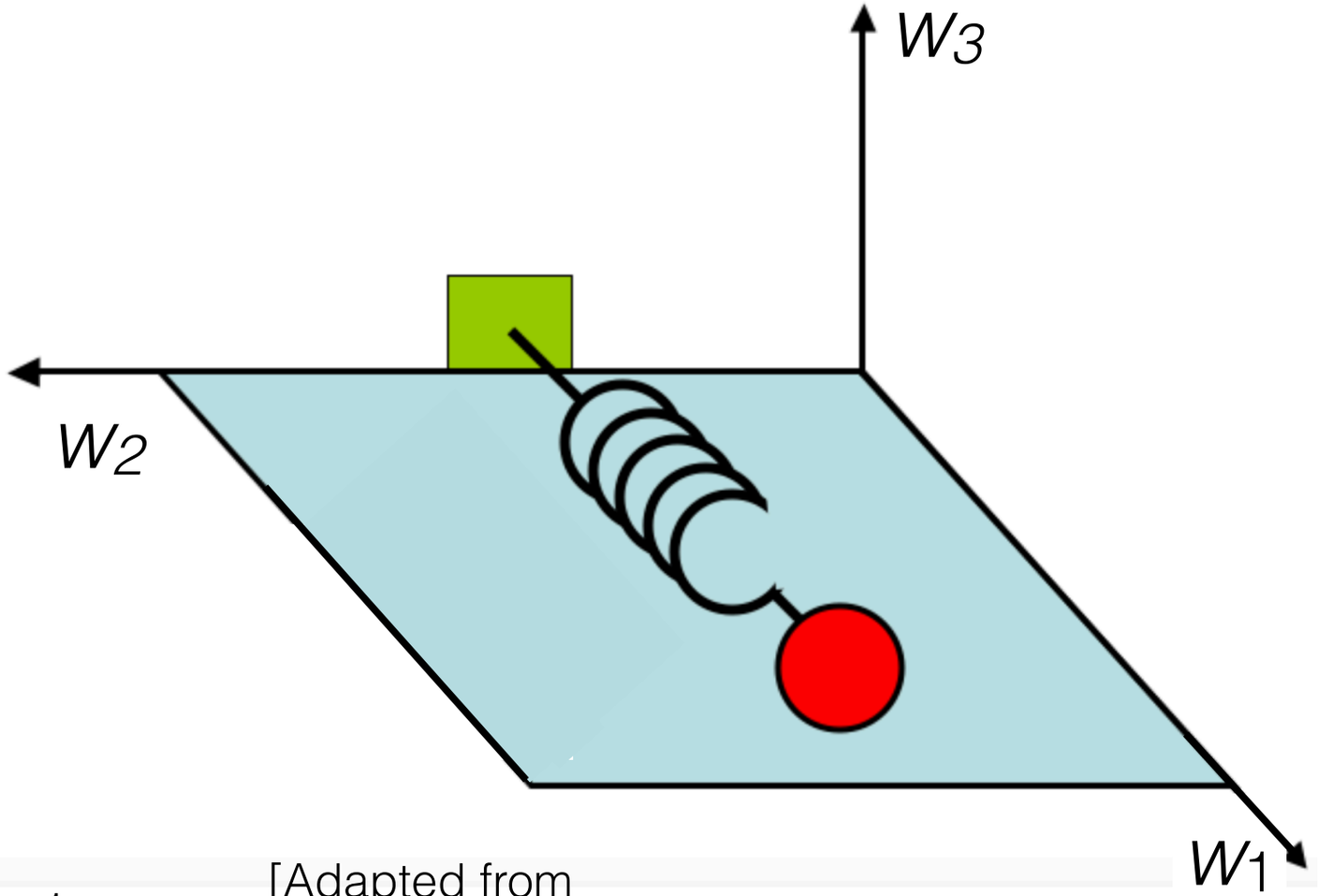
Motivating example

for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.
  - There are some true, unknown axes w_1, w_2, w_3 such that the ball and spring movement are very largely in w_1.

exploratory data analysis



[Adapted from Schlens 2014]

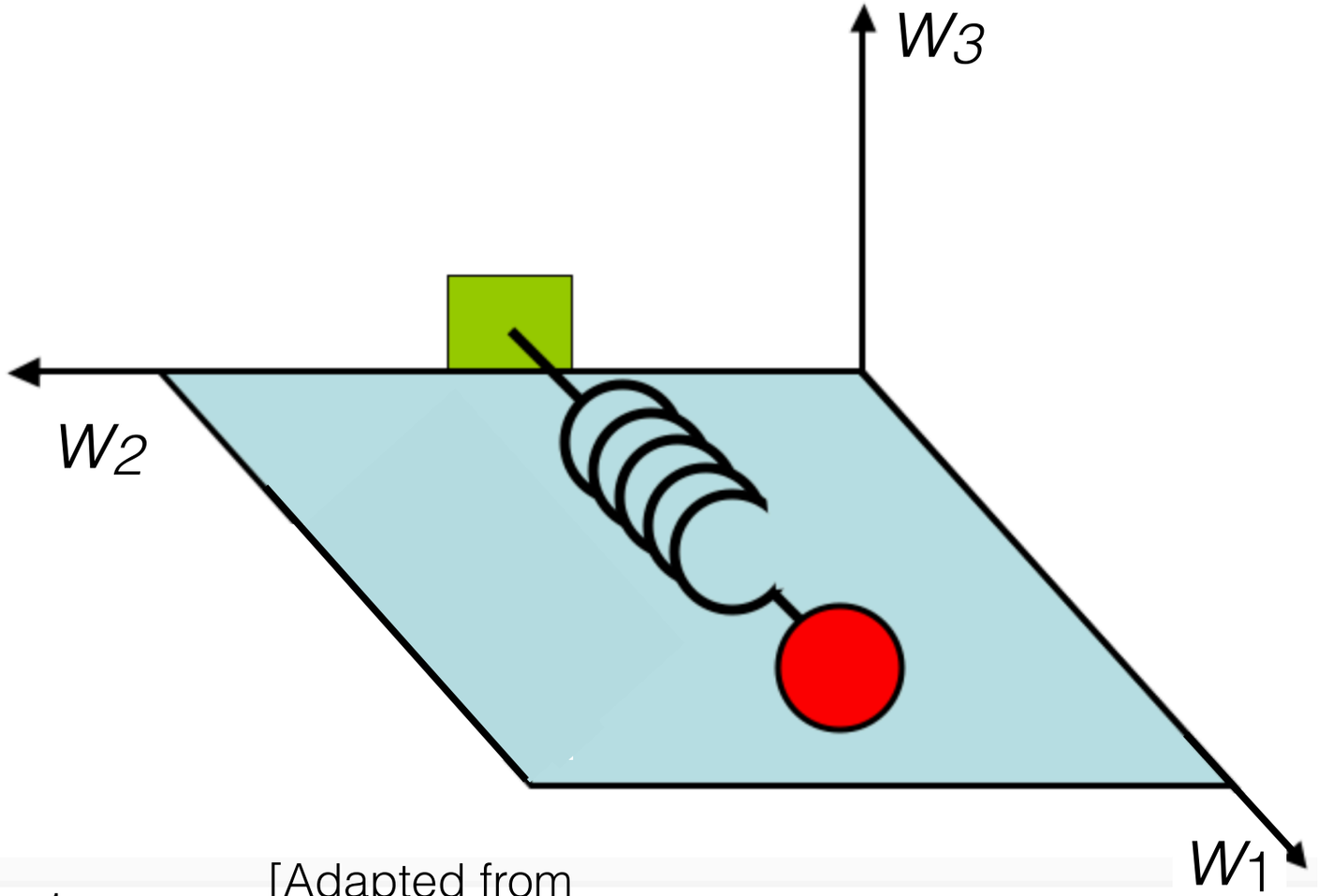
Motivating example

for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.
  - There are some true, unknown axes w_1, w_2, w_3 such that the ball and spring movement are very largely in w_1.
  - We set up three cameras, take occasional snapshots.

exploratory data analysis



[Adapted from Schlens 2014]

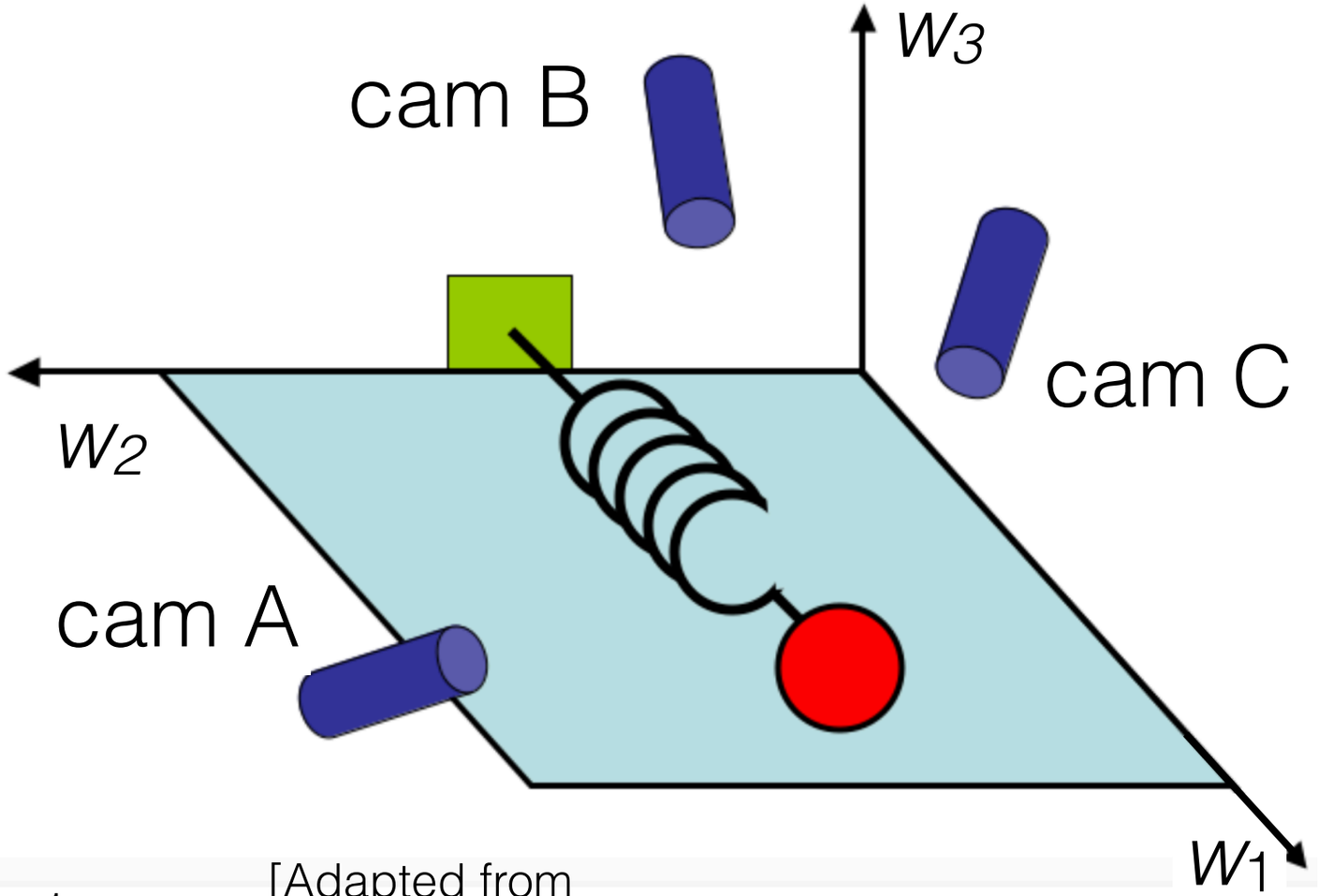
Motivating example

for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.
  - There are some true, unknown axes w_1, w_2, w_3 such that the ball and spring movement are very largely in w_1.
  - We set up three cameras, take occasional snapshots.

exploratory data analysis



[Adapted from Schlens 2014]

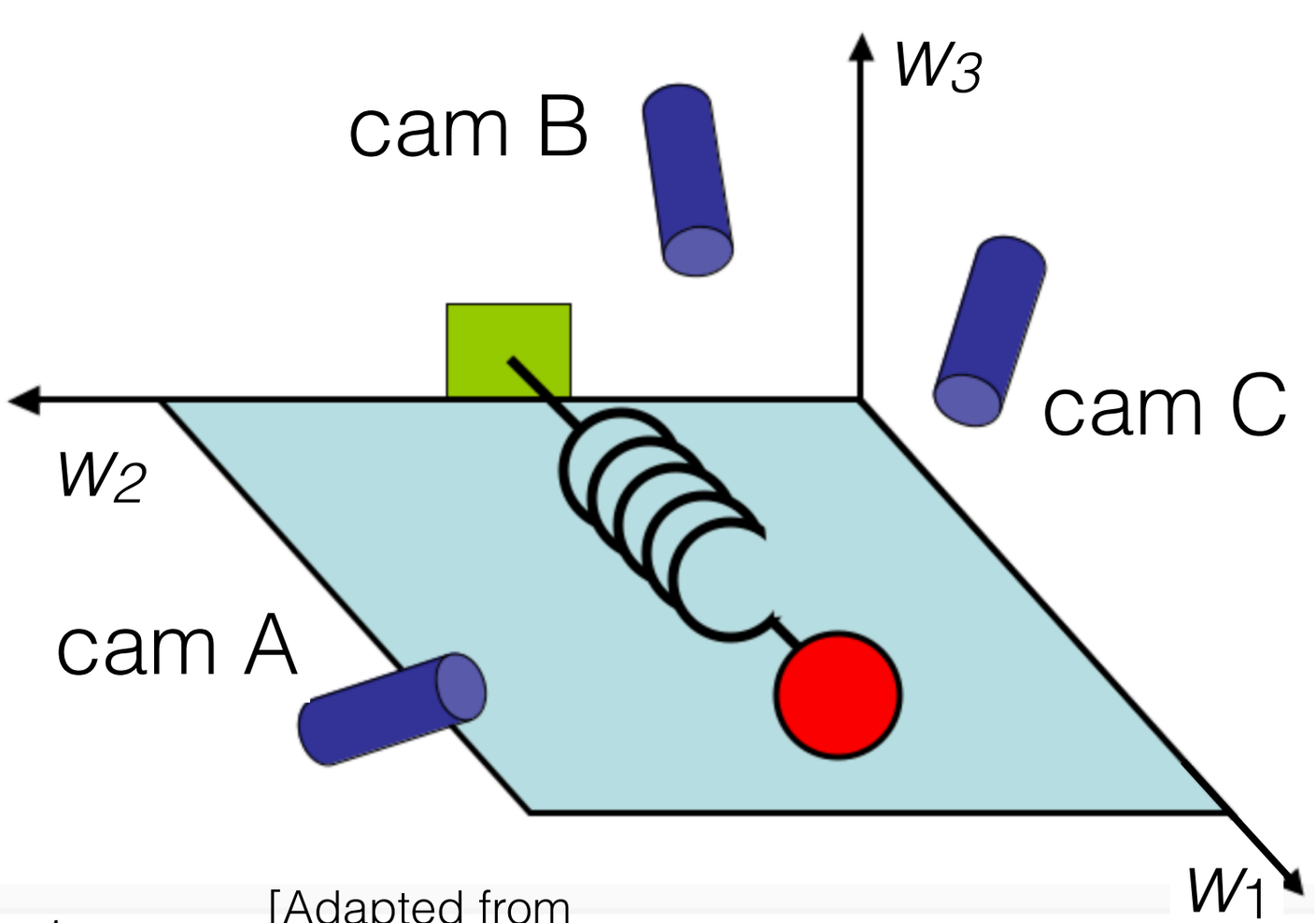
Motivating example

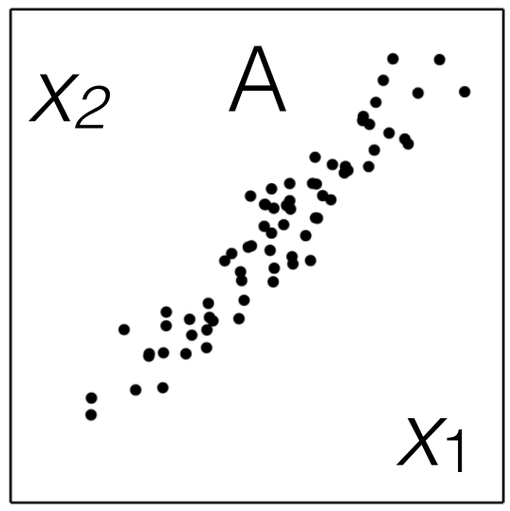
for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.
  - There are some true, unknown axes w_1, w_2, w_3 such that the ball and spring movement are very largely in w_1.
  - We set up three cameras, take occasional snapshots.

exploratory data analysis





[Adapted from Schlens 2014]

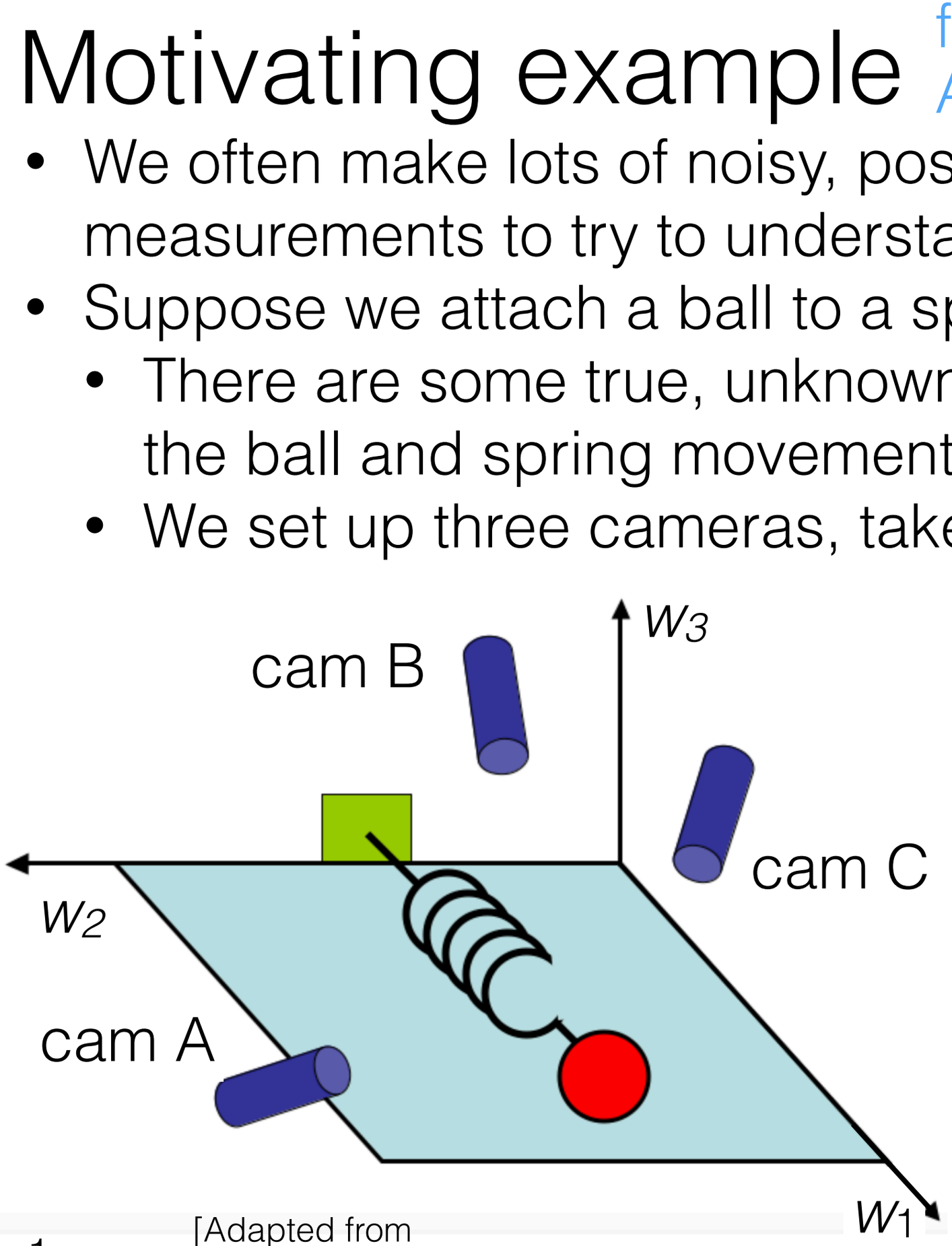
Motivating example

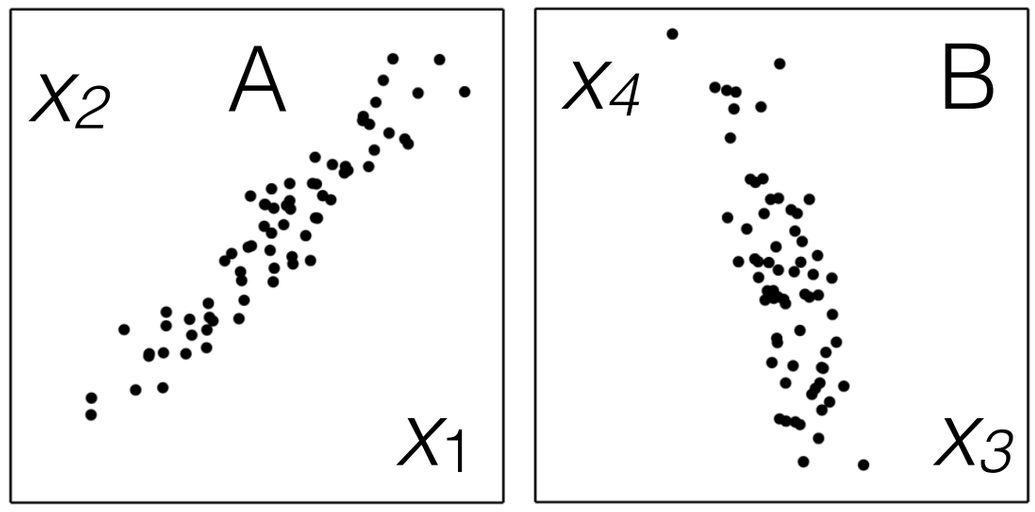
for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.
  - There are some true, unknown axes w_1, w_2, w_3 such that the ball and spring movement are very largely in w_1.
  - We set up three cameras, take occasional snapshots.

exploratory data analysis





[Adapted from Schlens 2014]

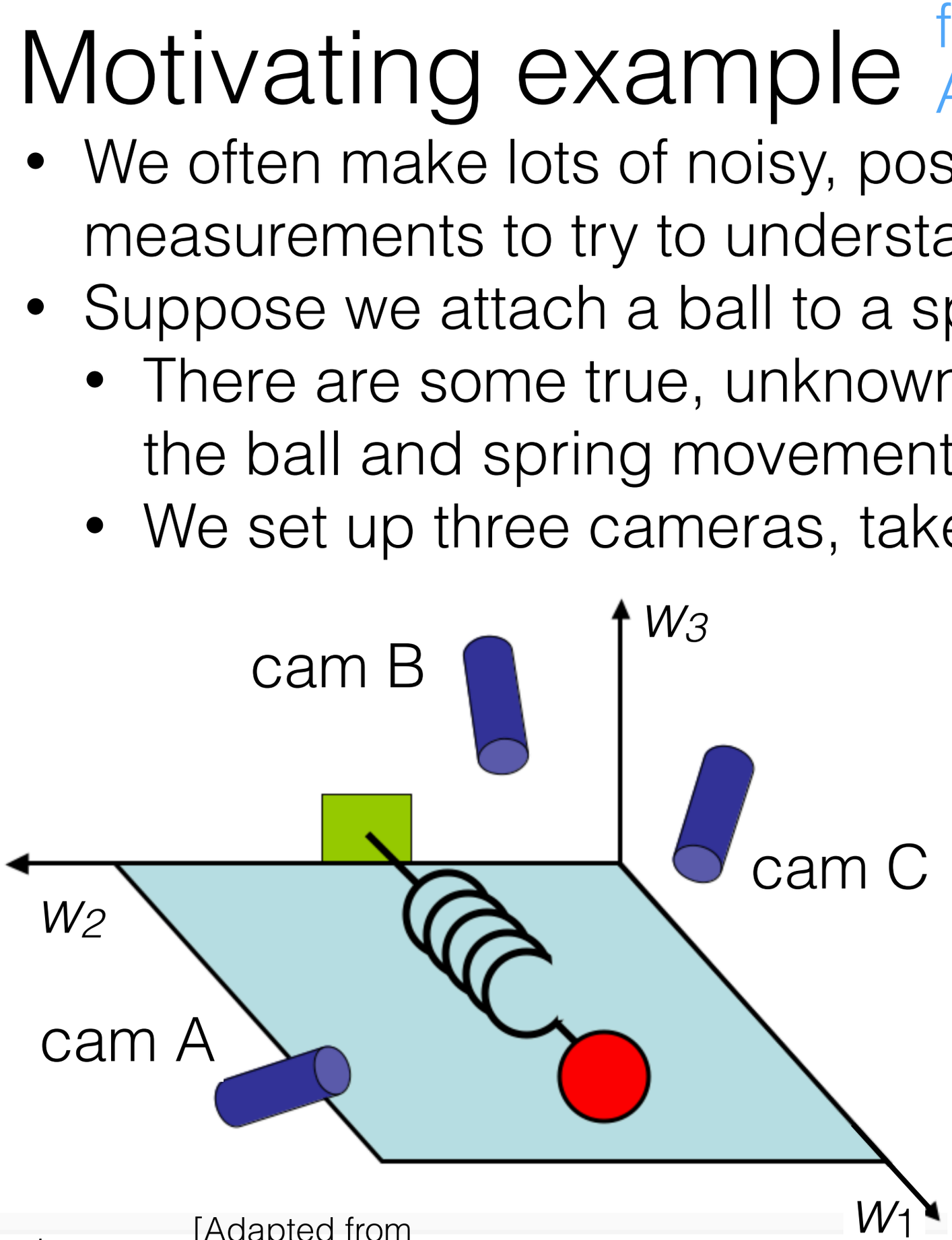
Motivating example

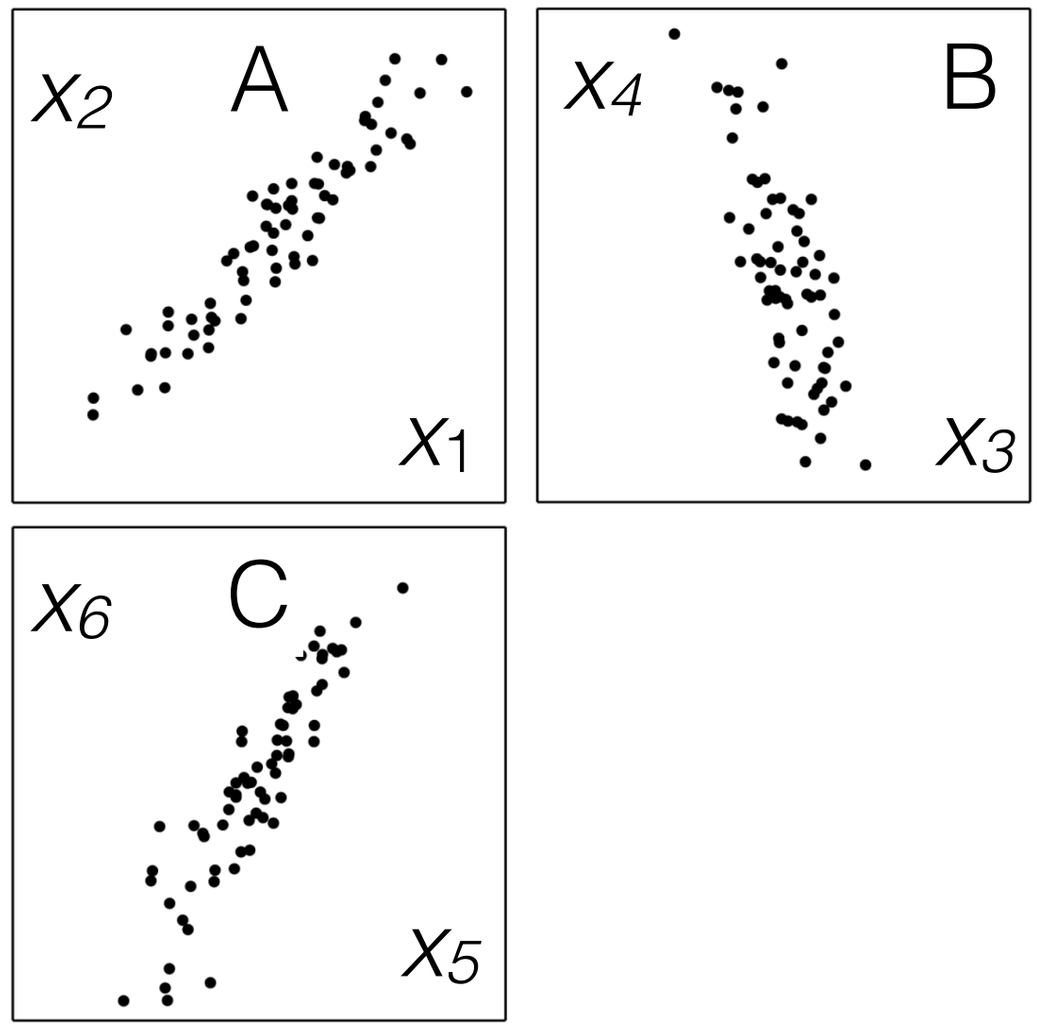
for Principal Components Analysis (PCA)

[Example thanks to Schlens 2014]

- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.
  - There are some true, unknown axes w_1, w_2, w_3 such that the ball and spring movement are very largely in w_1.
  - We set up three cameras, take occasional snapshots.

exploratory data analysis





[Adapted from Schlens 2014]

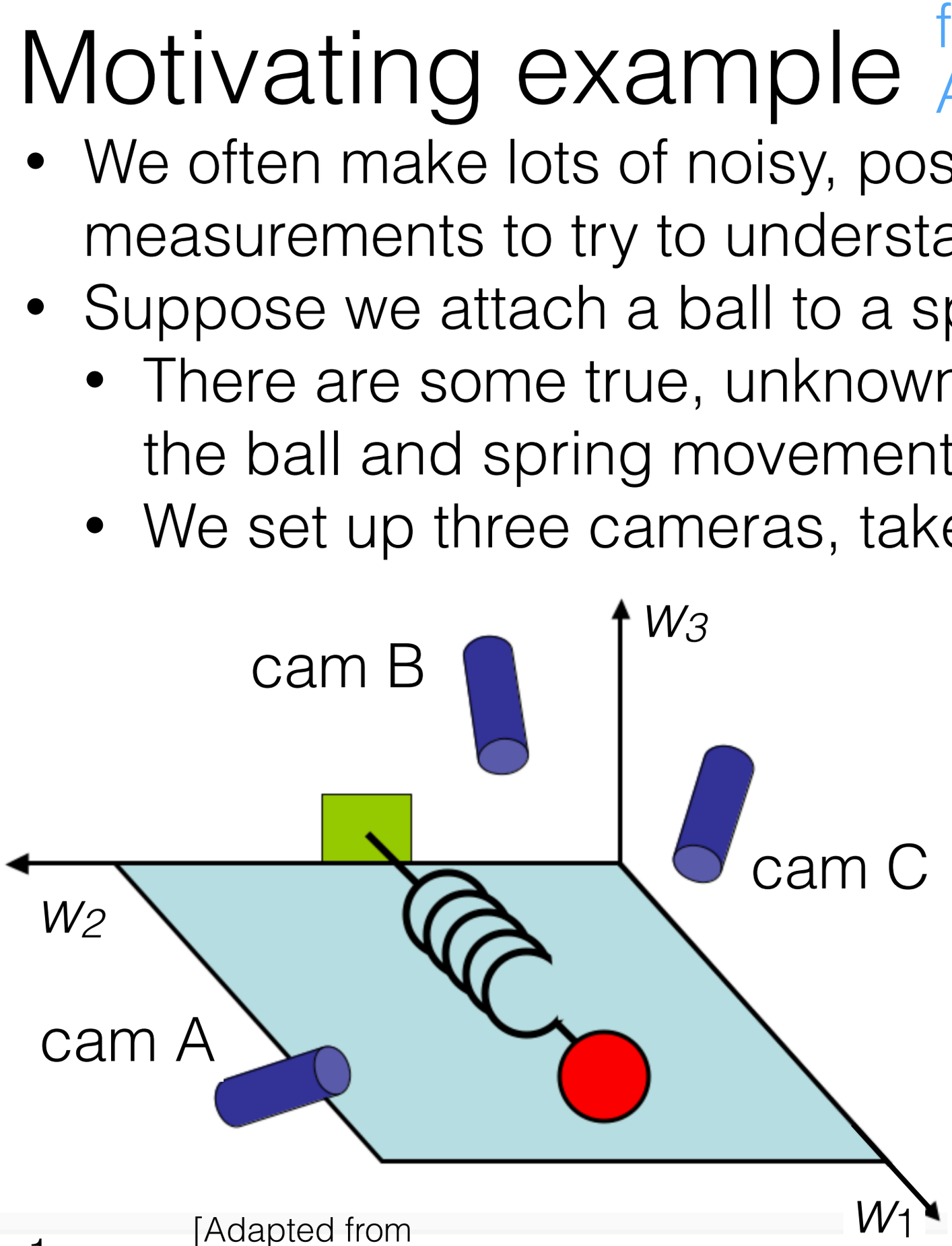
Motivating example

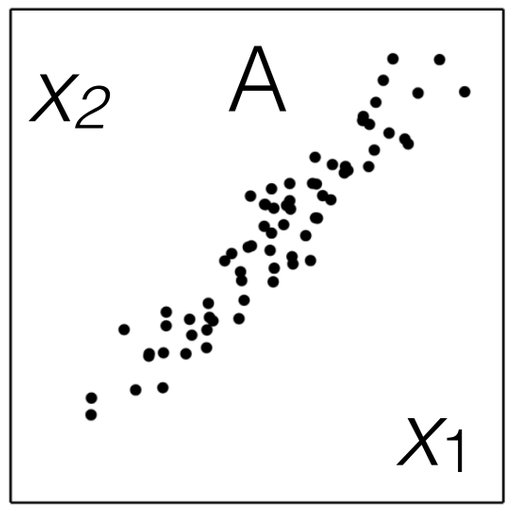
for Principal Components Analysis (PCA)

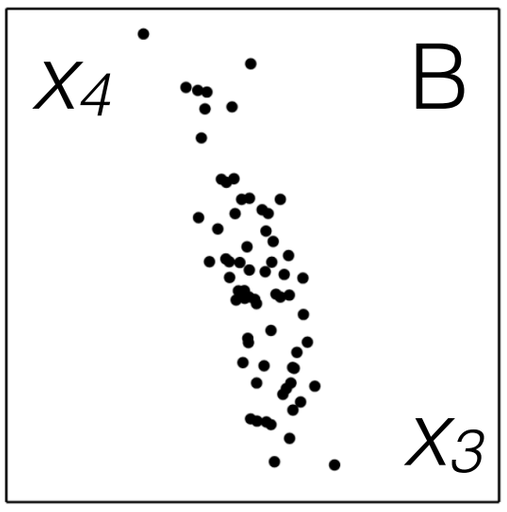
[Example thanks to Schlens 2014]

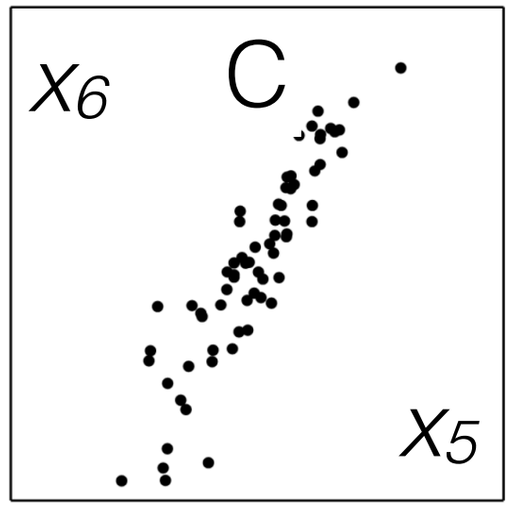
- We often make lots of noisy, possibly-redundant measurements to try to understand some phenomenon
- Suppose we attach a ball to a spring.
  - There are some true, unknown axes w_1, w_2, w_3 such that the ball and spring movement are very largely in w_1.
  - We set up three cameras, take occasional snapshots.

exploratory data analysis





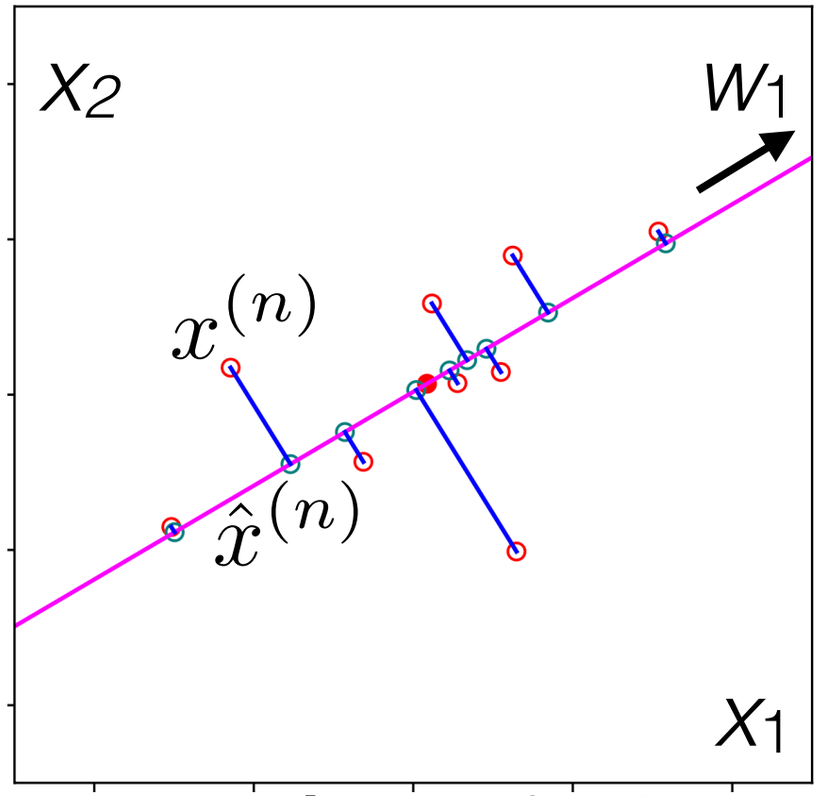




- We have 6 features, but need only 1

[Adapted from Schlens 2014]

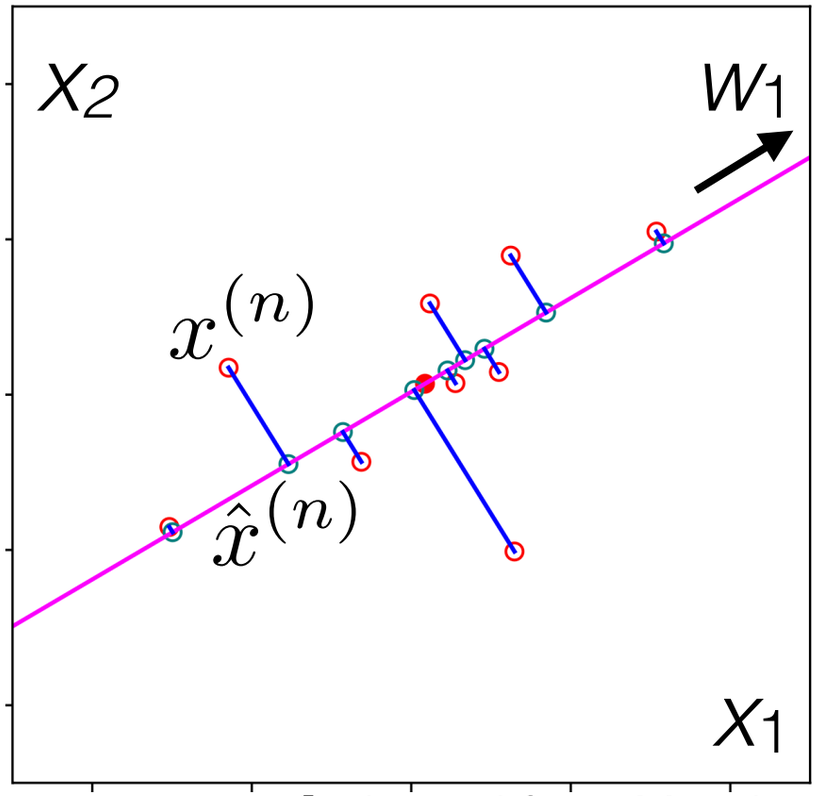
Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$

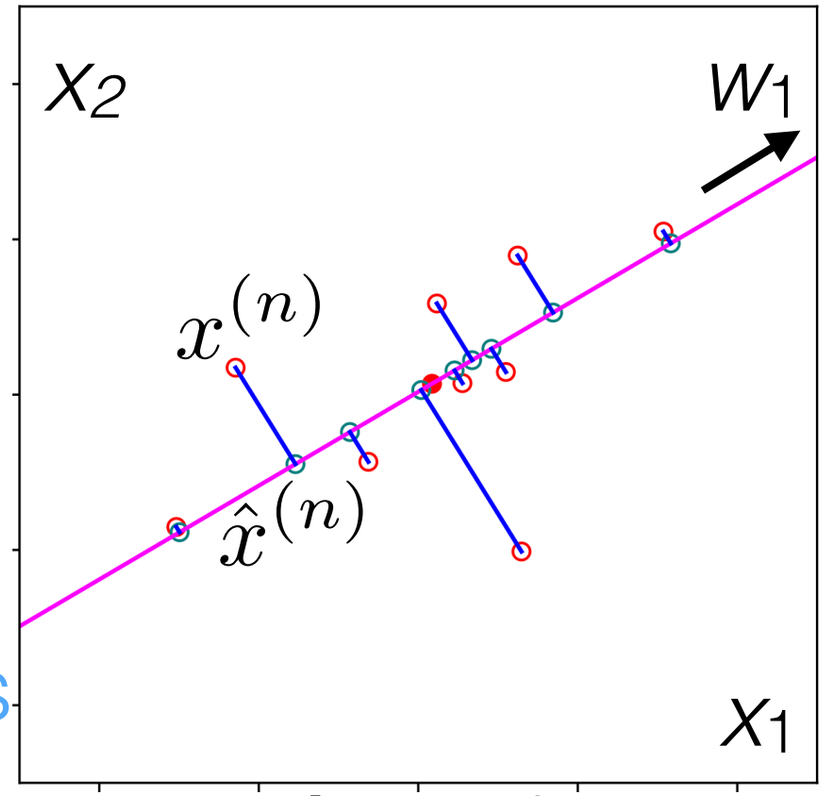


[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components



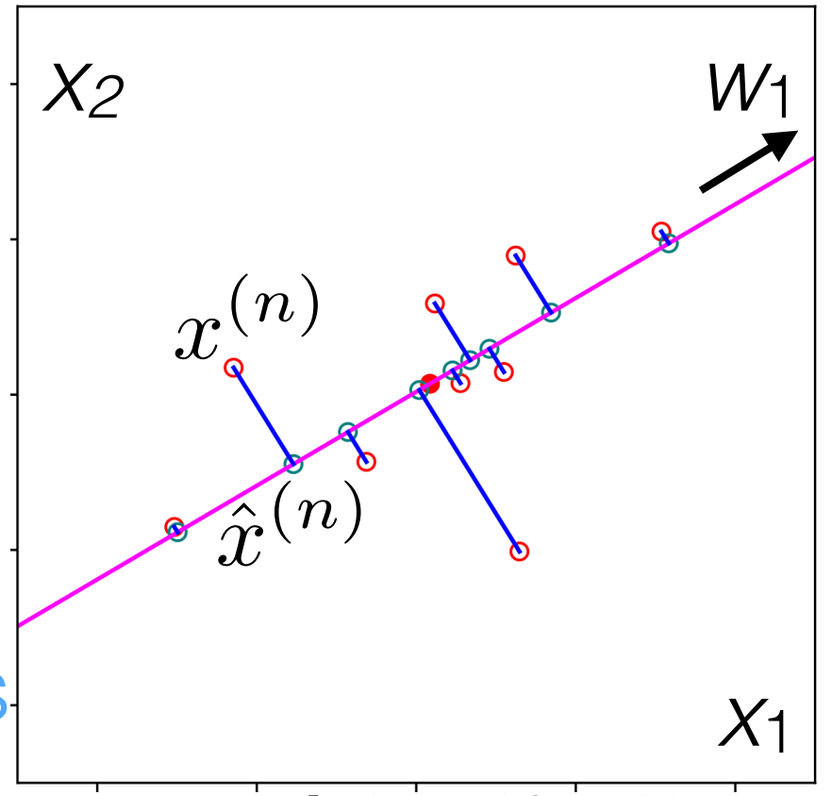
[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components



[adapted from Murphy 2022, Fig 20.1]

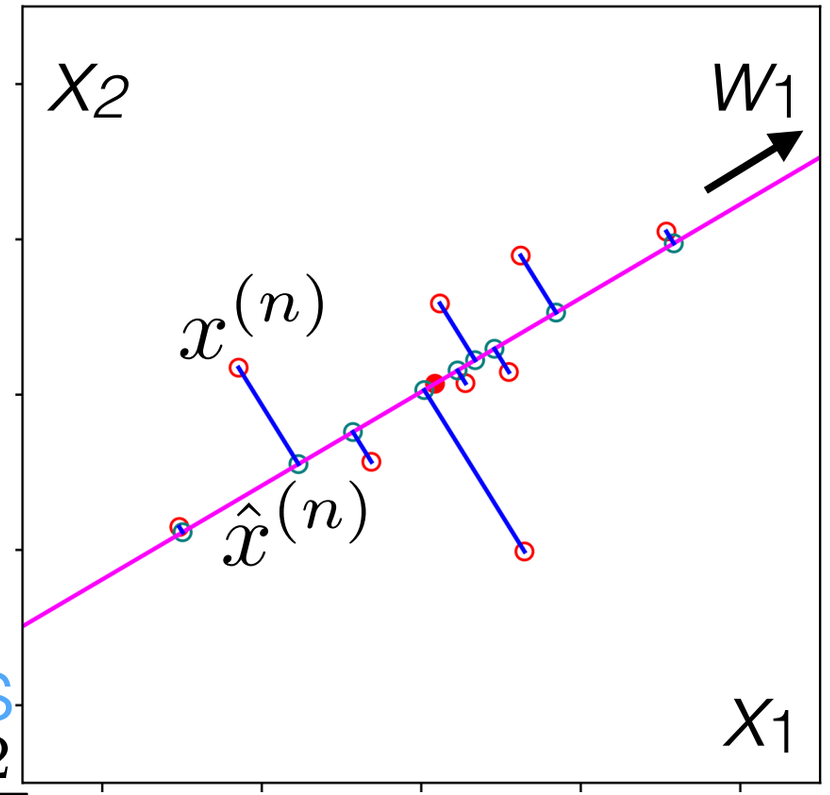
Problem setup for PCA

- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

principal components



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

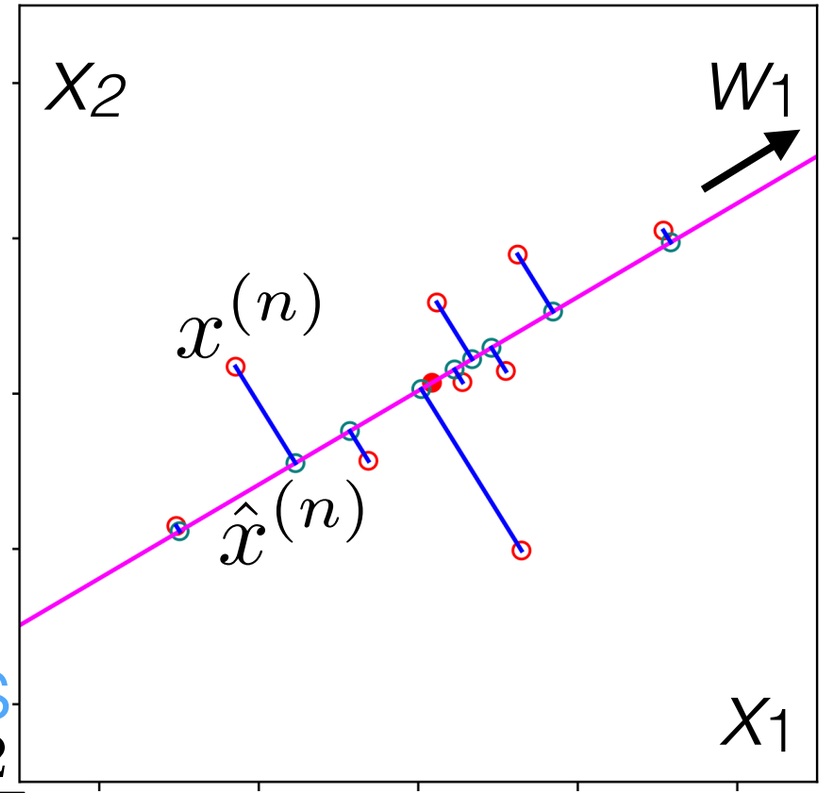
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over W (DxL) and Z (NxL)
  - constraint: W represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

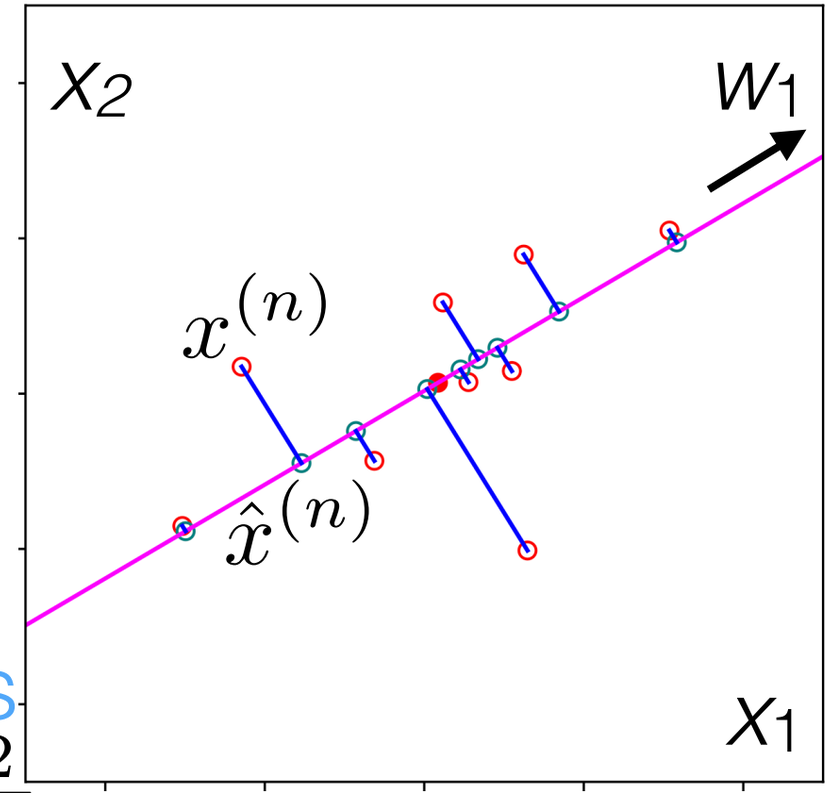
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis
- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

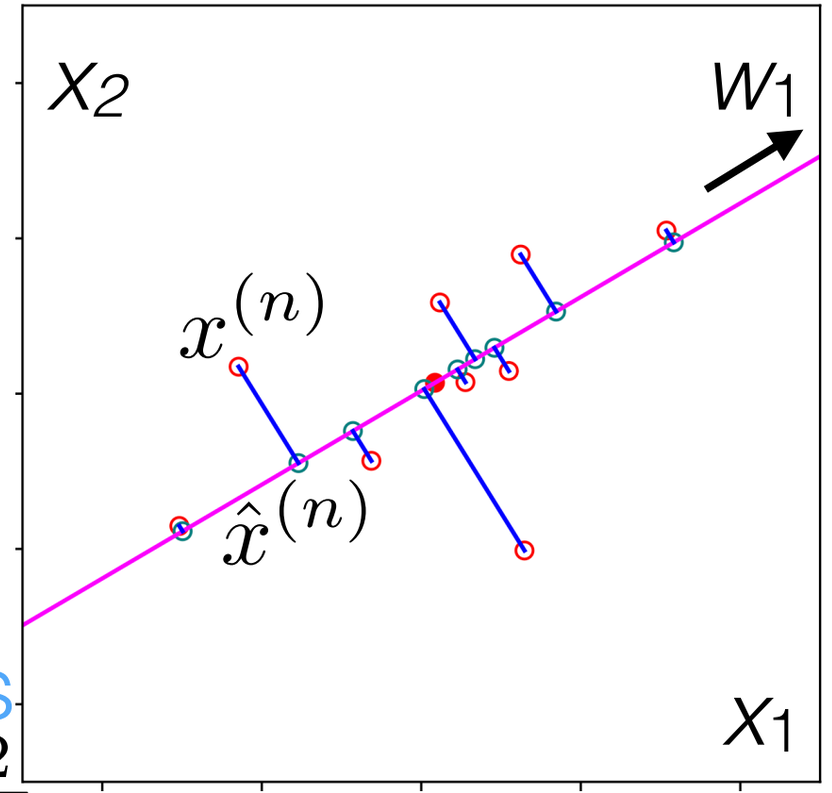
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data

Problem setup for PCA

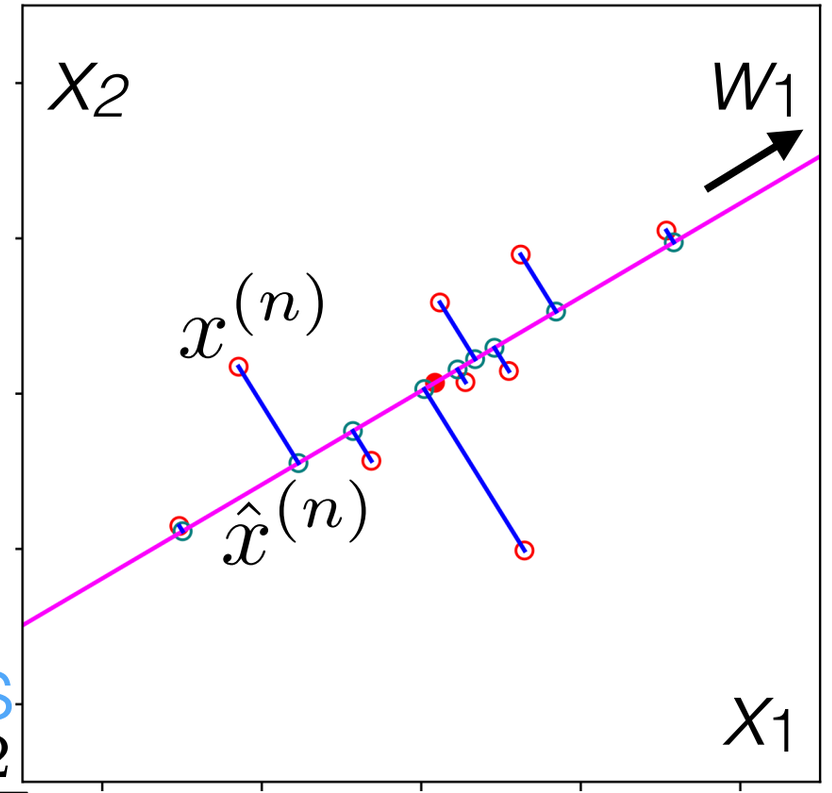
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:

Problem setup for PCA

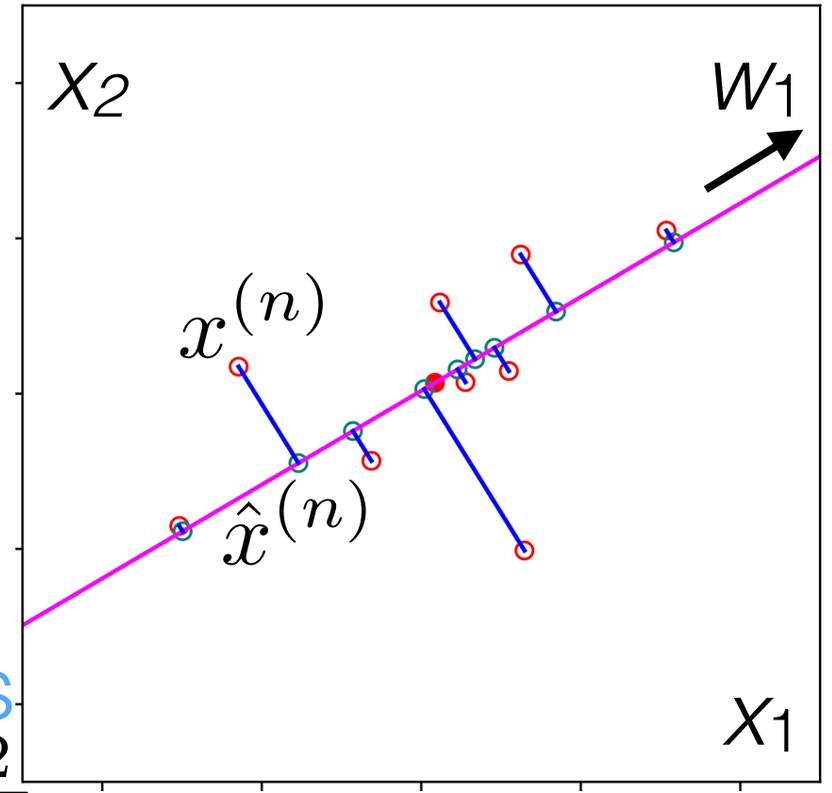
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data: $\sum_{n=1}^{N} z_1^{(n)}$

Problem setup for PCA

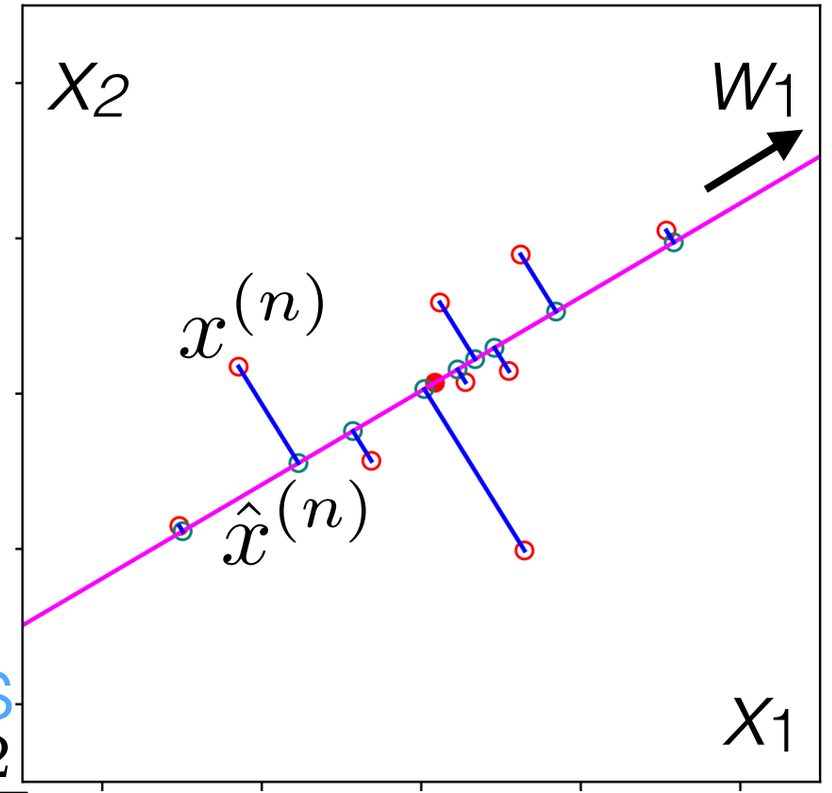
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:

$$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)}$$

Problem setup for PCA

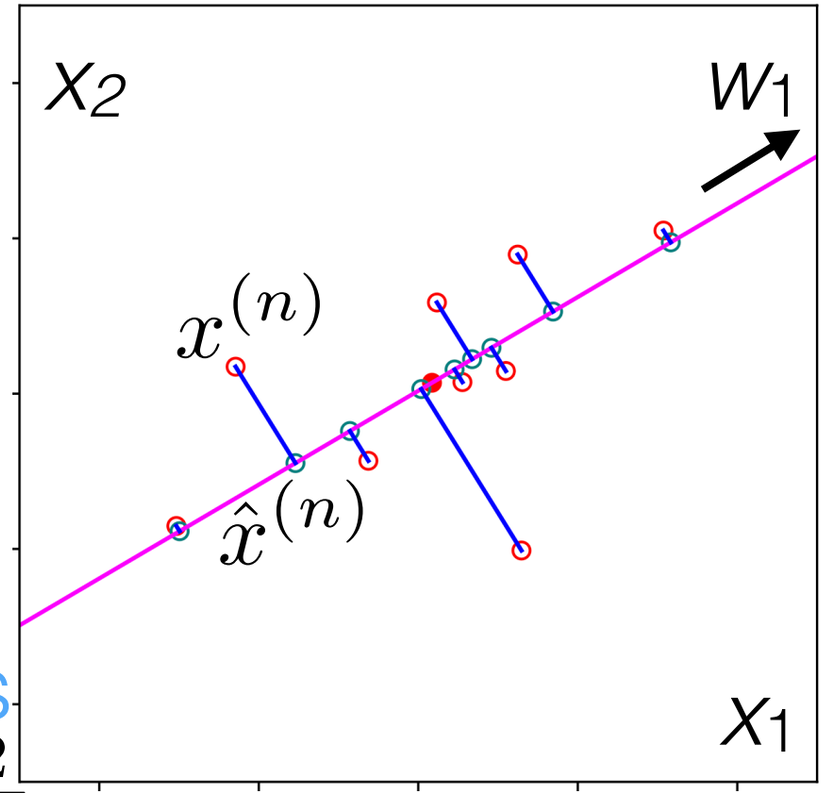
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:

$$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)}$$

Problem setup for PCA

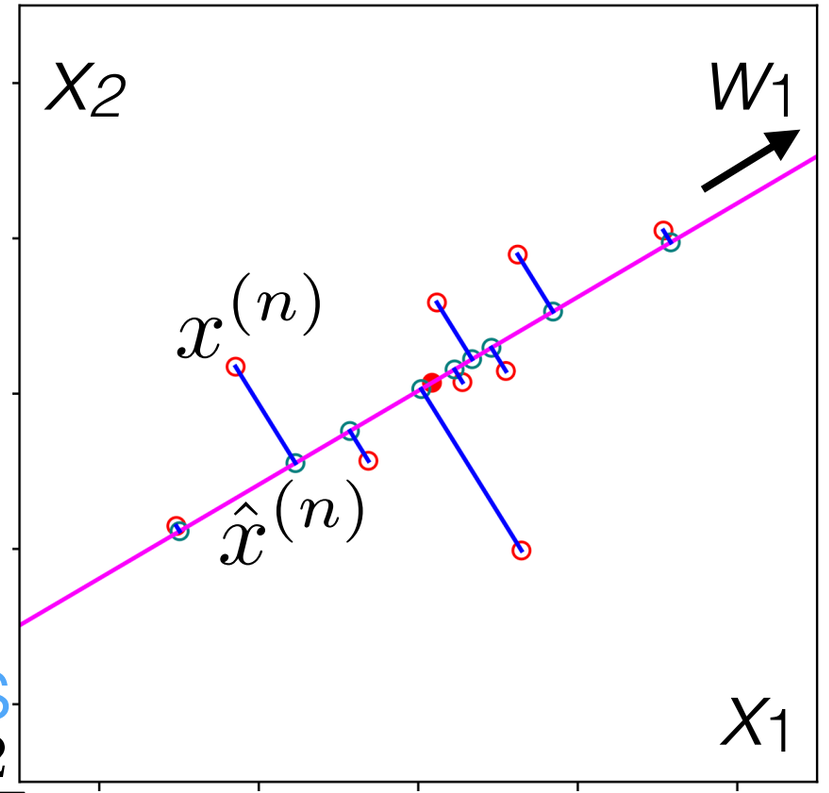
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:

$$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} =$$

Problem setup for PCA

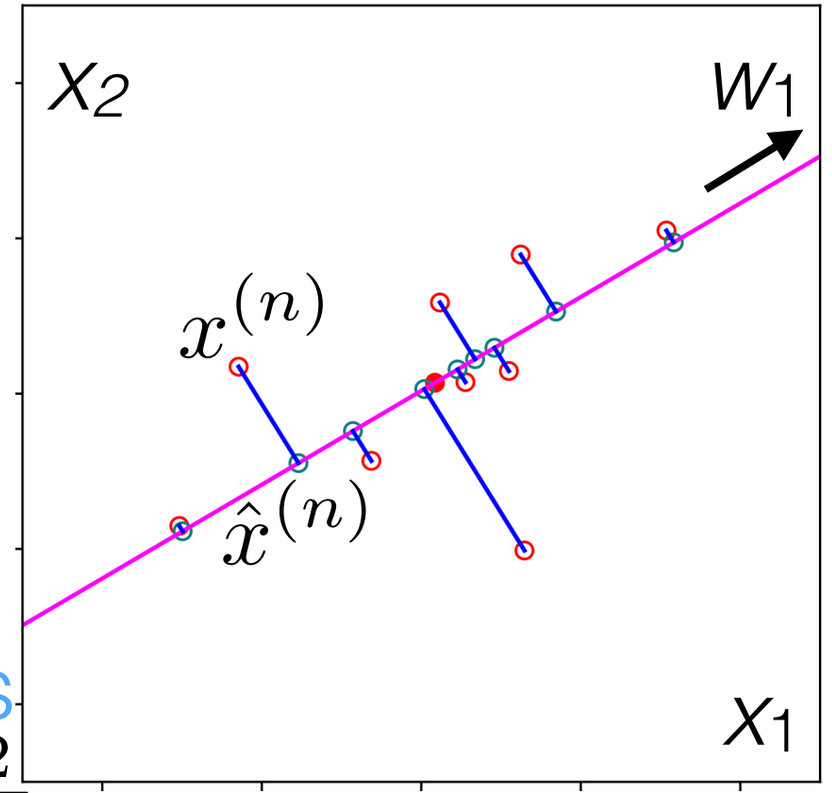
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:

$$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$

Problem setup for PCA

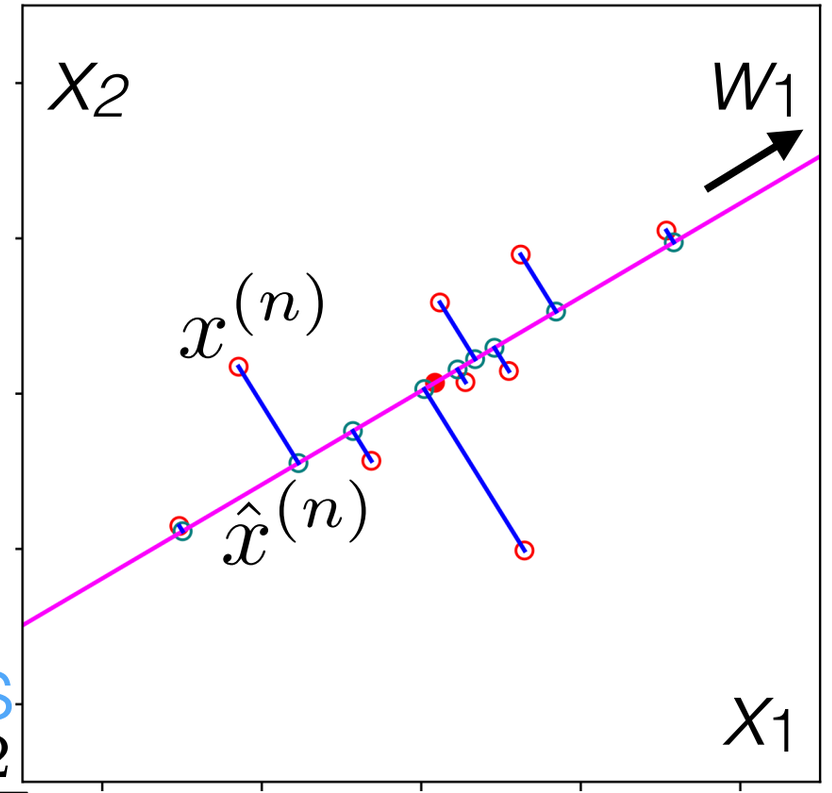
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:

Problem setup for PCA

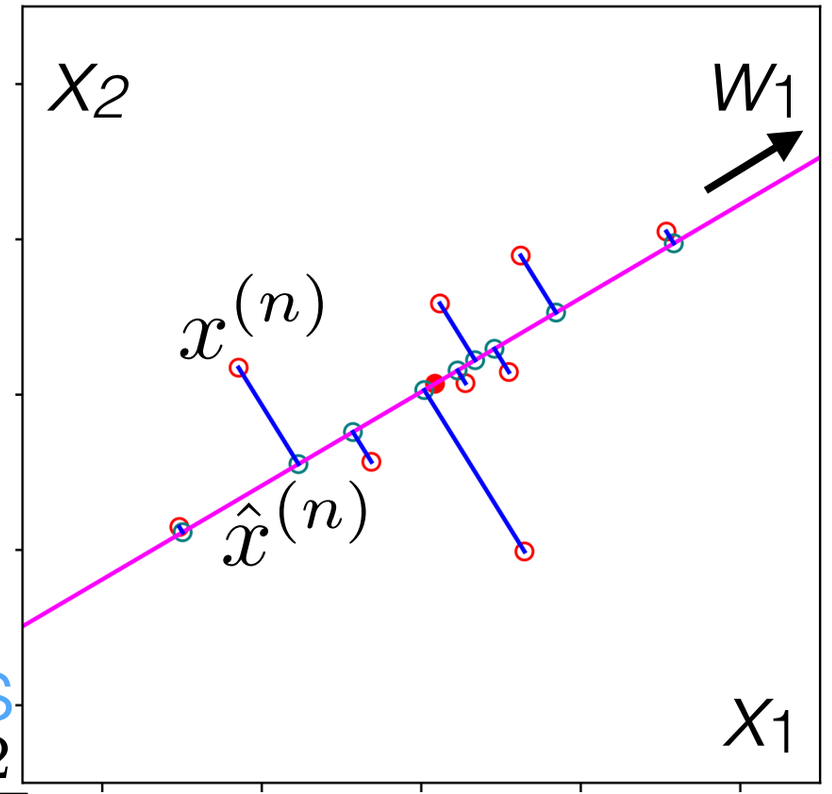
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:
    $$\sum_{n=1}^{N} (z_1^{(n)})^2$$

Problem setup for PCA

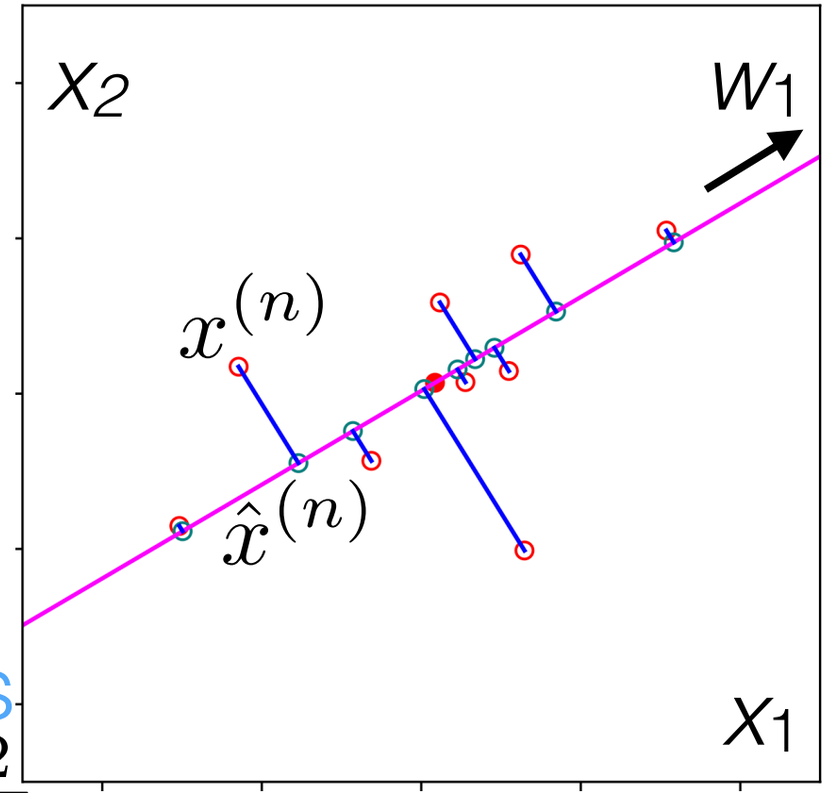
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:
    $$\sum_{n=1}^{N} (z_1^{(n)})^2 = \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$$

Problem setup for PCA

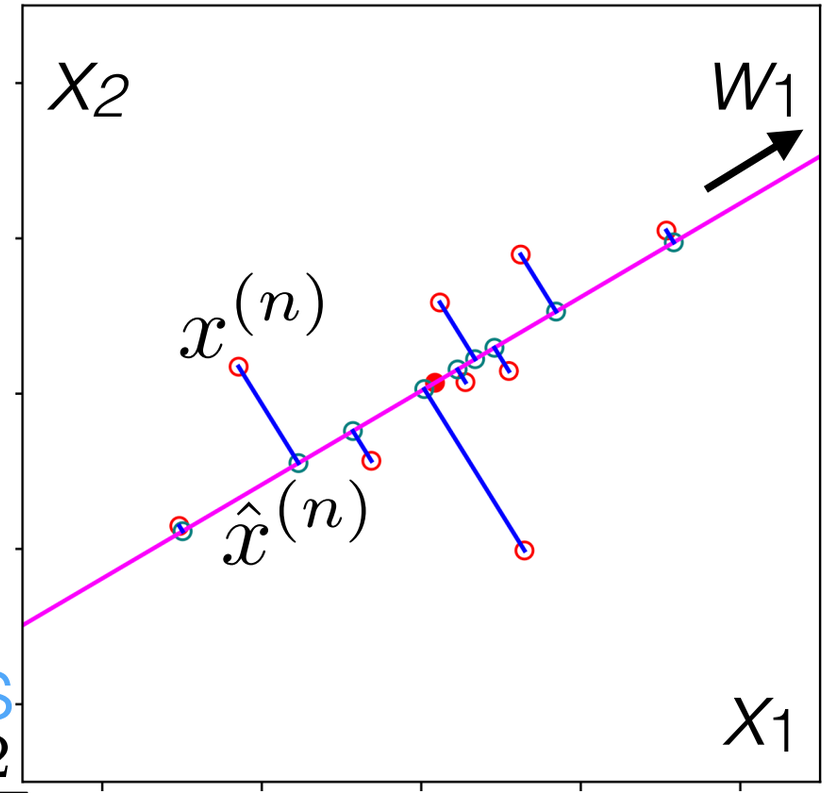
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[adapted from Murphy 2022, Fig 20.1]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:
    $$\sum_{n=1}^{N} (z_1^{(n)})^2 = \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$$ (-1 * the objective)

Problem setup for PCA

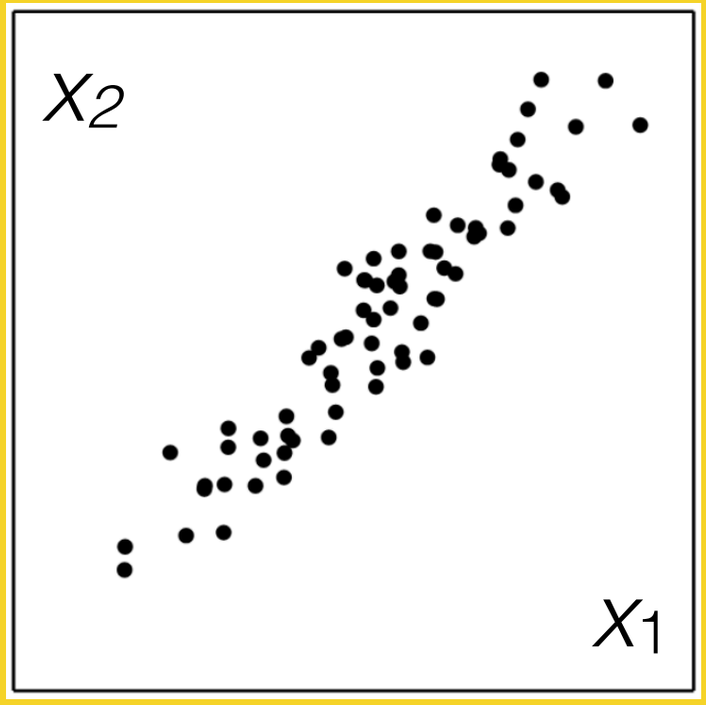
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[Adapted from Schlens 2014]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:
    $$\sum_{n=1}^{N} (z_1^{(n)})^2 = \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$$ (-1 * the objective)

Problem setup for PCA

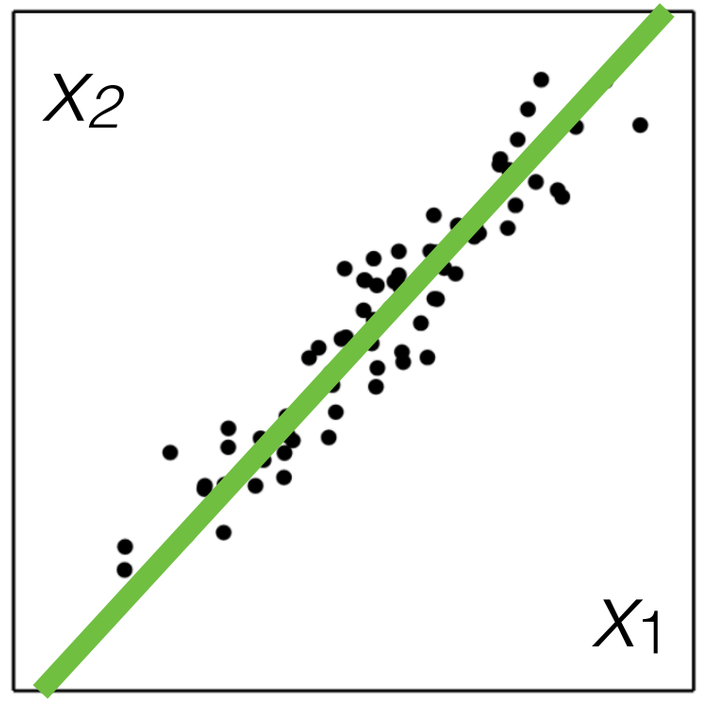
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[Adapted from Schlens 2014]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:
    $$\sum_{n=1}^{N} (z_1^{(n)})^2 = \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$$ (-1 * the objective)

Problem setup for PCA

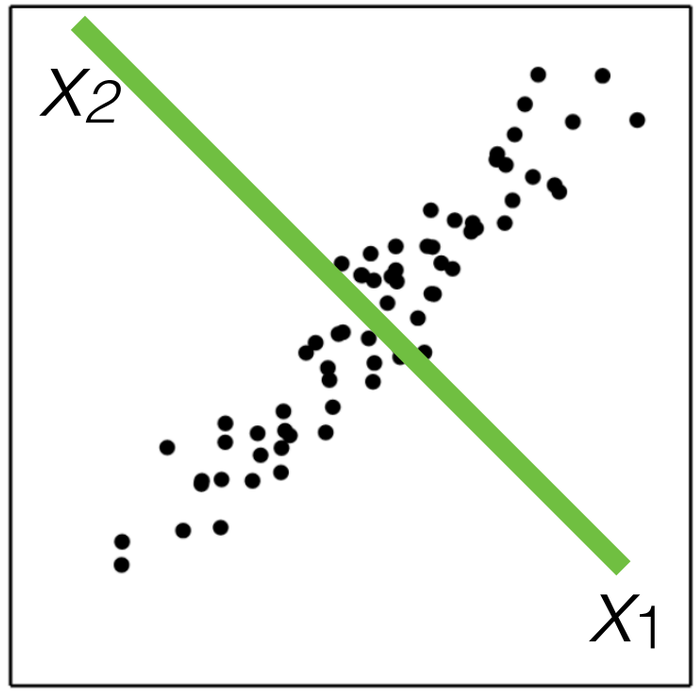
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[Adapted from Schlens 2014]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:
    $$\sum_{n=1}^{N} (z_1^{(n)})^2 = \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$$ (-1 * the objective)

Problem setup for PCA

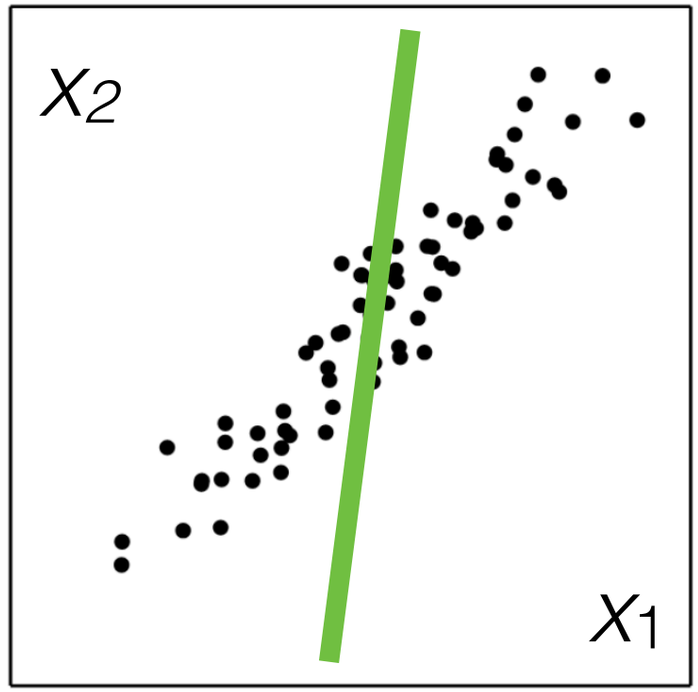
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[Adapted from Schlens 2014]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:
    $$\sum_{n=1}^{N} (z_1^{(n)})^2 = \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$$ (-1 * the objective)

Problem setup for PCA

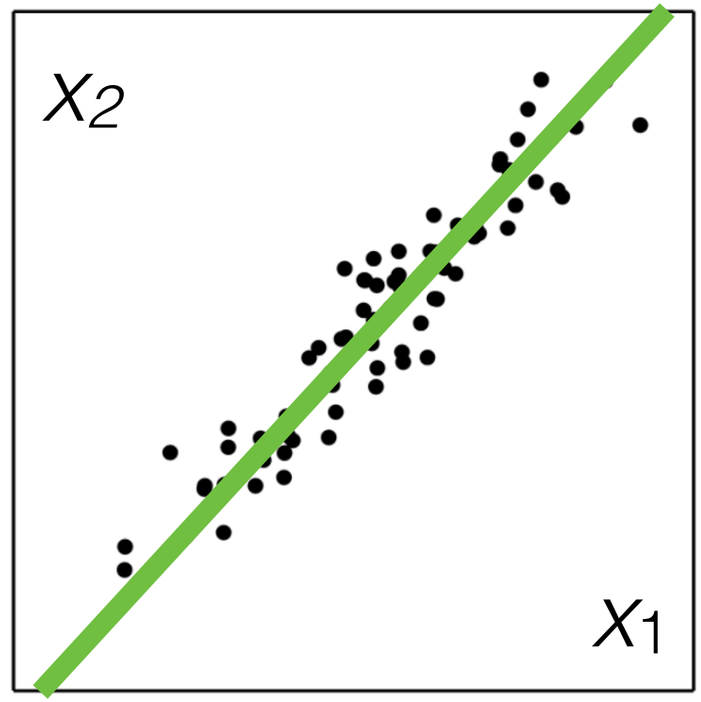
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Goal: approx the data with its projection onto a low-dim subspace, with orthonormal basis $w_1, \ldots, w_L$

$$x^{(n)} \approx \sum_{\ell=1}^{L} z_\ell^{(n)} w_\ell =: \hat{x}^{(n)}$$

principal components

$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - WZ^\top\|_F^2$$

  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis



[Adapted from Schlens 2014]

- *L*=1: we found $z_1^{(n)} = w_1^\top x^{(n)}$ & goal: $\min - \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$
- Another perspective on PCA: at each step, it finds the direction that maximizes variance of the projected data
  - *L*=1: Empirical mean of the projected data:
    $$\sum_{n=1}^{N} z_1^{(n)} = \sum_{n=1}^{N} w_1^\top x^{(n)} = w_1^\top \sum_{n=1}^{N} x^{(n)} = 0$$
  - *L*=1: Empirical variance of the projected data:
    $$\sum_{n=1}^{N} (z_1^{(n)})^2 = \sum_{n=1}^{N} (w_1^\top x^{(n)})^2$$ (-1 * the objective)

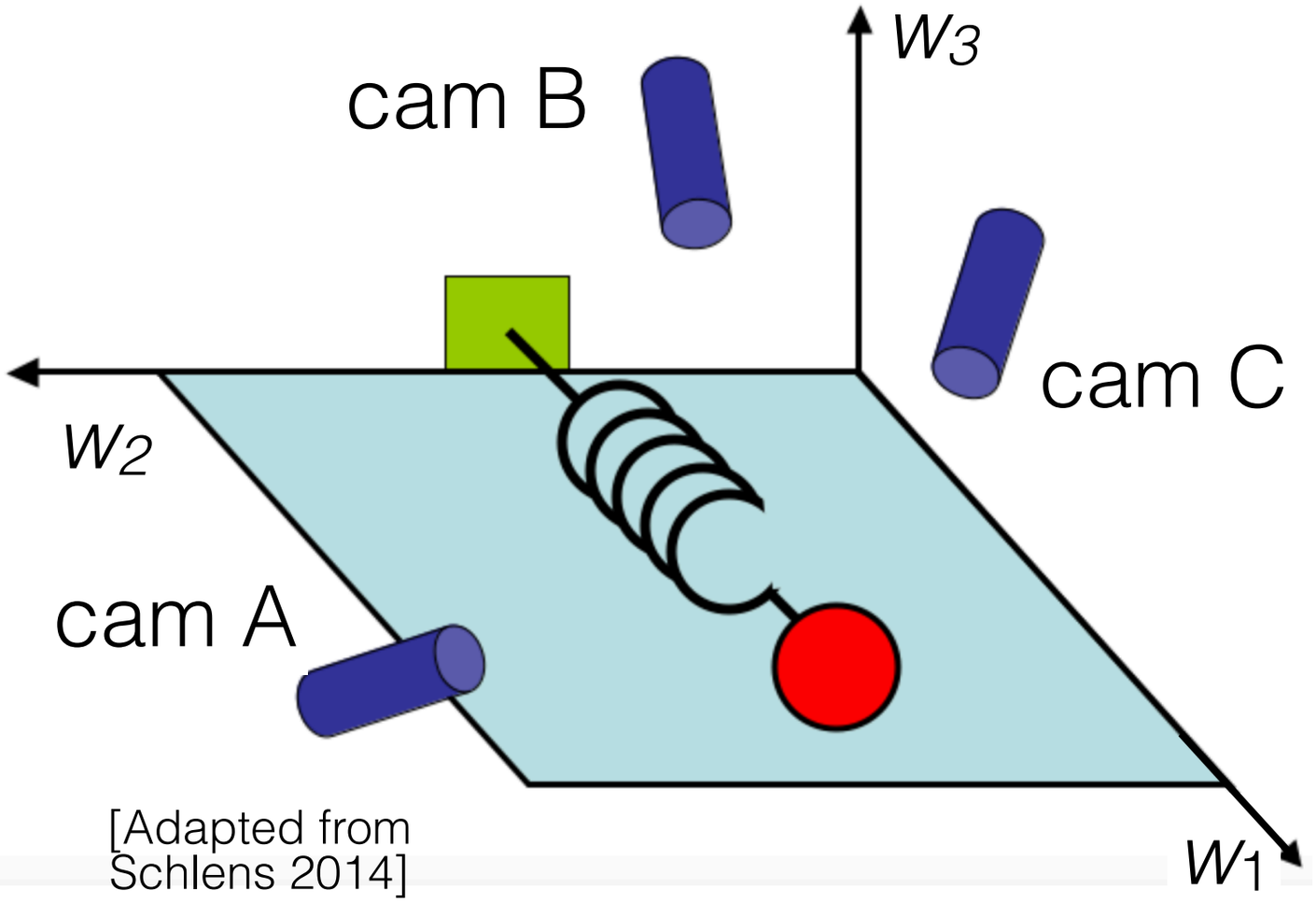
PCA Summary and Benefits

PCA Summary and Benefits

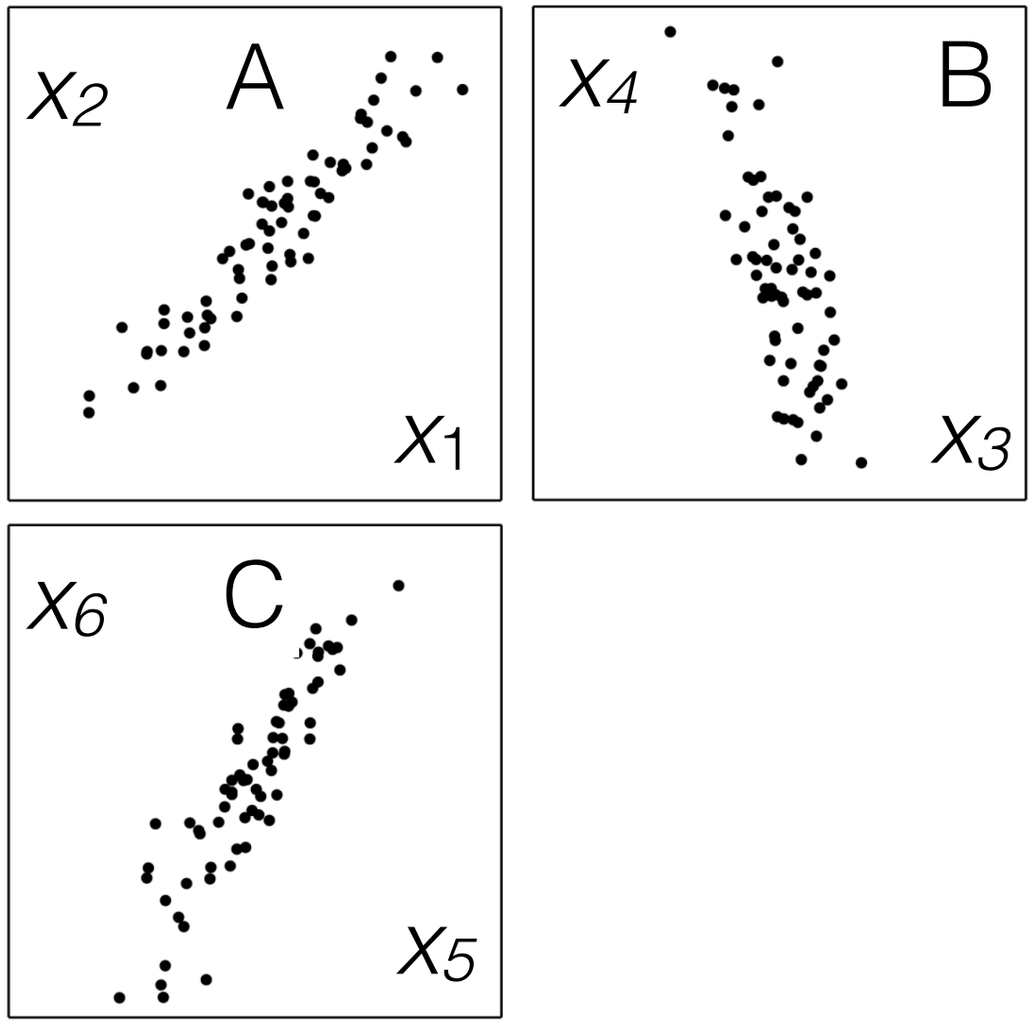
- In general, PCA returns the *L* eigenvectors of $\hat{\Sigma}$ corresponding to the *L* largest eigenvalues

PCA Summary and Benefits

- In general, PCA returns the *L* eigenvectors of $\hat{\Sigma}$ corresponding to the *L* largest eigenvalues

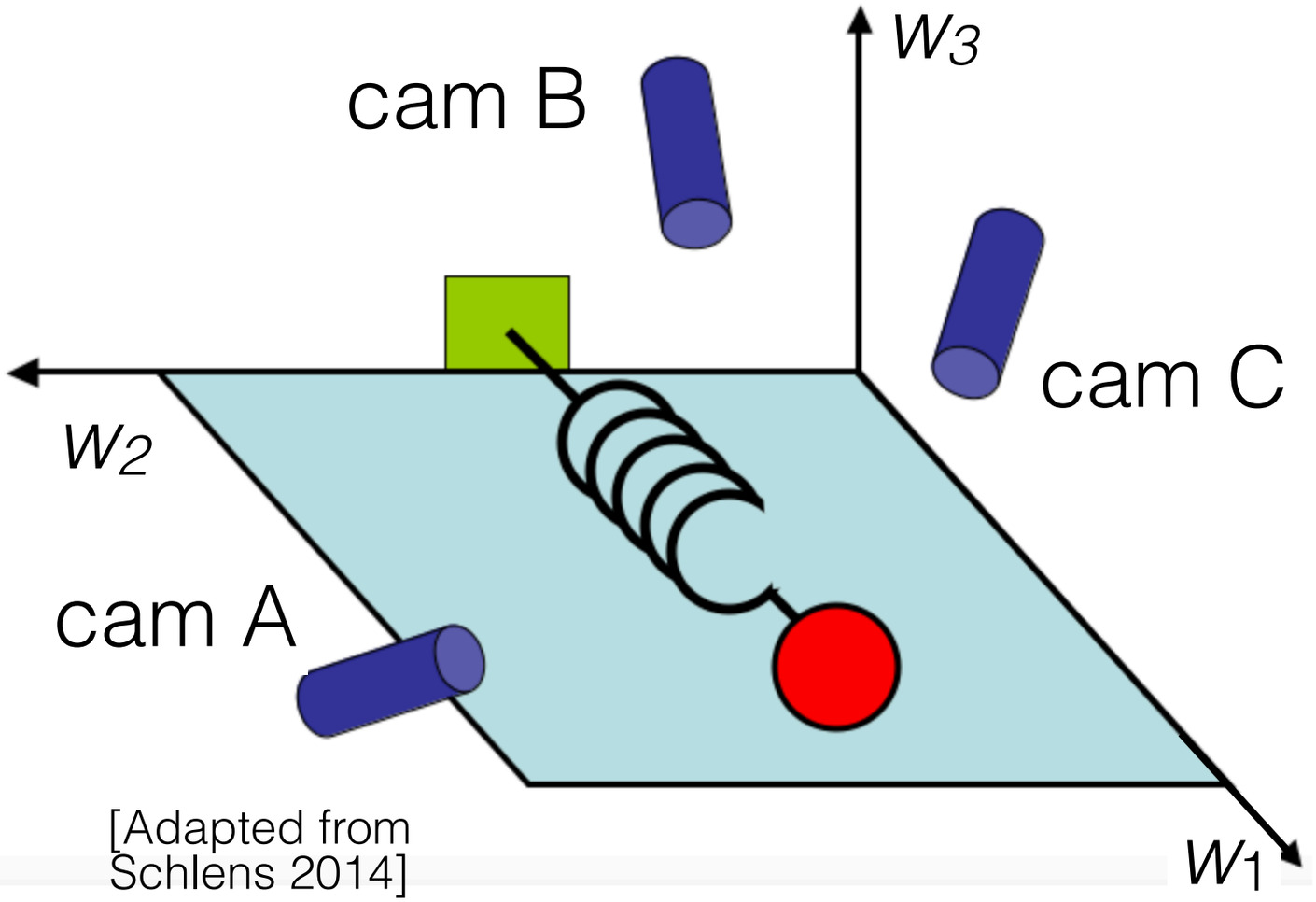


[Adapted from Schlens 2014]

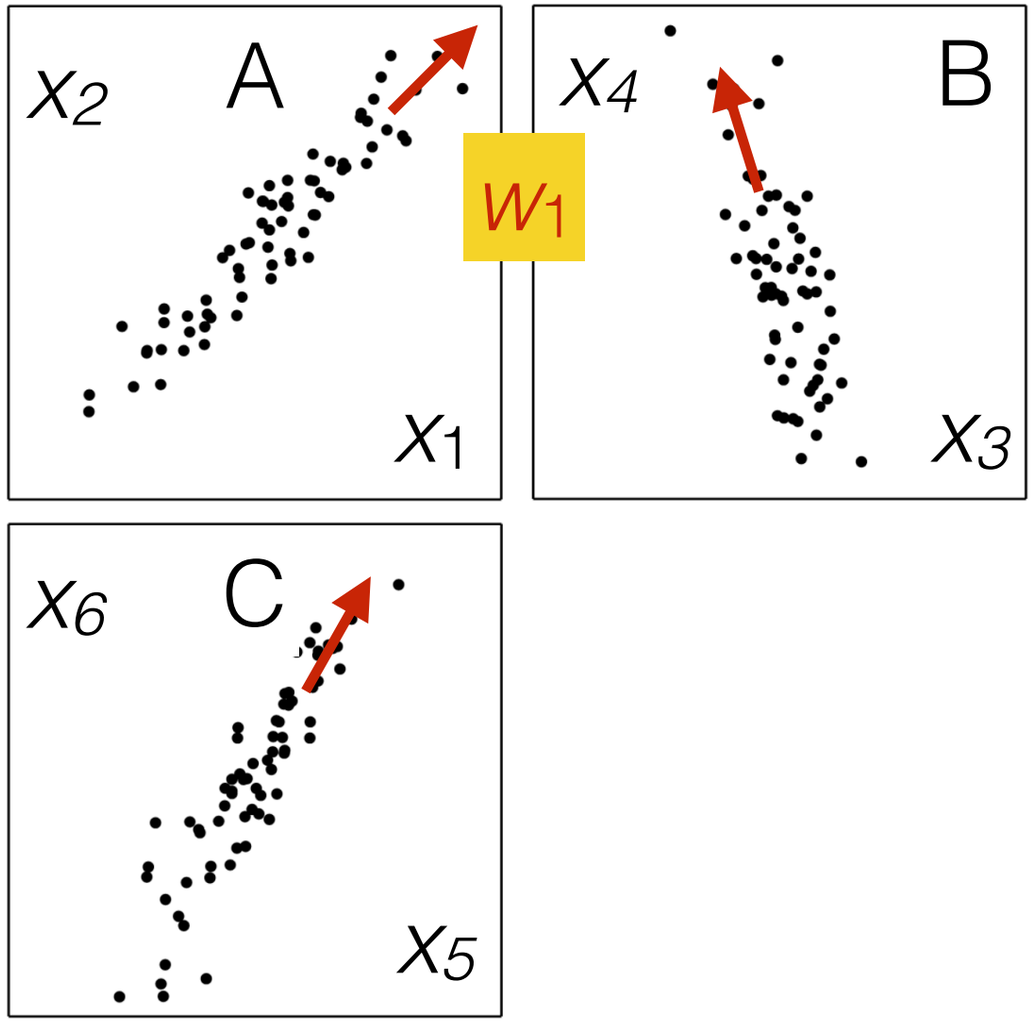


PCA Summary and Benefits

- In general, PCA returns the *L* eigenvectors of $\hat{\Sigma}$ corresponding to the *L* largest eigenvalues

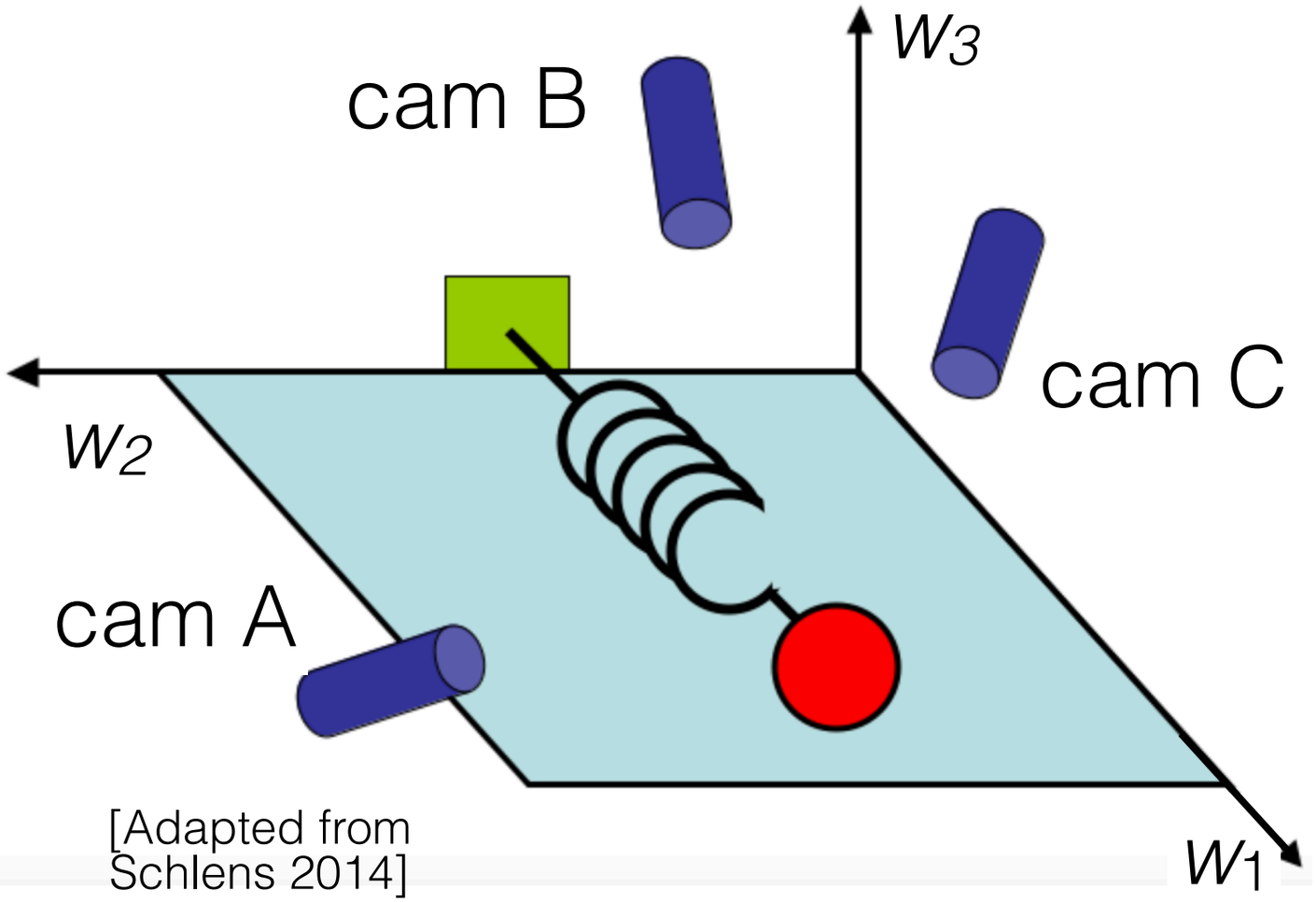


[Adapted from Schlens 2014]

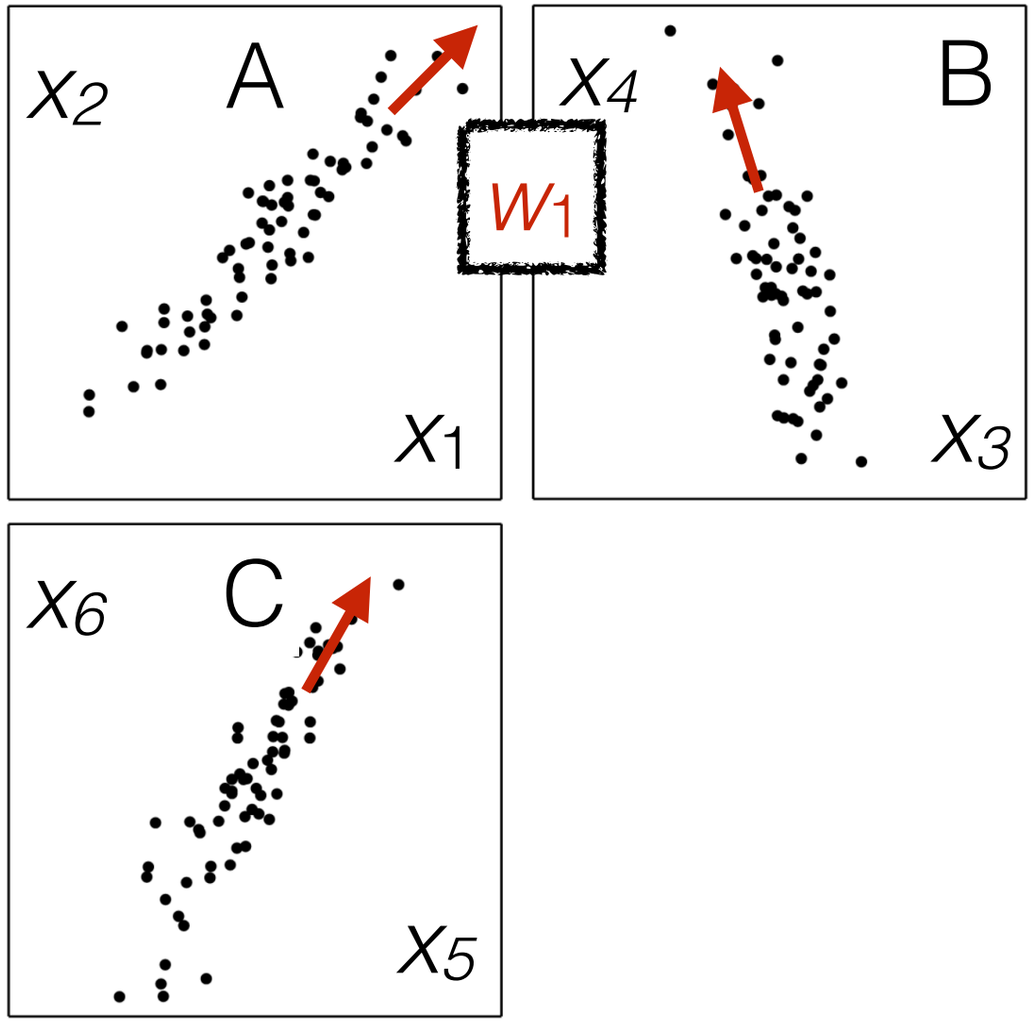


PCA Summary and Benefits

- In general, PCA returns the *L* eigenvectors of $\hat{\Sigma}$ corresponding to the *L* largest eigenvalues

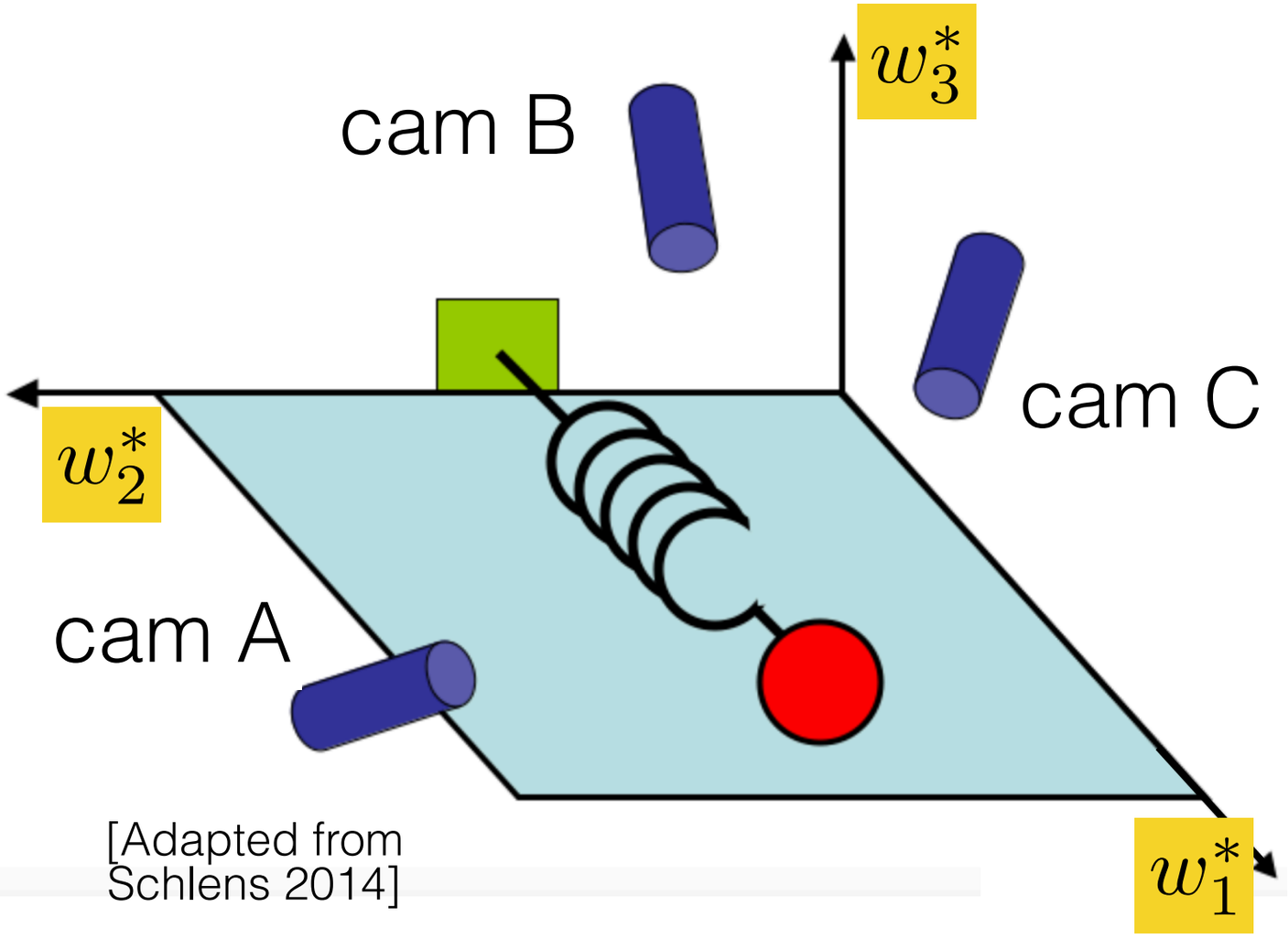


[Adapted from Schlens 2014]

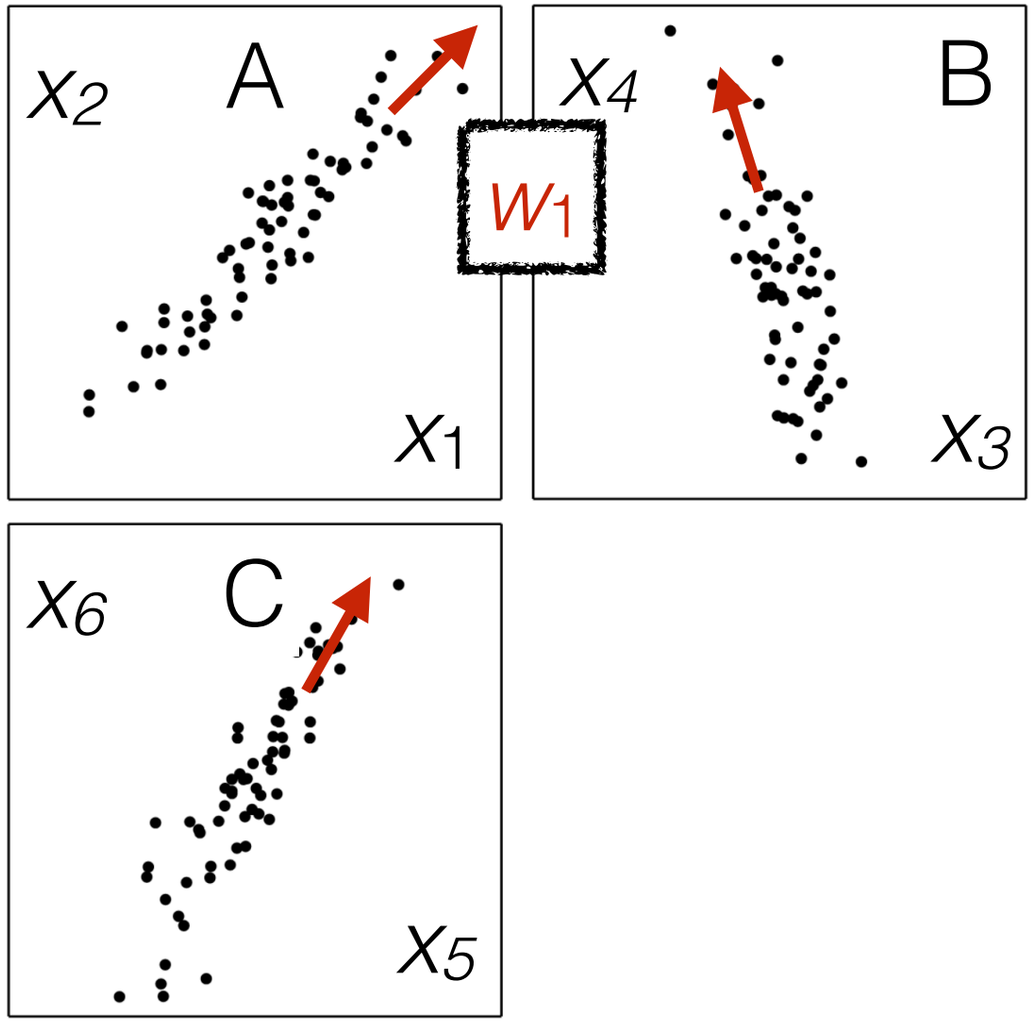


PCA Summary and Benefits

- In general, PCA returns the *L* eigenvectors of $\hat{\Sigma}$ corresponding to the *L* largest eigenvalues

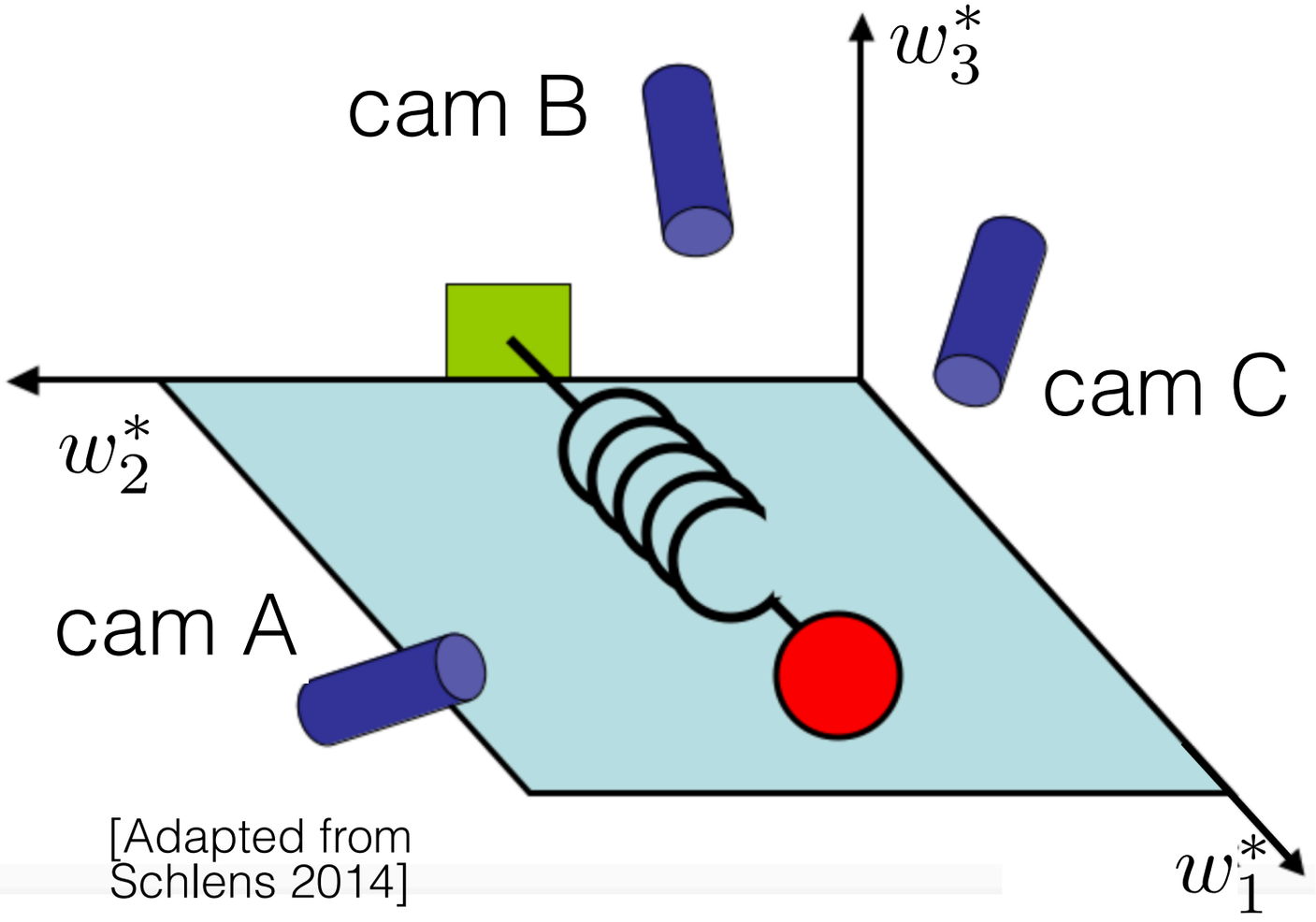


[Adapted from Schlens 2014]

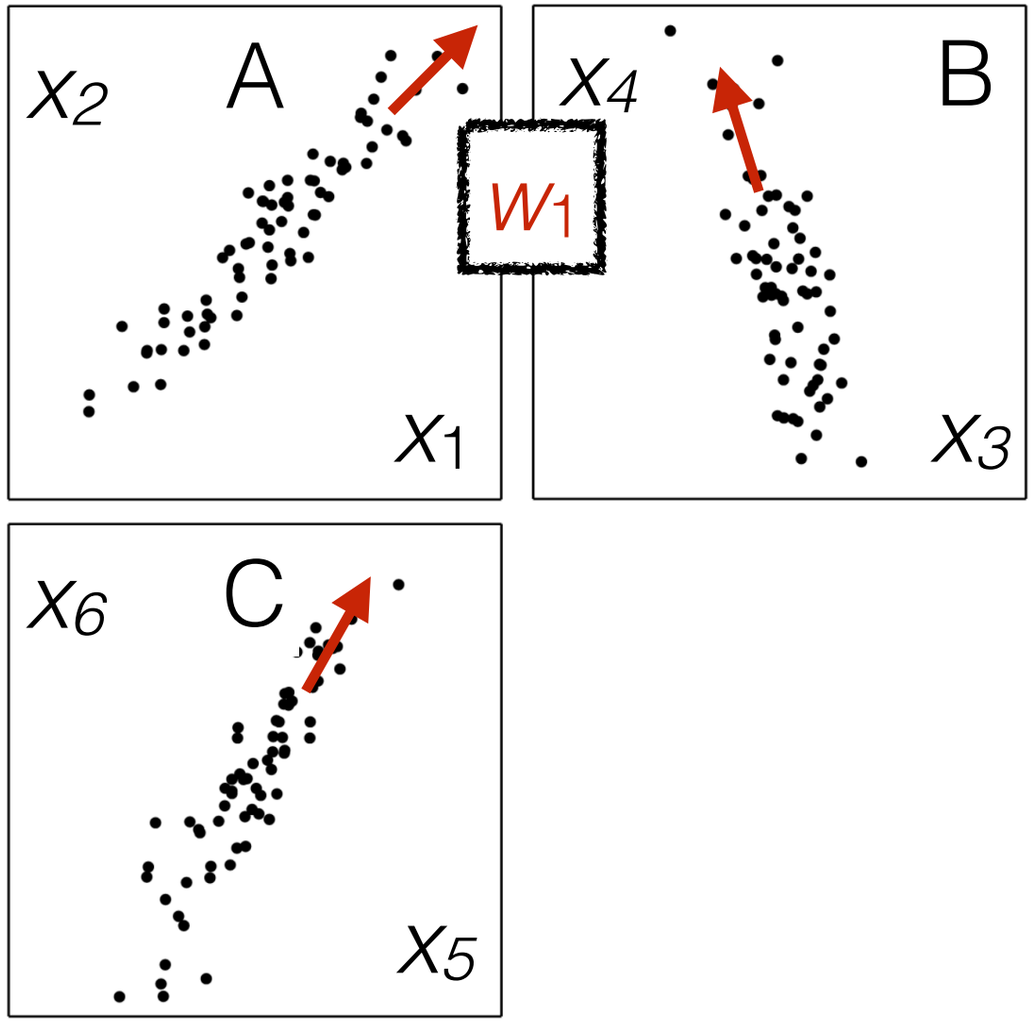


PCA Summary and Benefits

- In general, PCA returns the *L* eigenvectors of $\hat{\Sigma}$ corresponding to the *L* largest eigenvalues



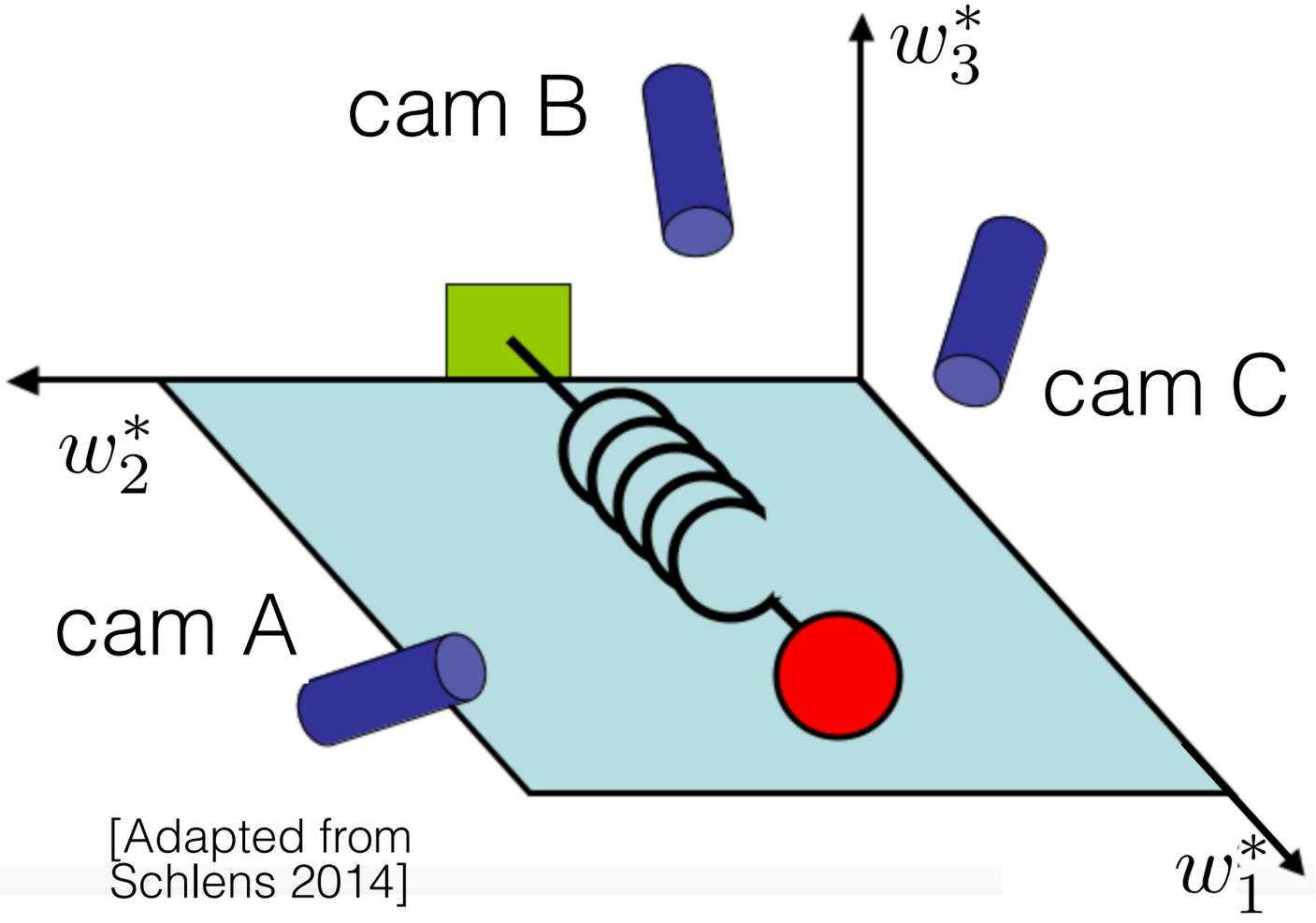
[Adapted from Schlens 2014]



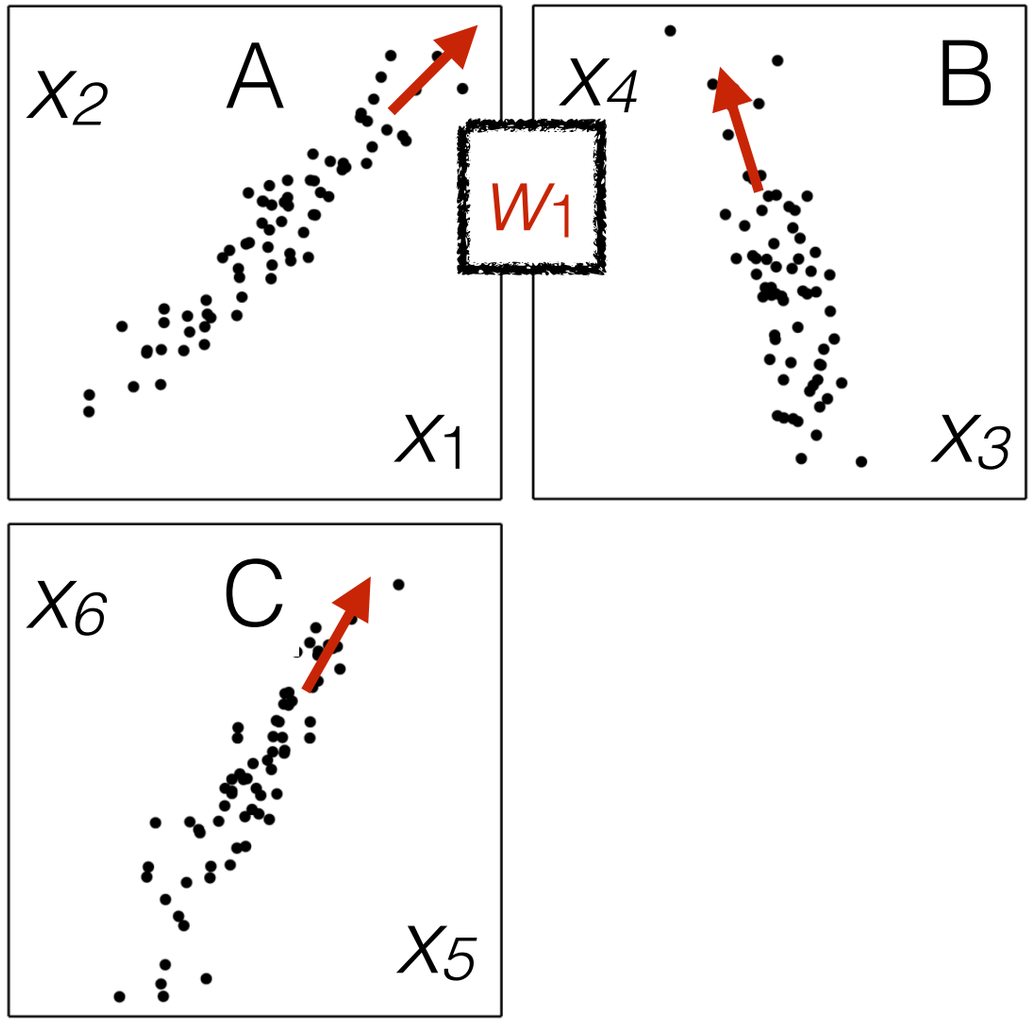
- In our motivating example, we expect PCA to estimate direction of interest well from the observed data

PCA Summary and Benefits

- In general, PCA returns the *L* eigenvectors of $\hat{\Sigma}$ corresponding to the *L* largest eigenvalues

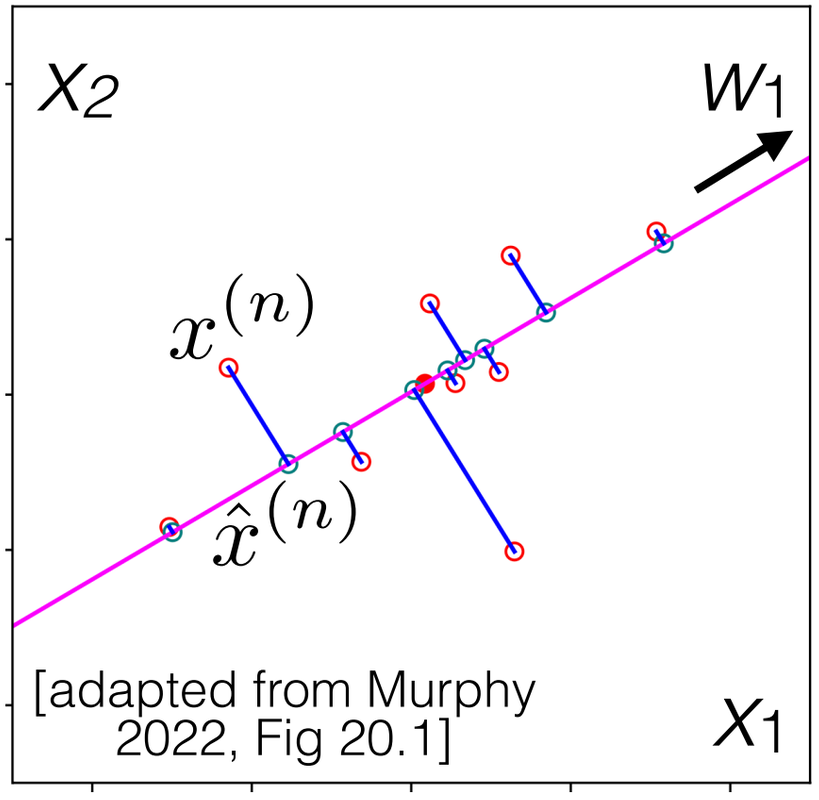


[Adapted from Schlens 2014]



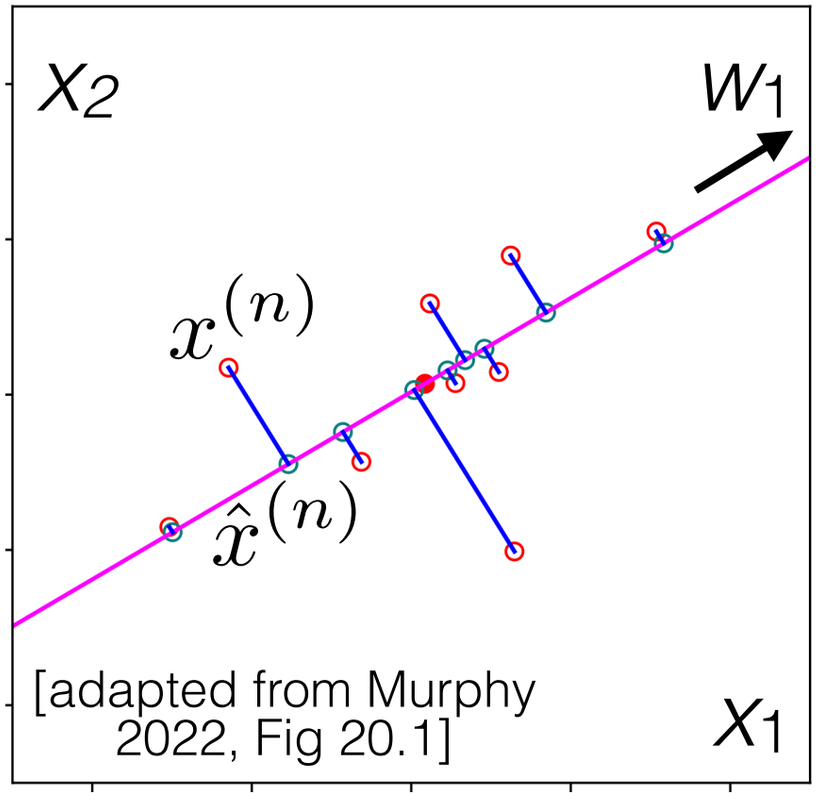
- In our motivating example, we expect PCA to estimate direction of interest well from the observed data
- When plotting, we expect PCA to find coordinates that most distribute the data out across the plot (max variance)

Challenges of PCA



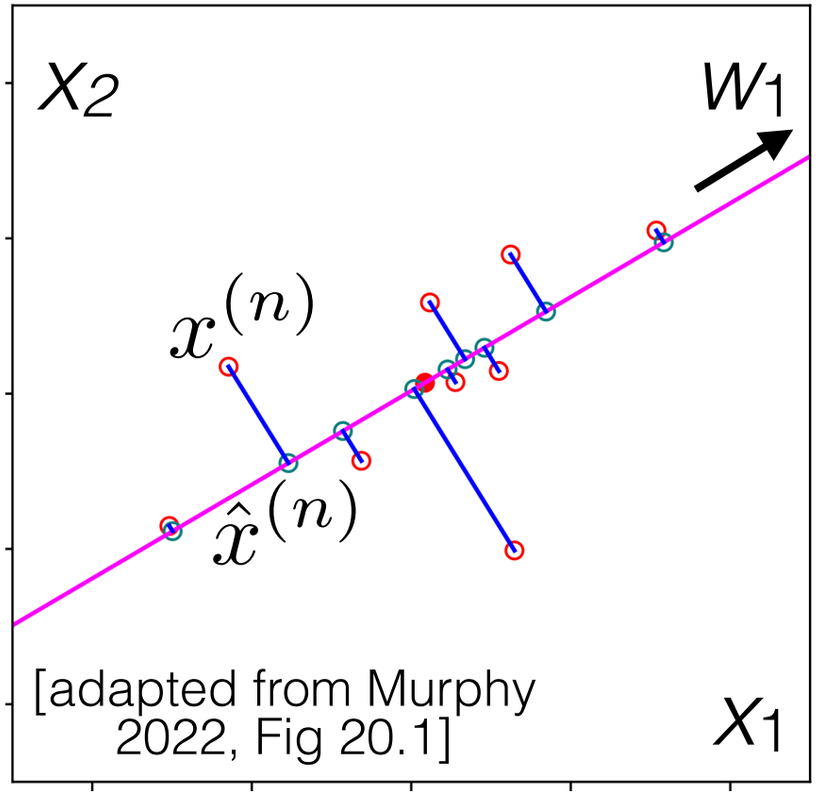
Challenges of PCA

- Choosing the number of latent dimensions *L*



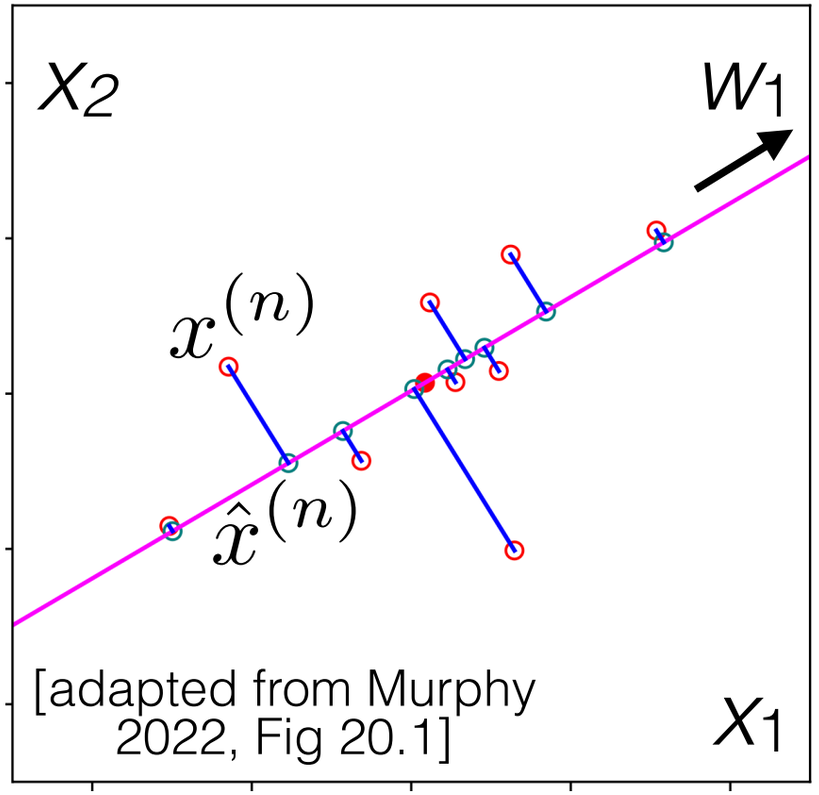
Challenges of PCA

- Choosing the number of latent dimensions *L*
  - Domain knowledge



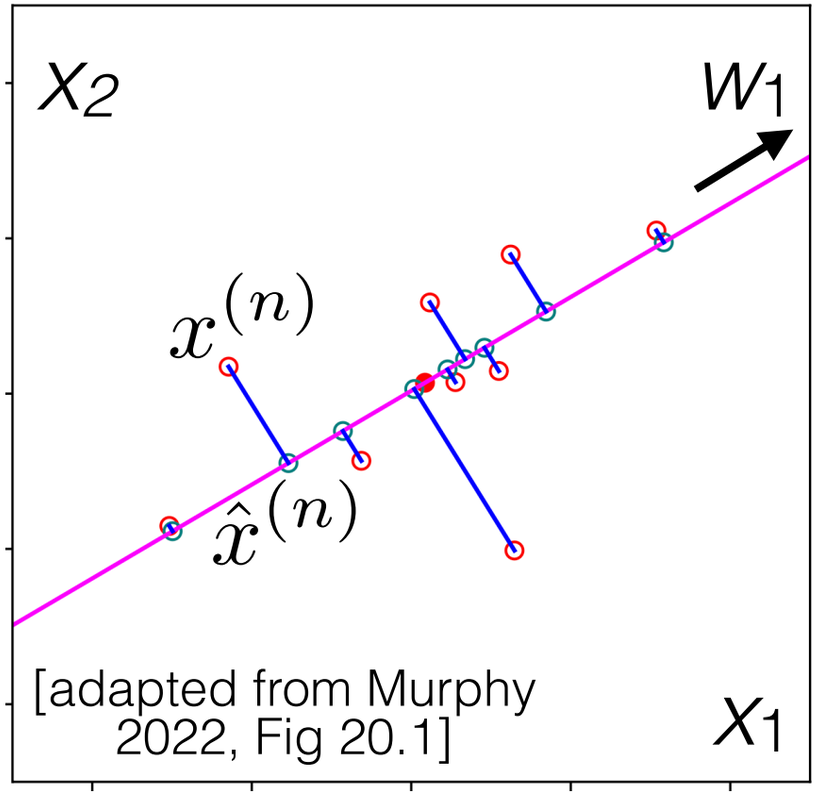
Challenges of PCA

- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2



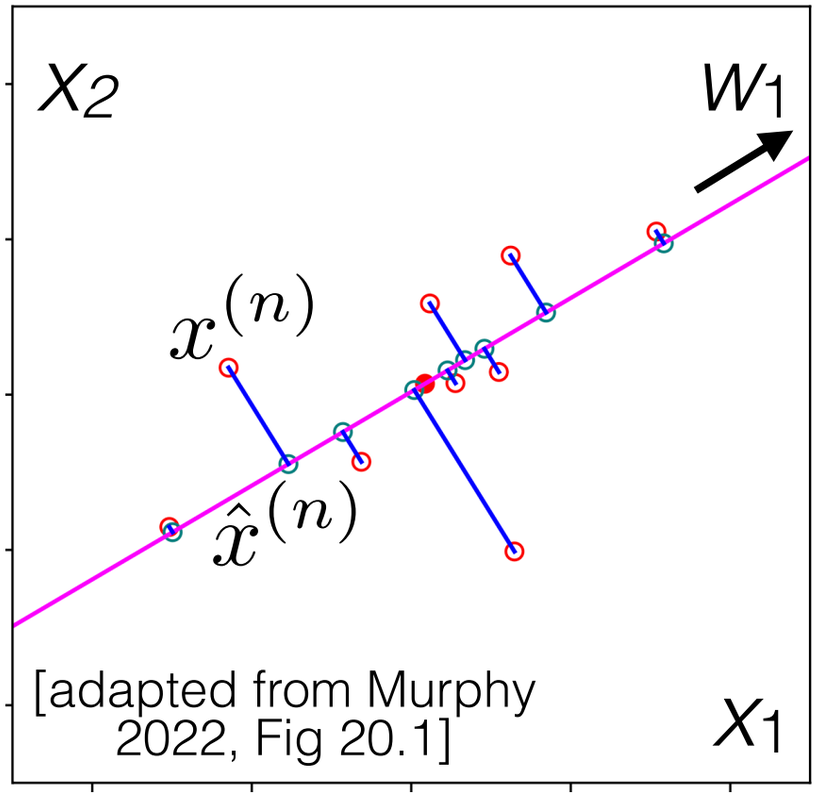
Challenges of PCA

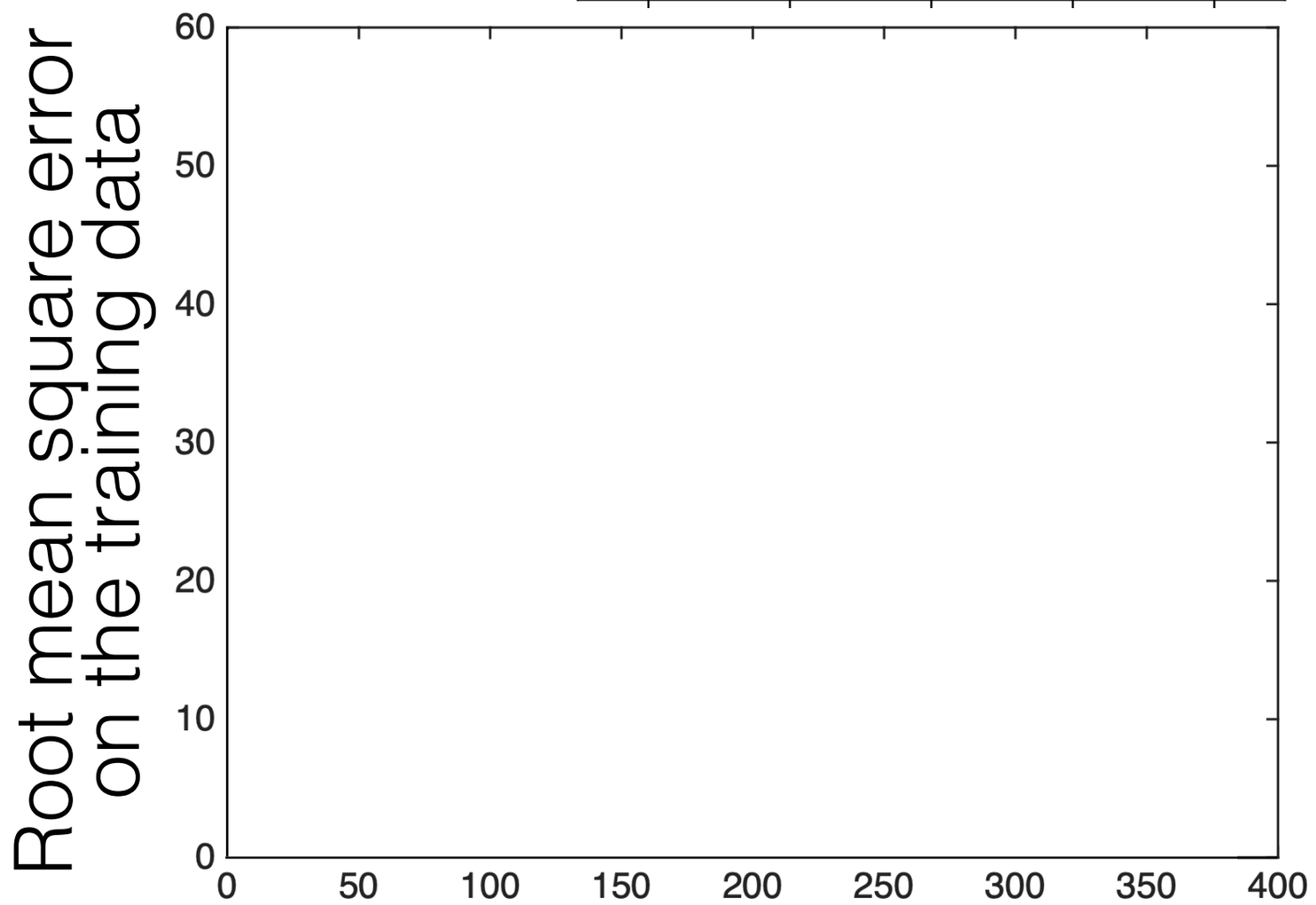
- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2
  - Plot loss



Challenges of PCA

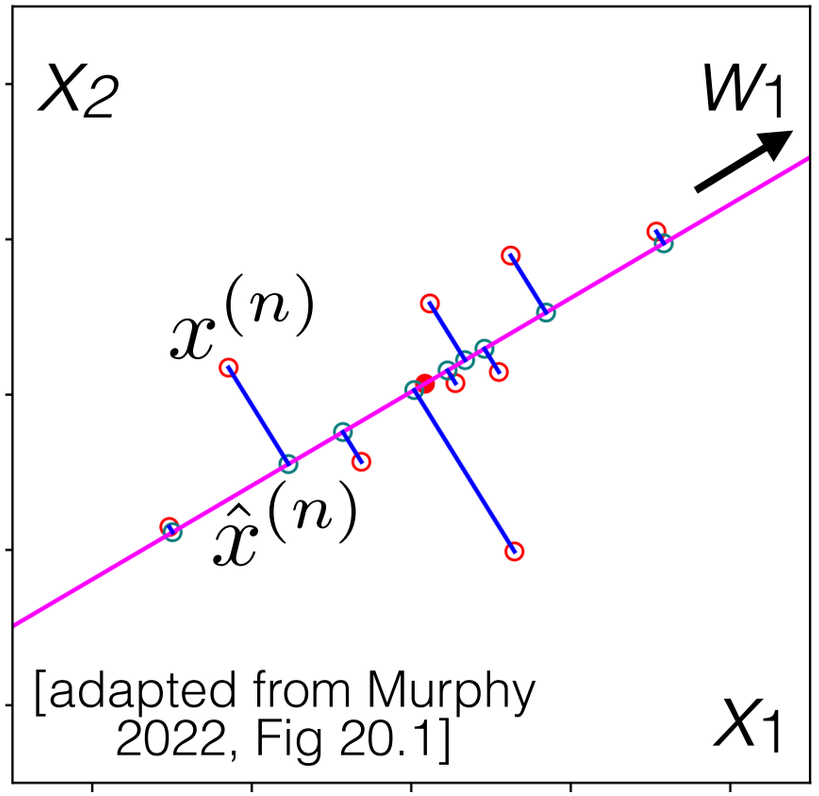
- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2
  - Plot loss

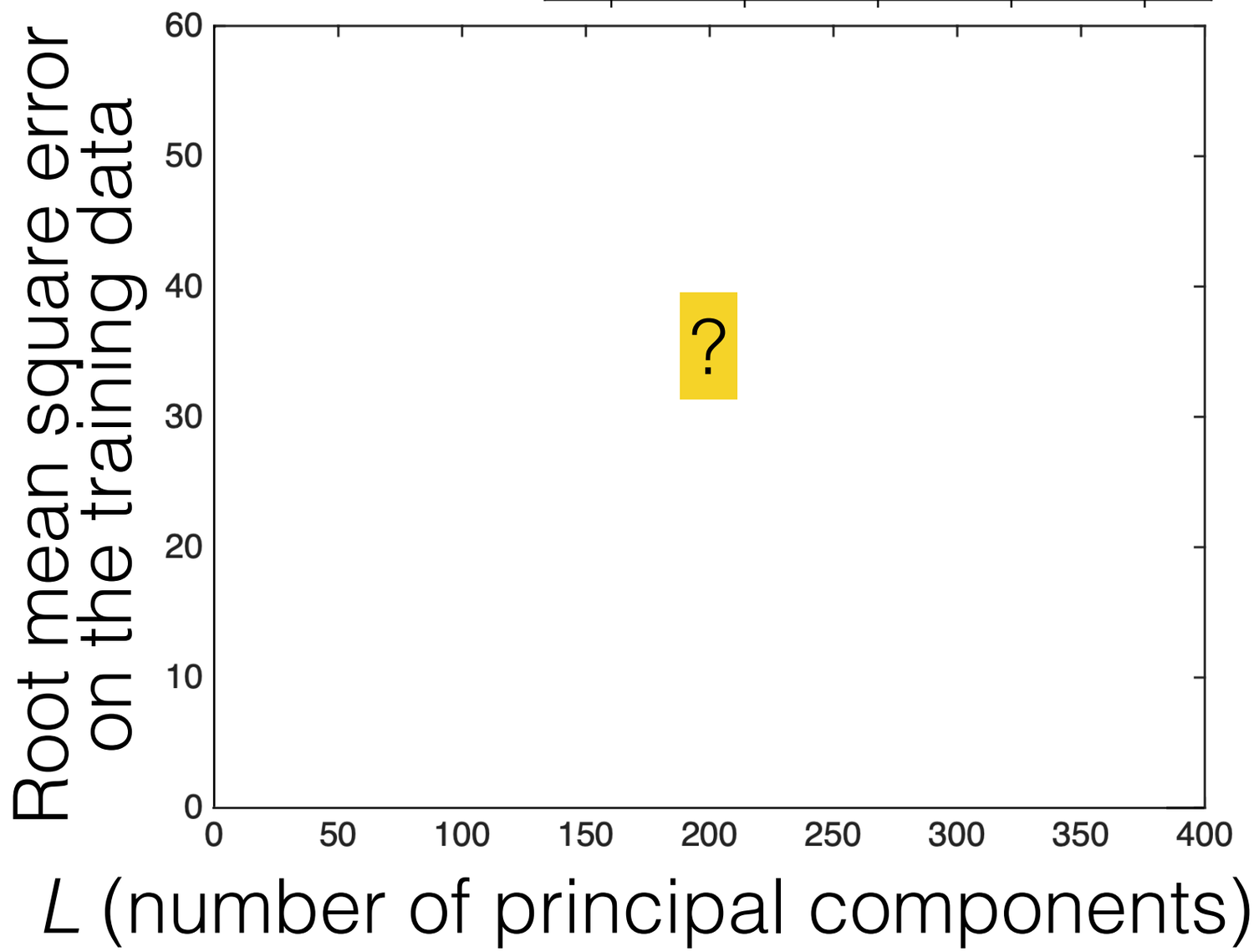




Challenges of PCA

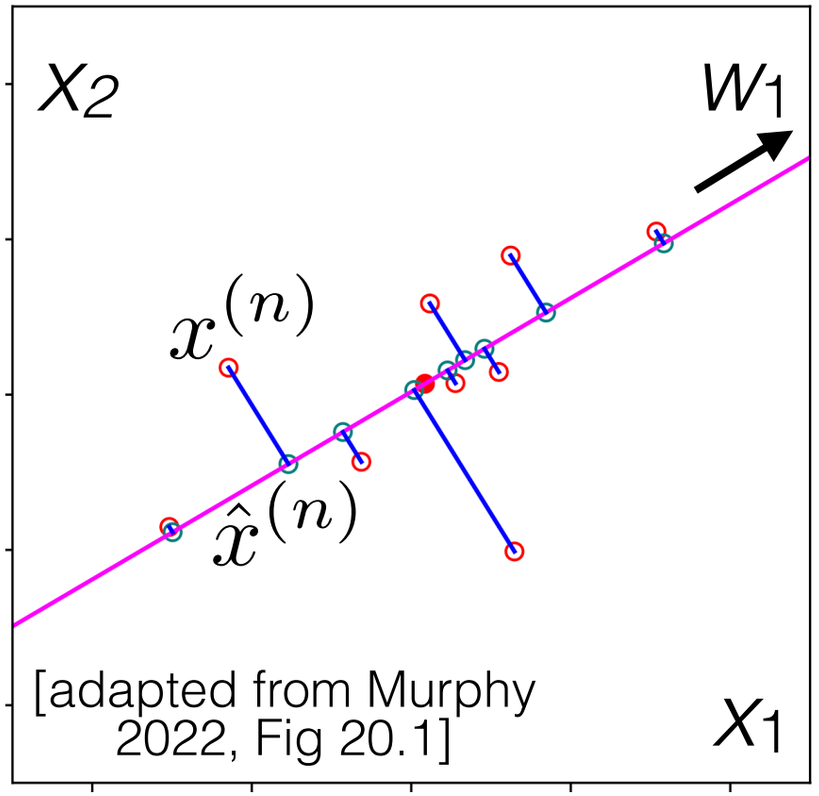
- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2
  - Plot loss

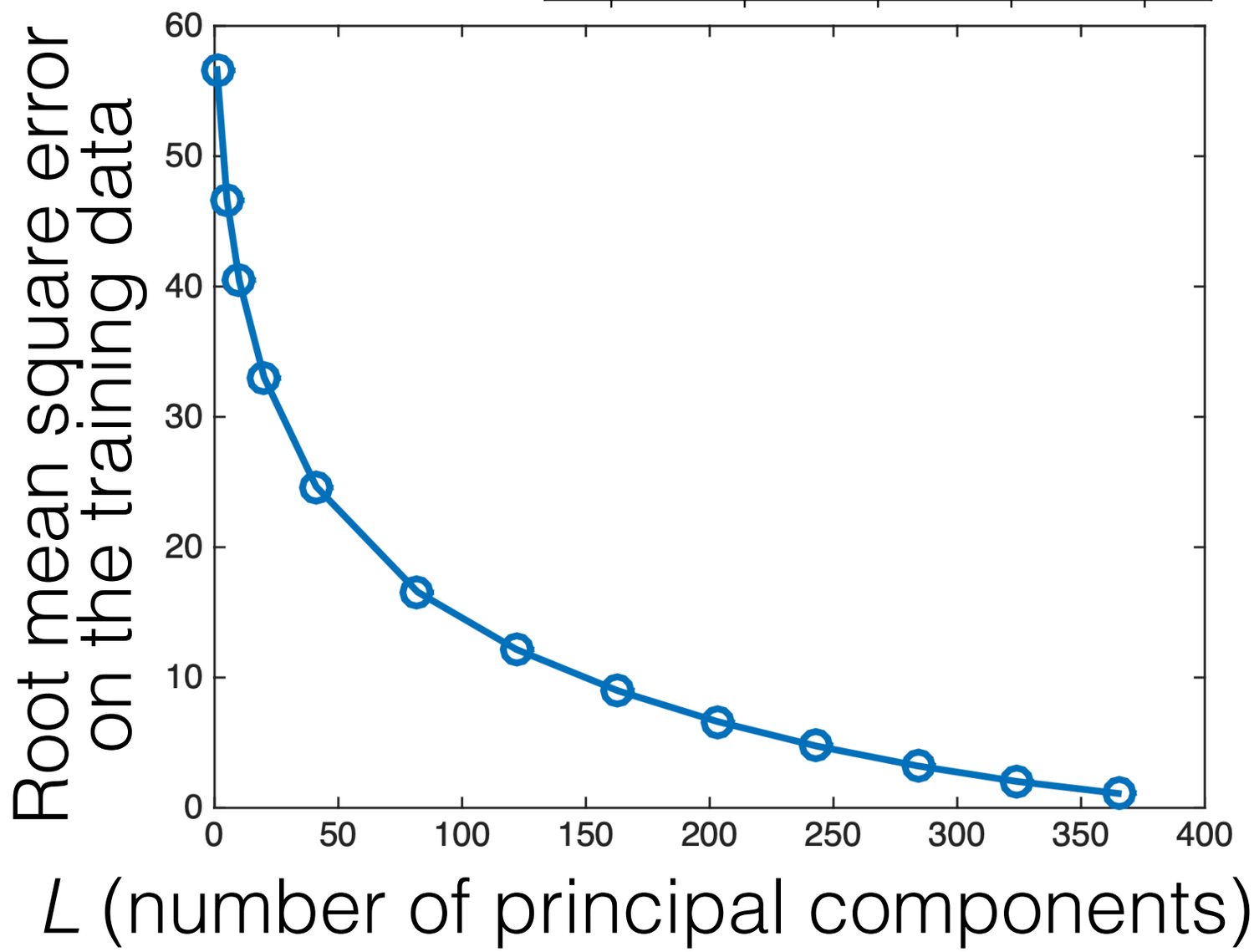




Challenges of PCA

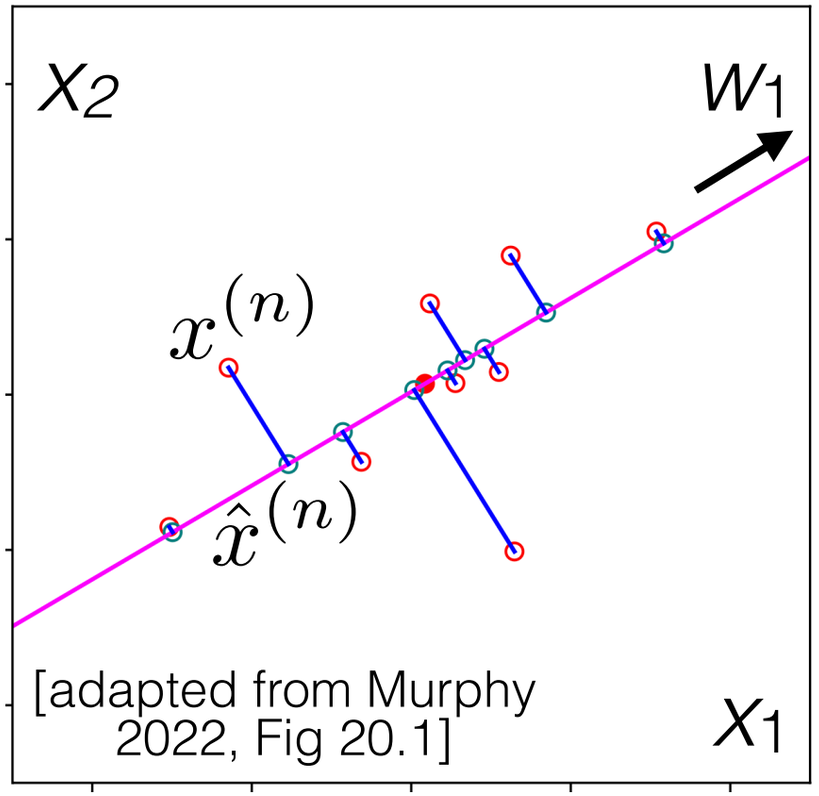
- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2
  - Plot loss



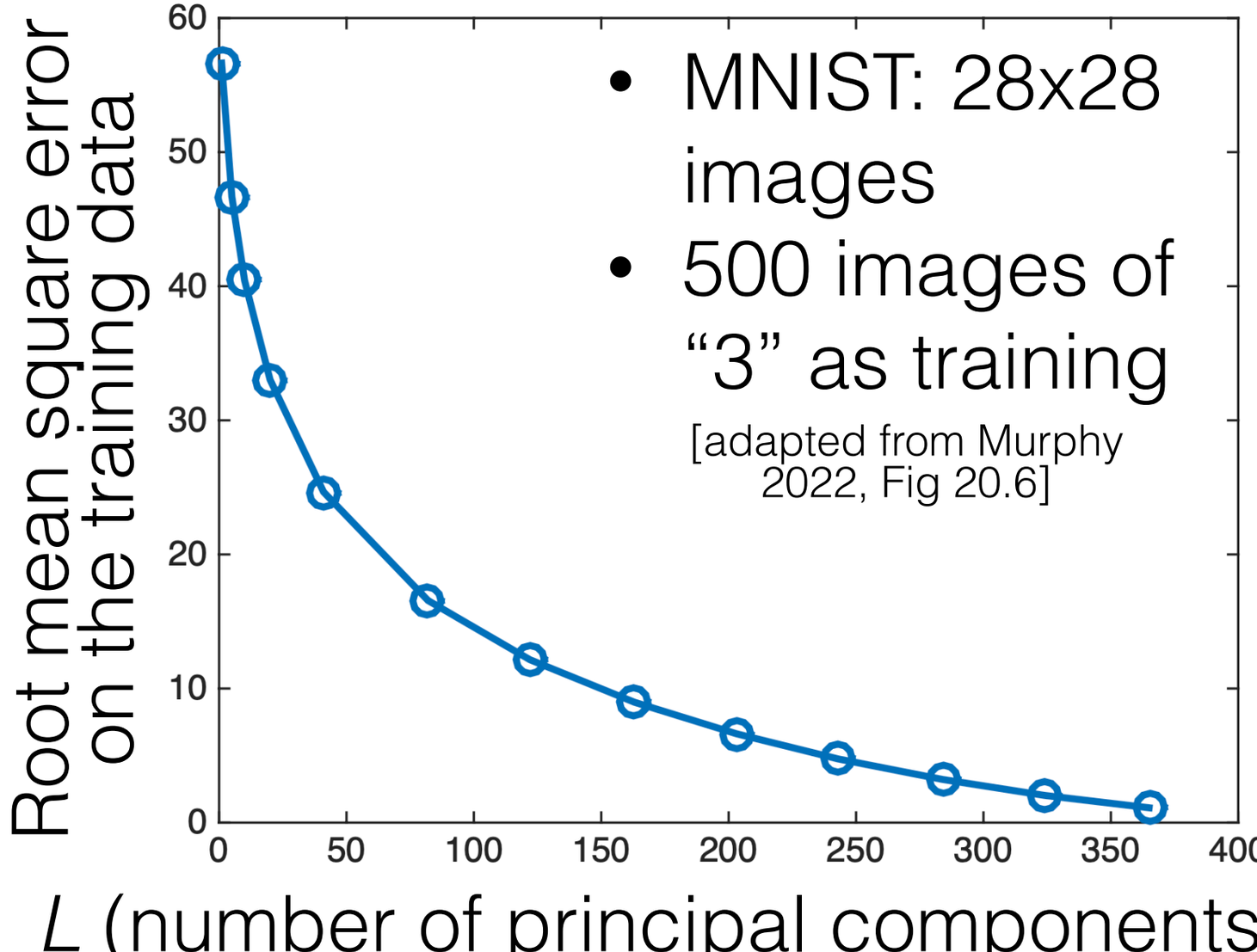


Challenges of PCA

- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2
  - Plot loss



[adapted from Murphy 2022, Fig 20.1]

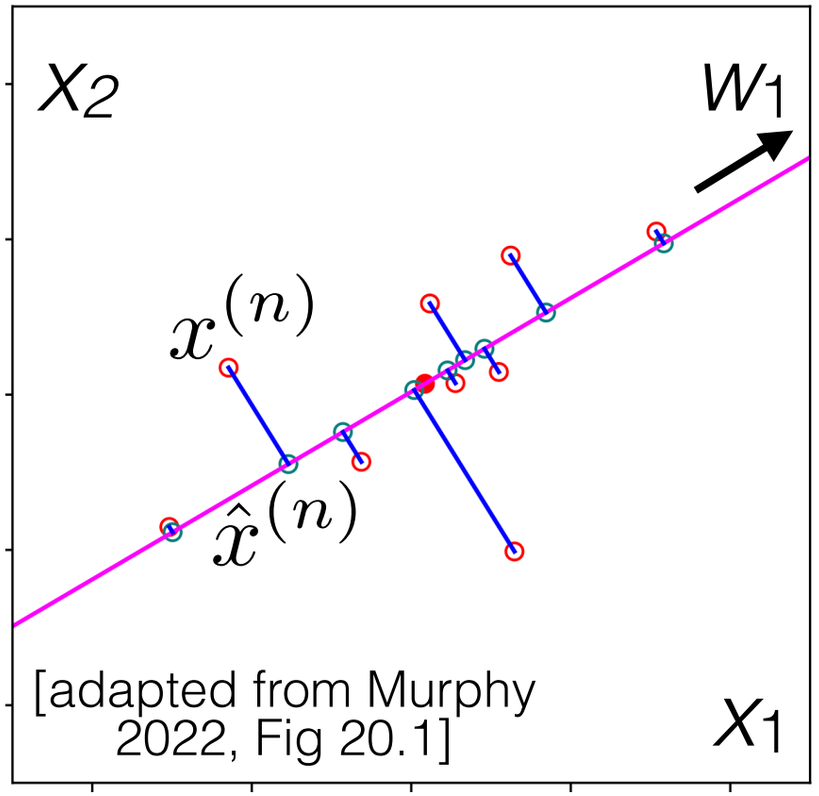


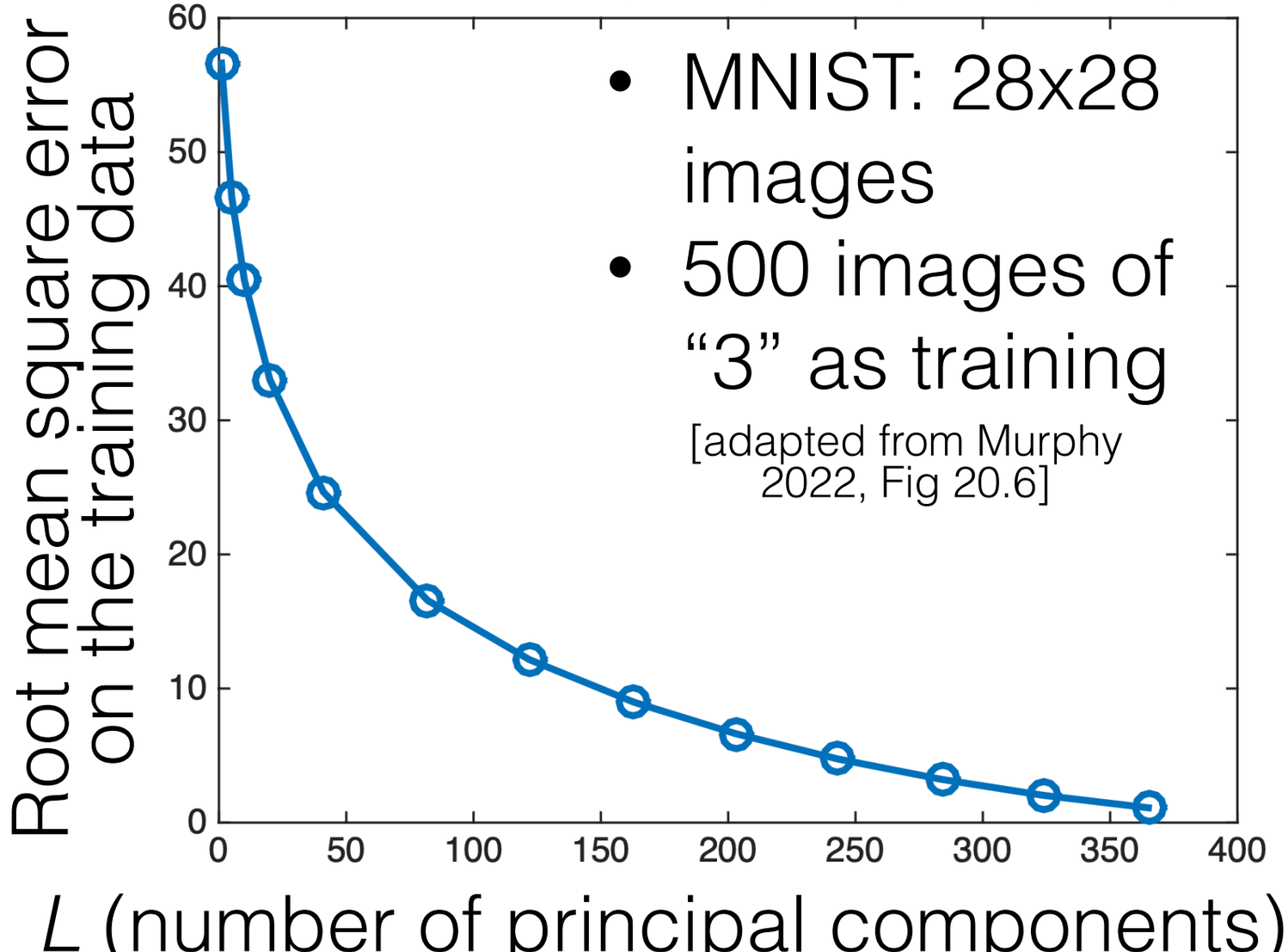
- MNIST: 28x28 images
- 500 images of "3" as training

[adapted from Murphy 2022, Fig 20.6]

Challenges of PCA

- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2
  - "Elbow" advice: Check for loss below a threshold or for when the loss levels out



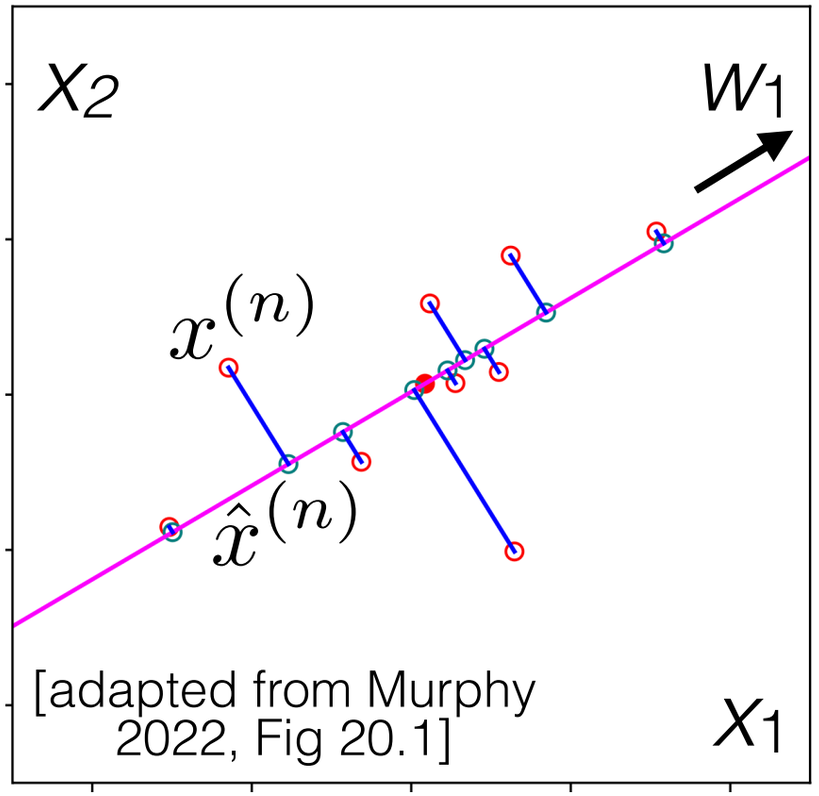


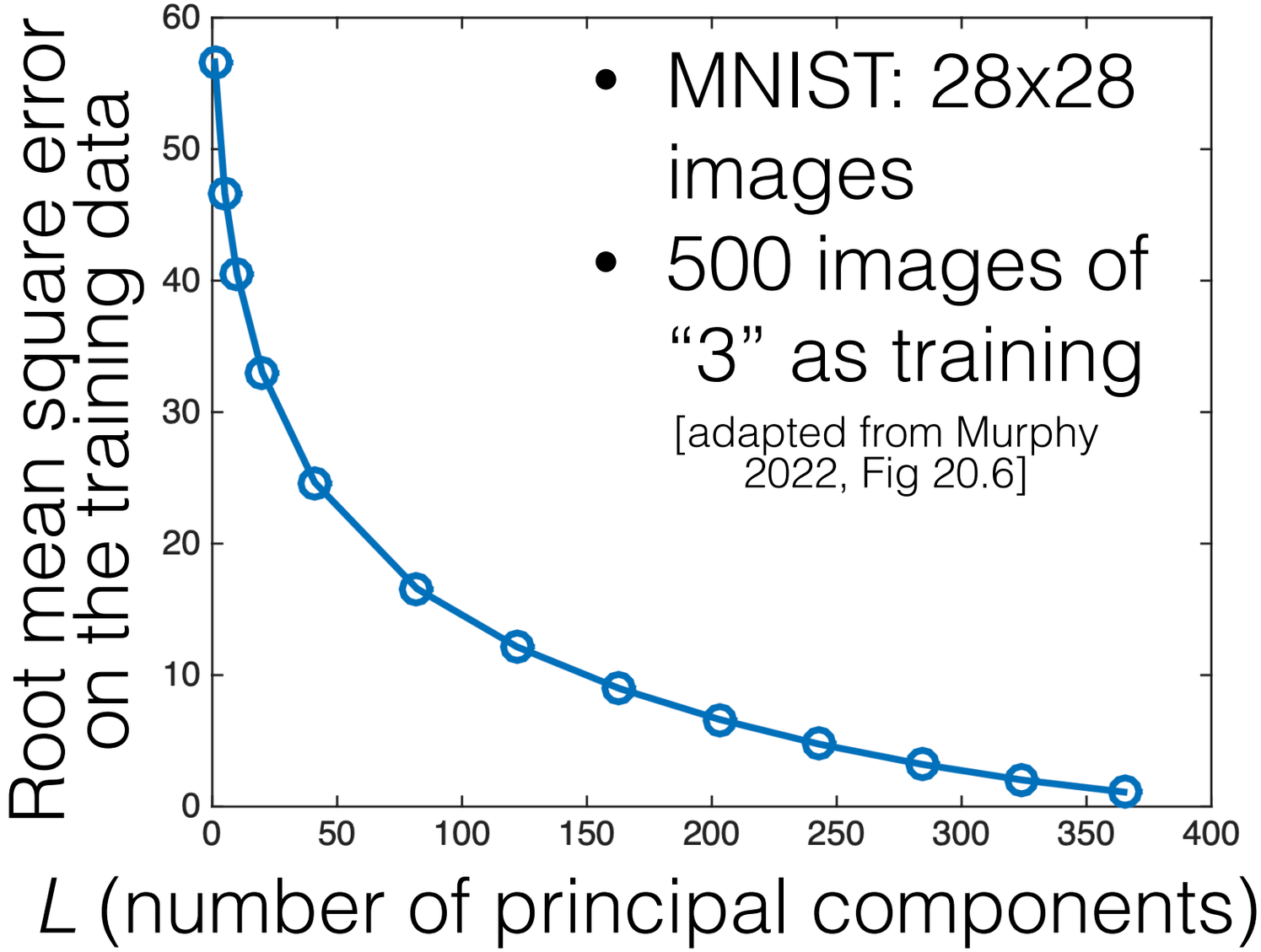
- MNIST: 28x28 images
- 500 images of "3" as training

[adapted from Murphy 2022, Fig 20.6]

Challenges of PCA

- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2
  - "Elbow" advice: Check for loss below a threshold or for when the loss levels out
- Accept that there need not be a "true" or "correct" value of *L*



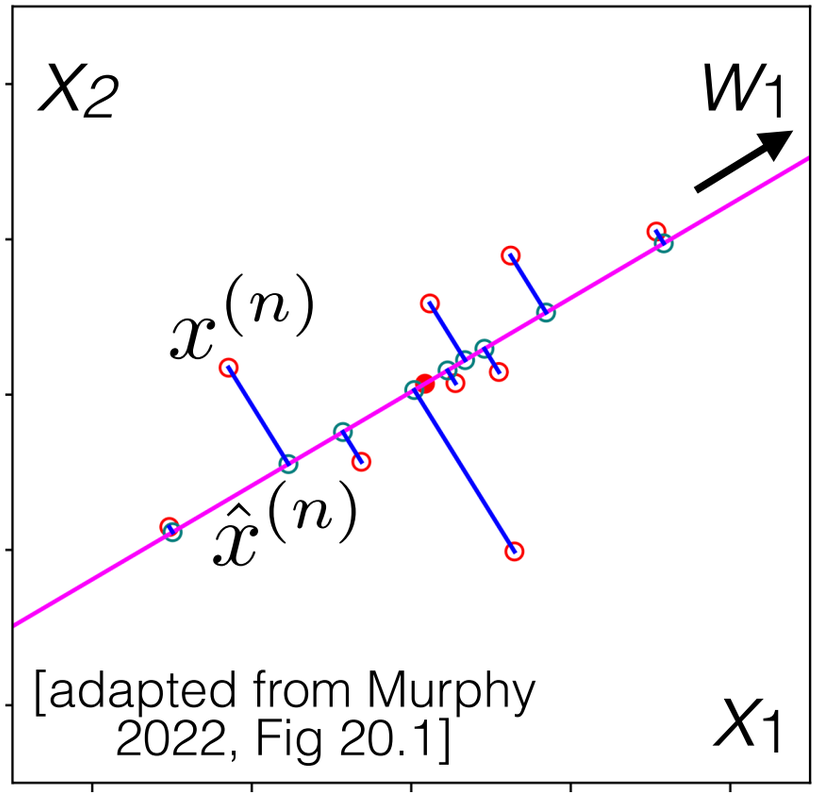


- MNIST: 28x28 images
- 500 images of "3" as training

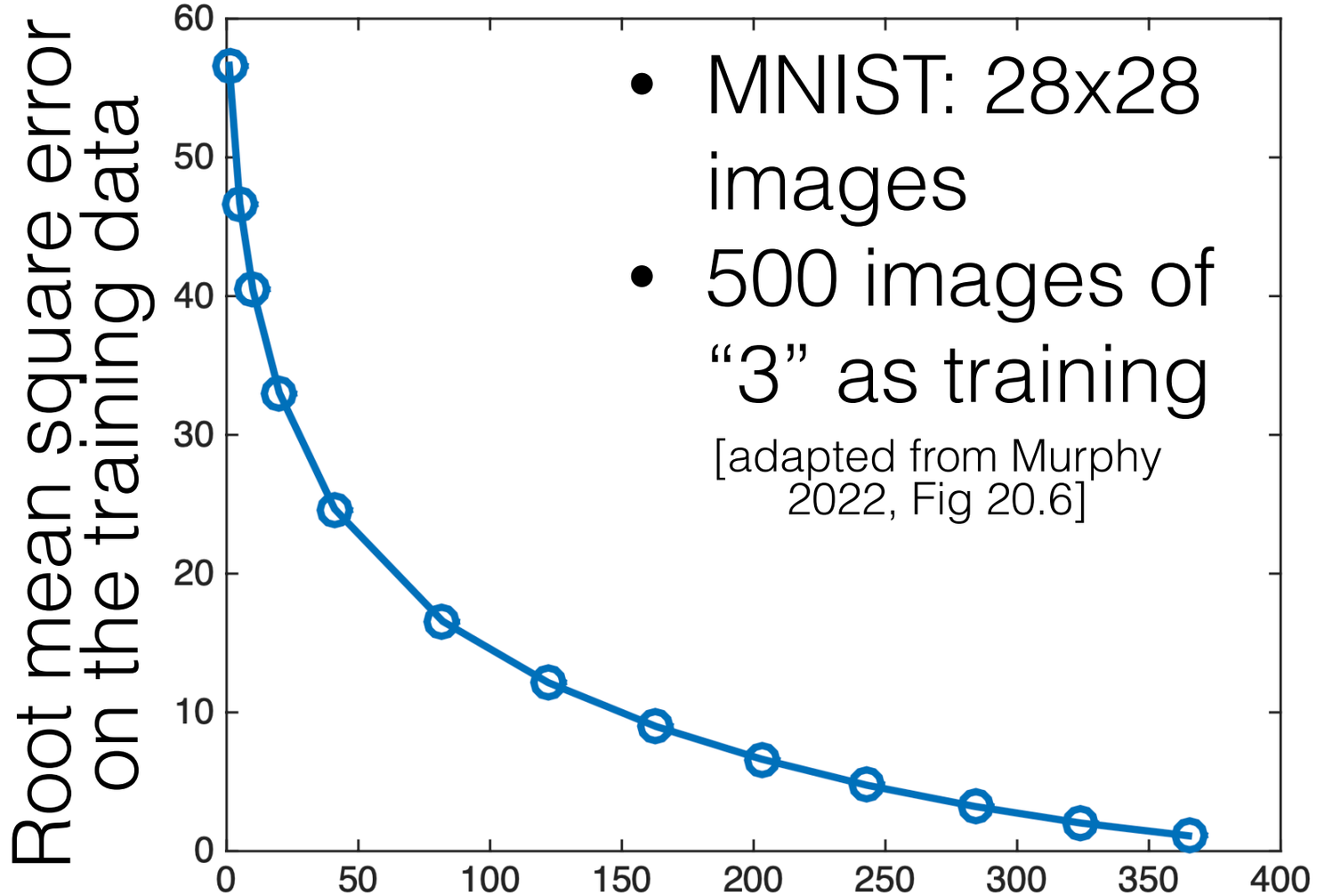
[adapted from Murphy 2022, Fig 20.6]

Challenges of PCA

- Choosing the number of latent dimensions *L*
  - Domain knowledge
  - If you're making a plot, typically *L*=2
  - "Elbow" advice: Check for loss below a threshold or for when the loss levels out
  - Accept that there need not be a "true" or "correct" value of *L*
  - Cf. clustering & the # of clusters



[adapted from Murphy 2022, Fig 20.1]



- MNIST: 28x28 images
- 500 images of "3" as training

[adapted from Murphy 2022, Fig 20.6]

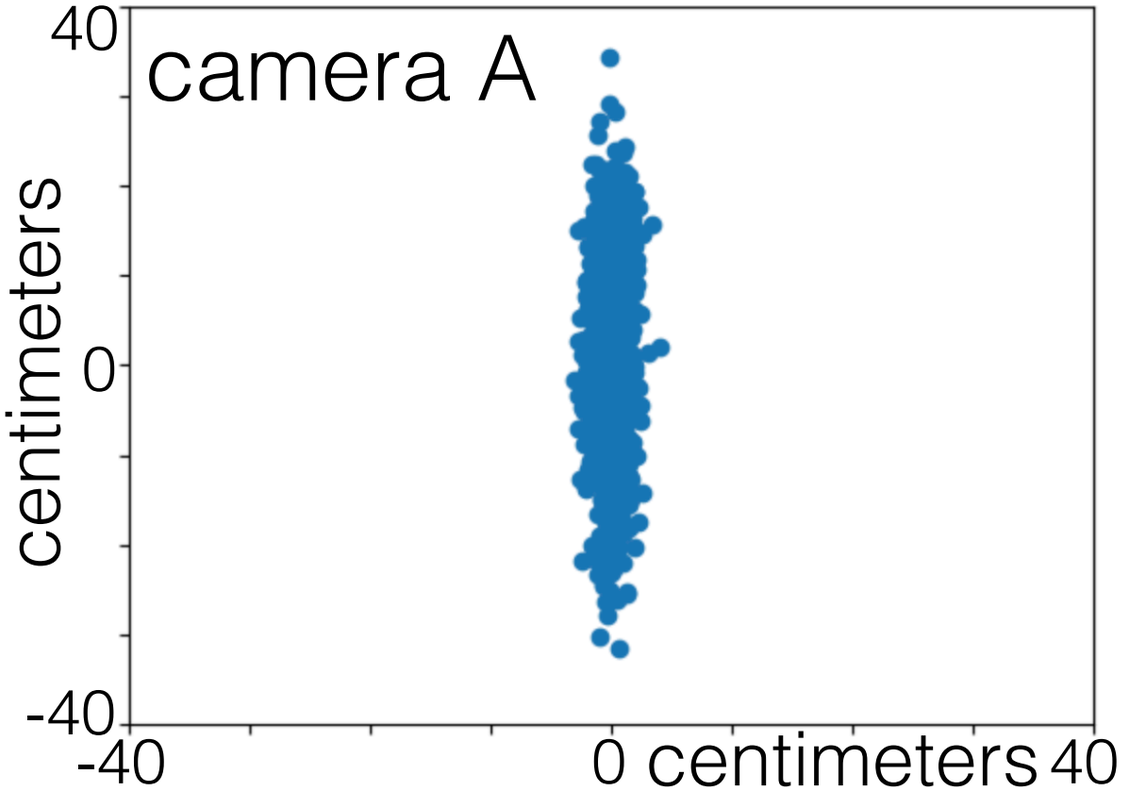
Challenges of PCA

Challenges of PCA

- Scale of the data matters

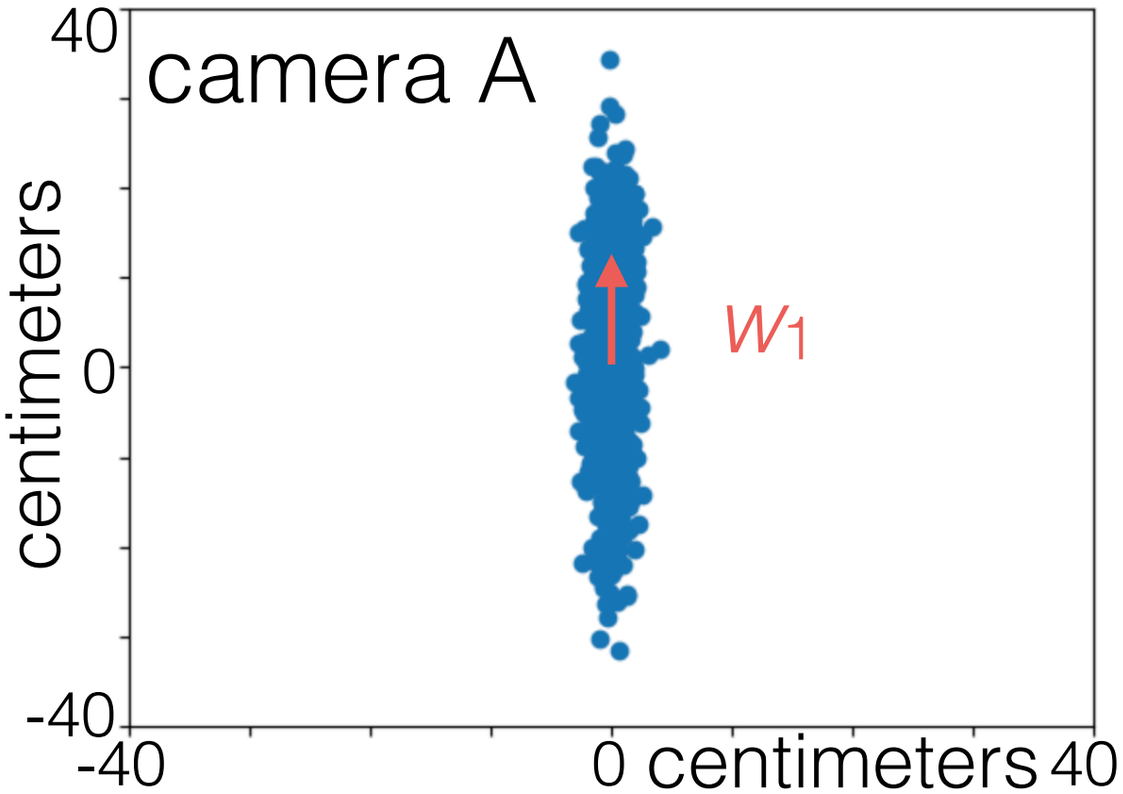
Challenges of PCA

- Scale of the data matters



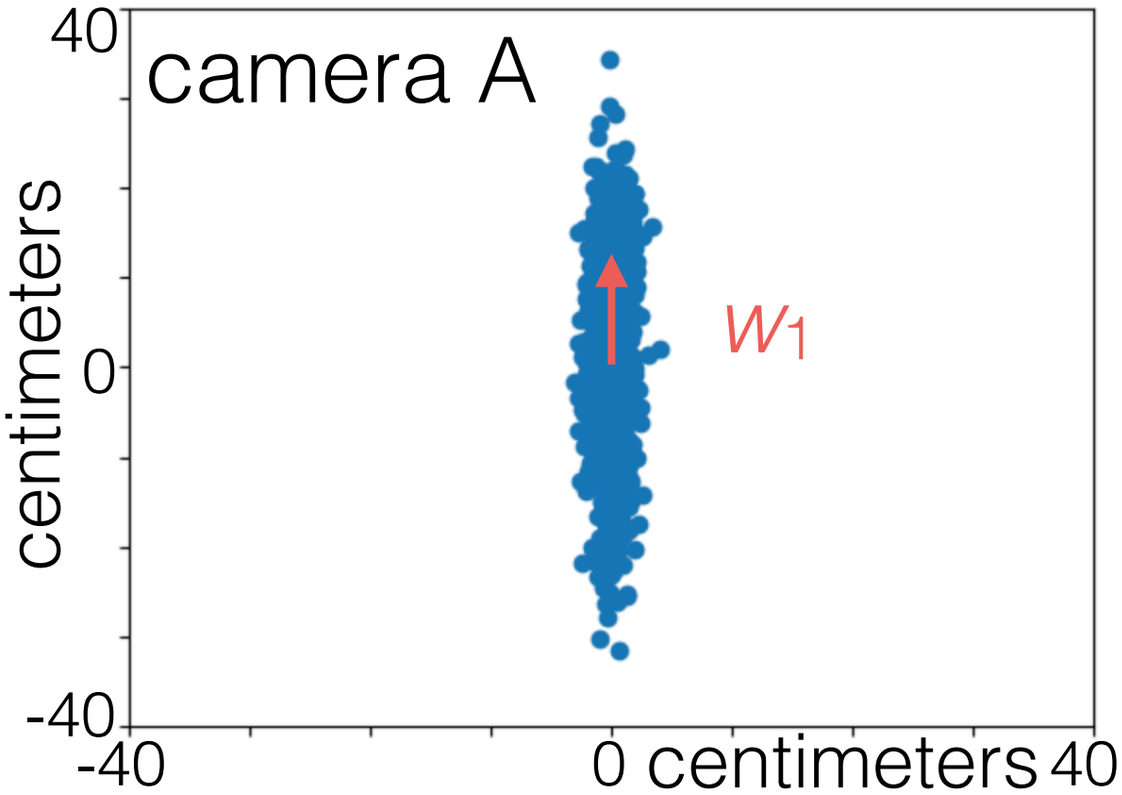
Challenges of PCA

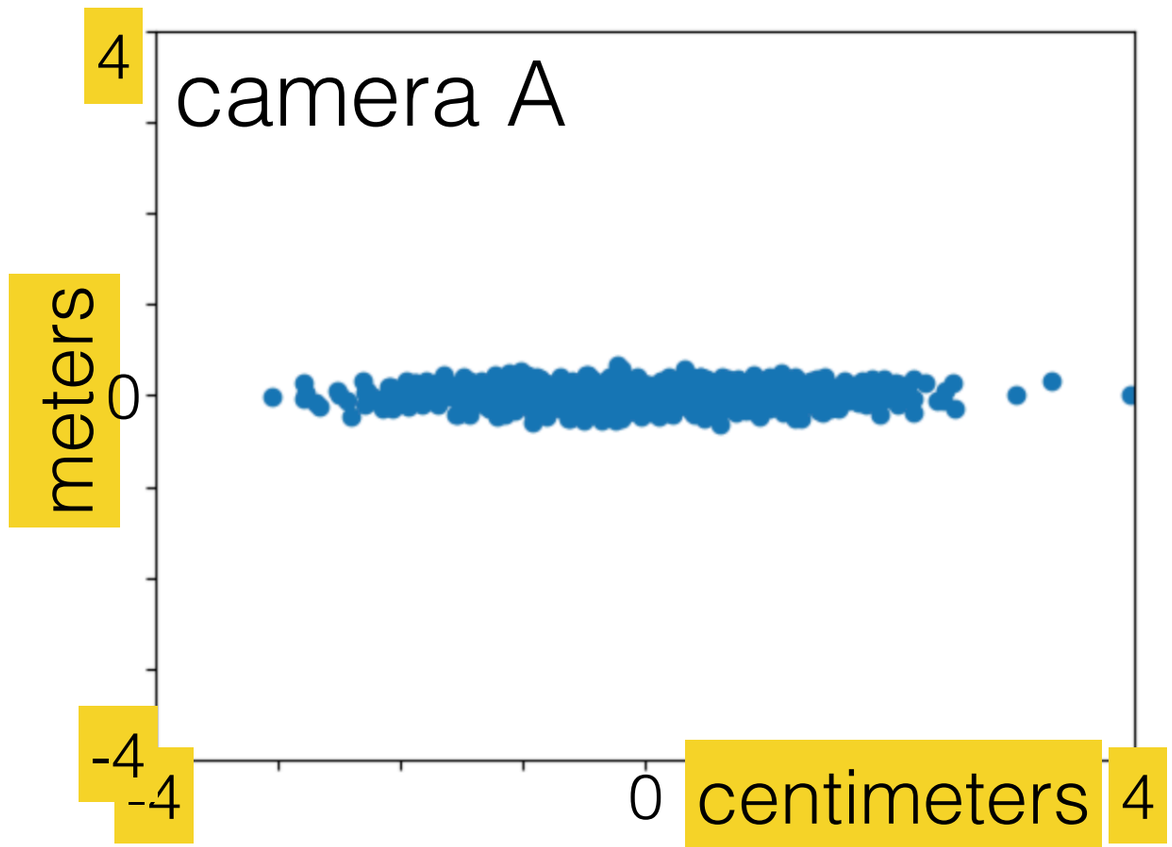
- Scale of the data matters



Challenges of PCA

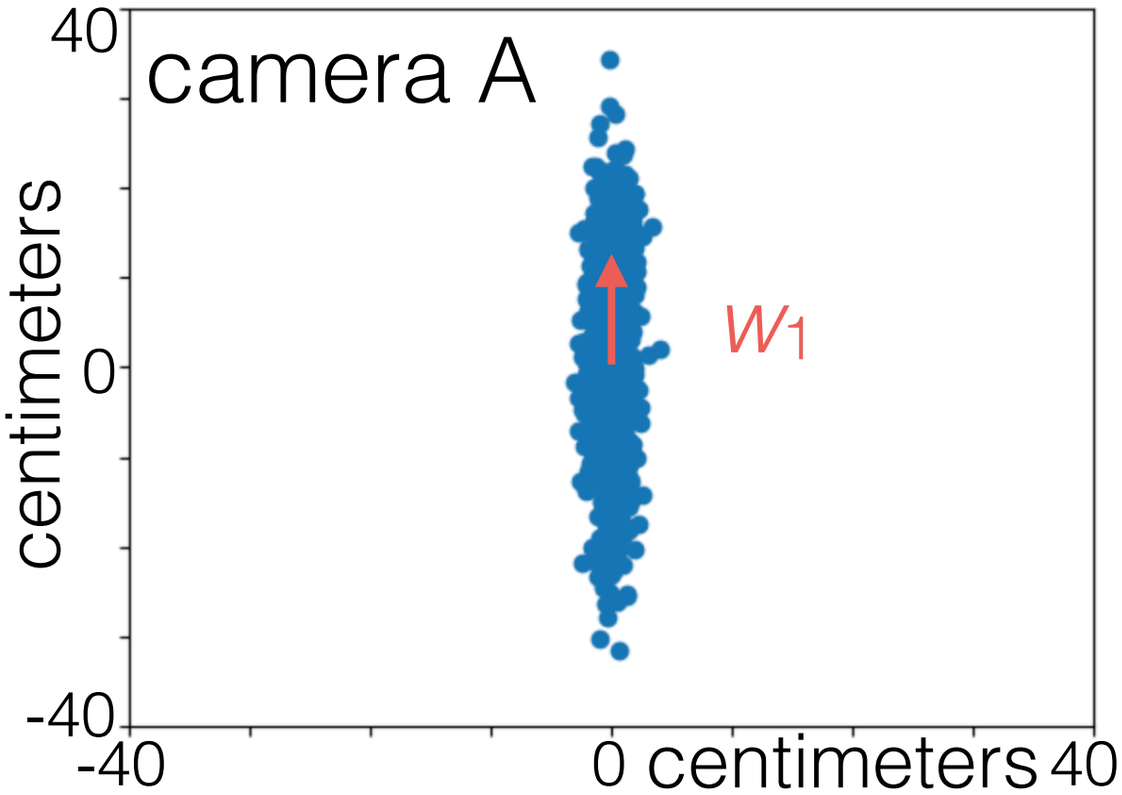
- Scale of the data matters

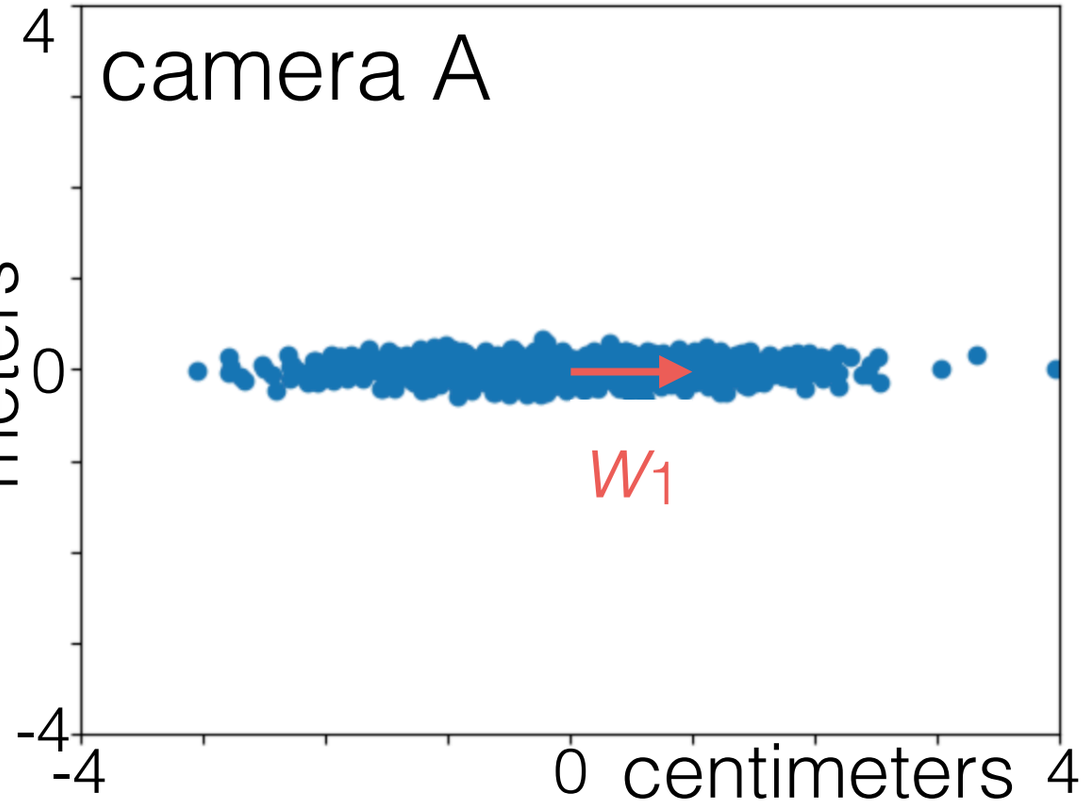




Challenges of PCA

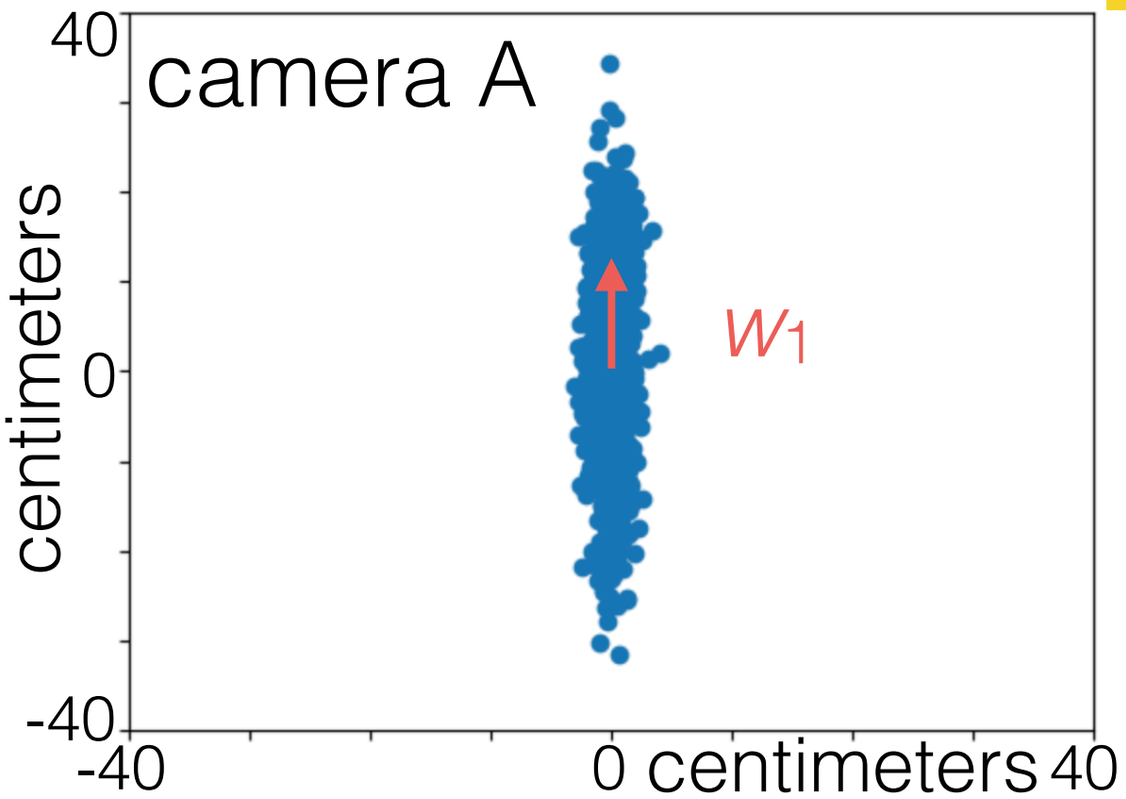
- Scale of the data matters

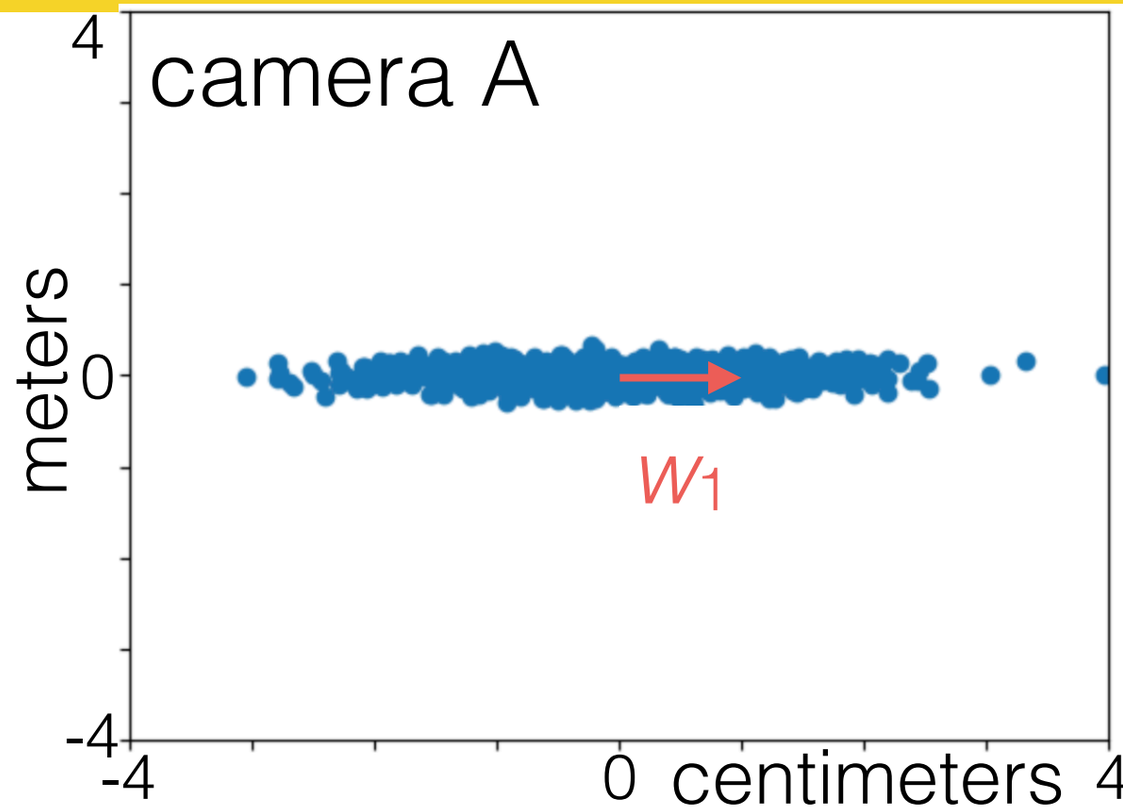




Challenges of PCA

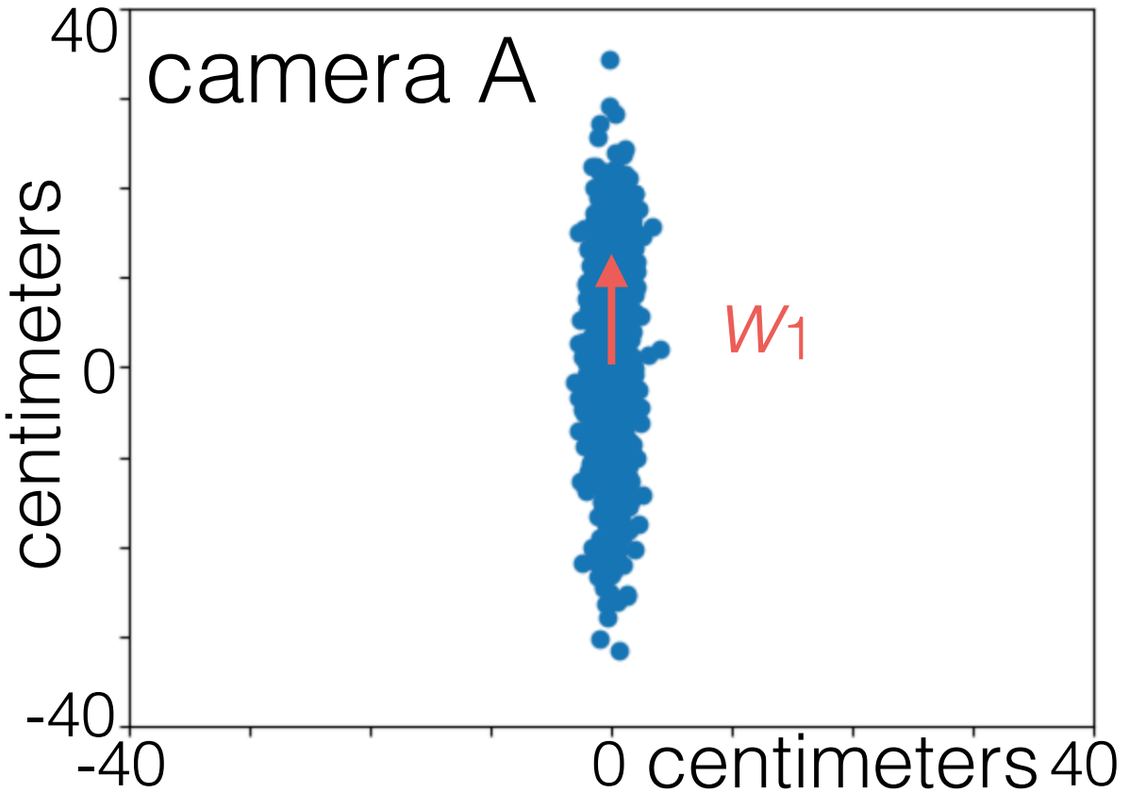
- Scale of the data matters; should we normalize data first?

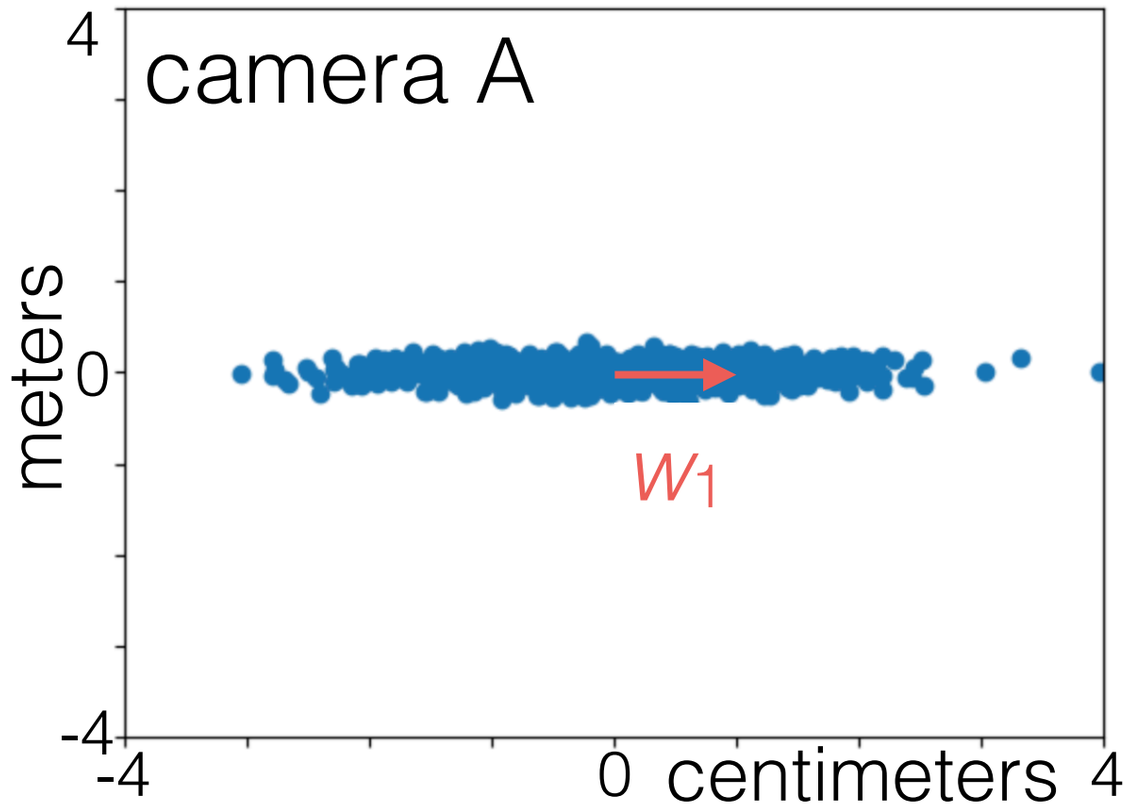


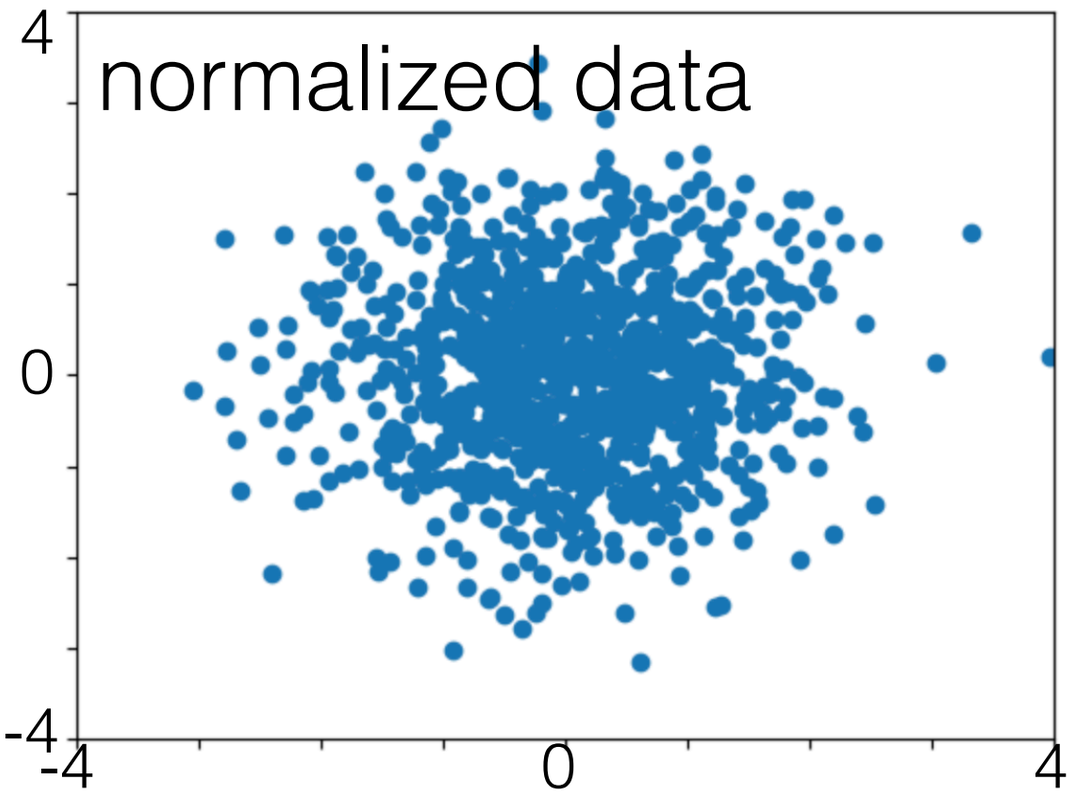


Challenges of PCA

- Scale of the data matters; should we normalize data first?

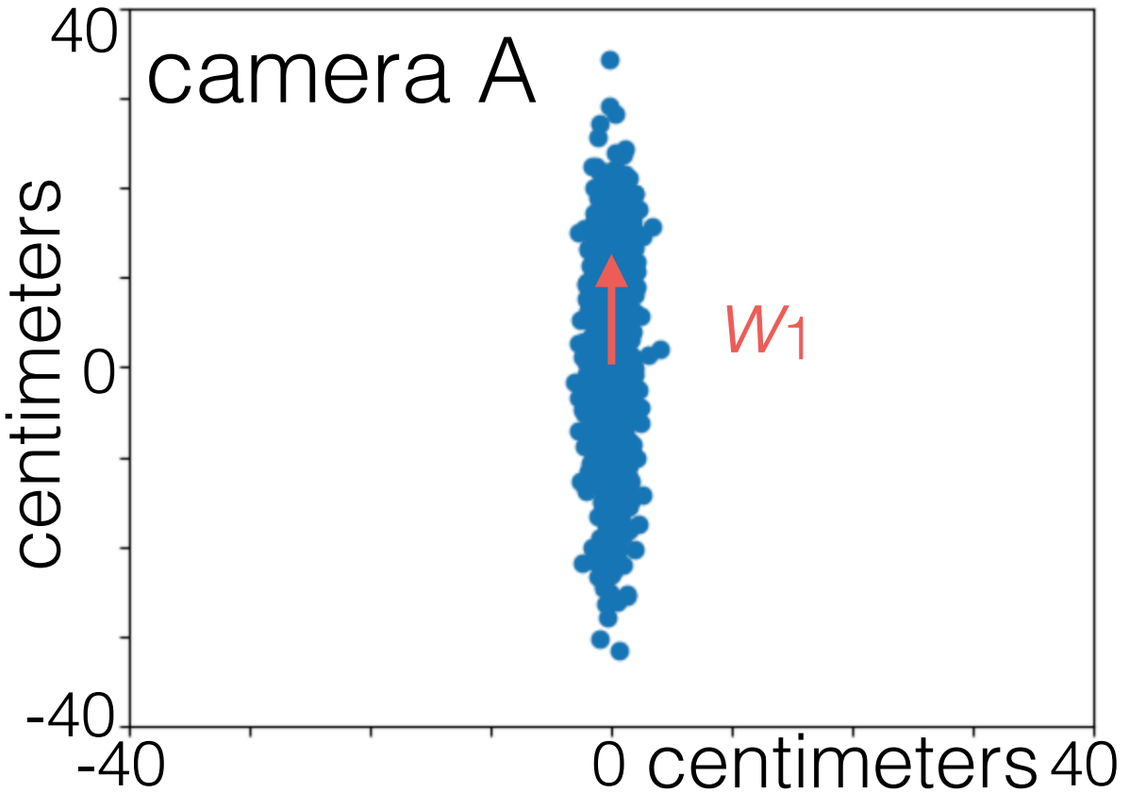


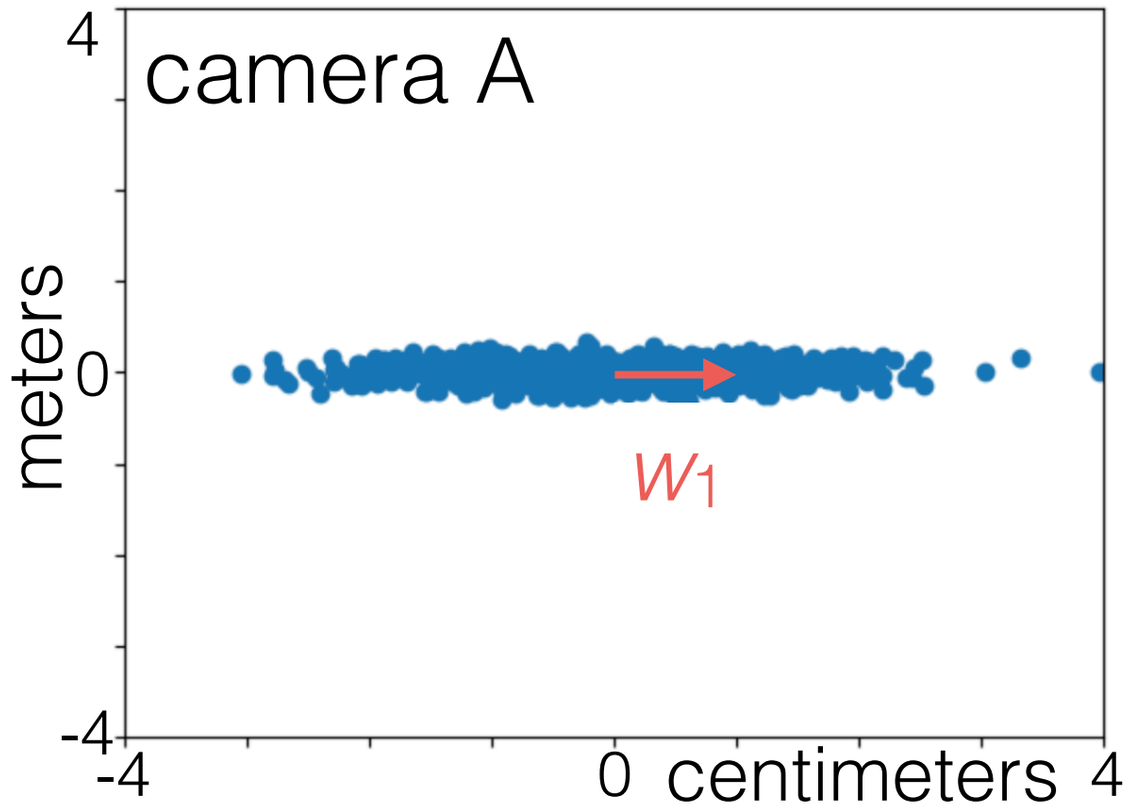


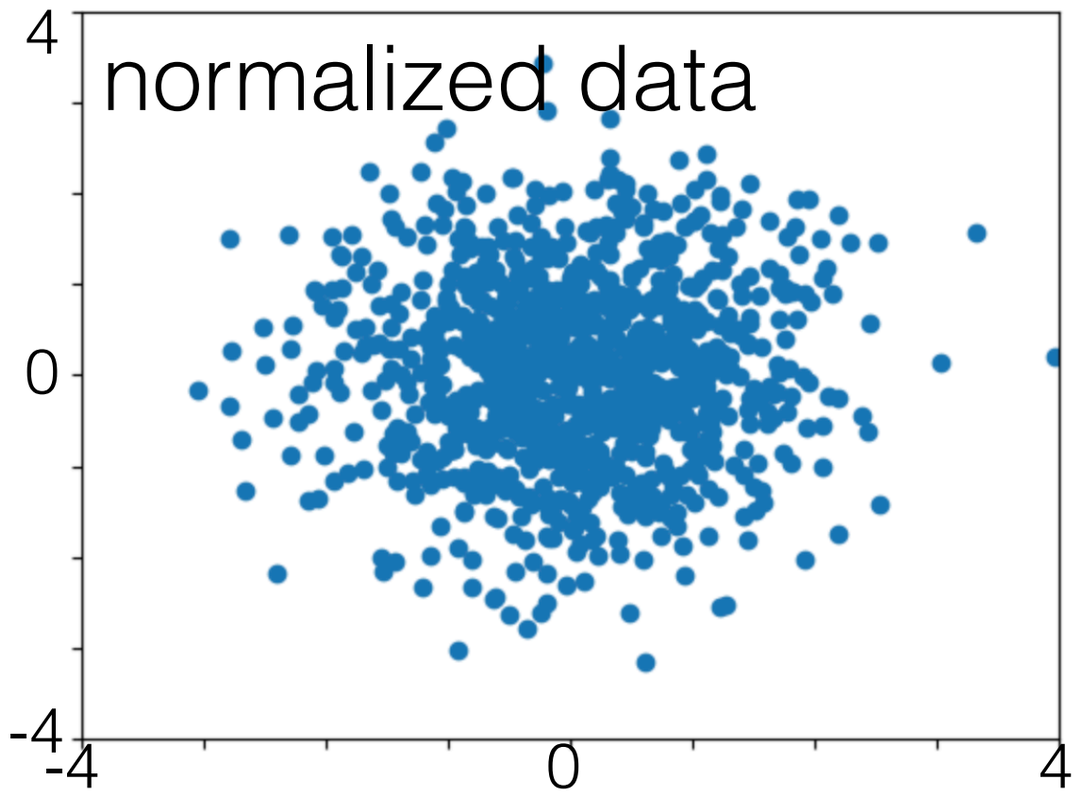


Challenges of PCA

- Scale of the data matters; should we normalize data first?







"We have been working with the eigendecomposition of the covariance matrix. However, it is better to use the correlation matrix instead. […O]therwise PCA can be "misled" by directions in which the variance is high merely because of the measurement scale." [Murphy 2022, 20.1.3.1]

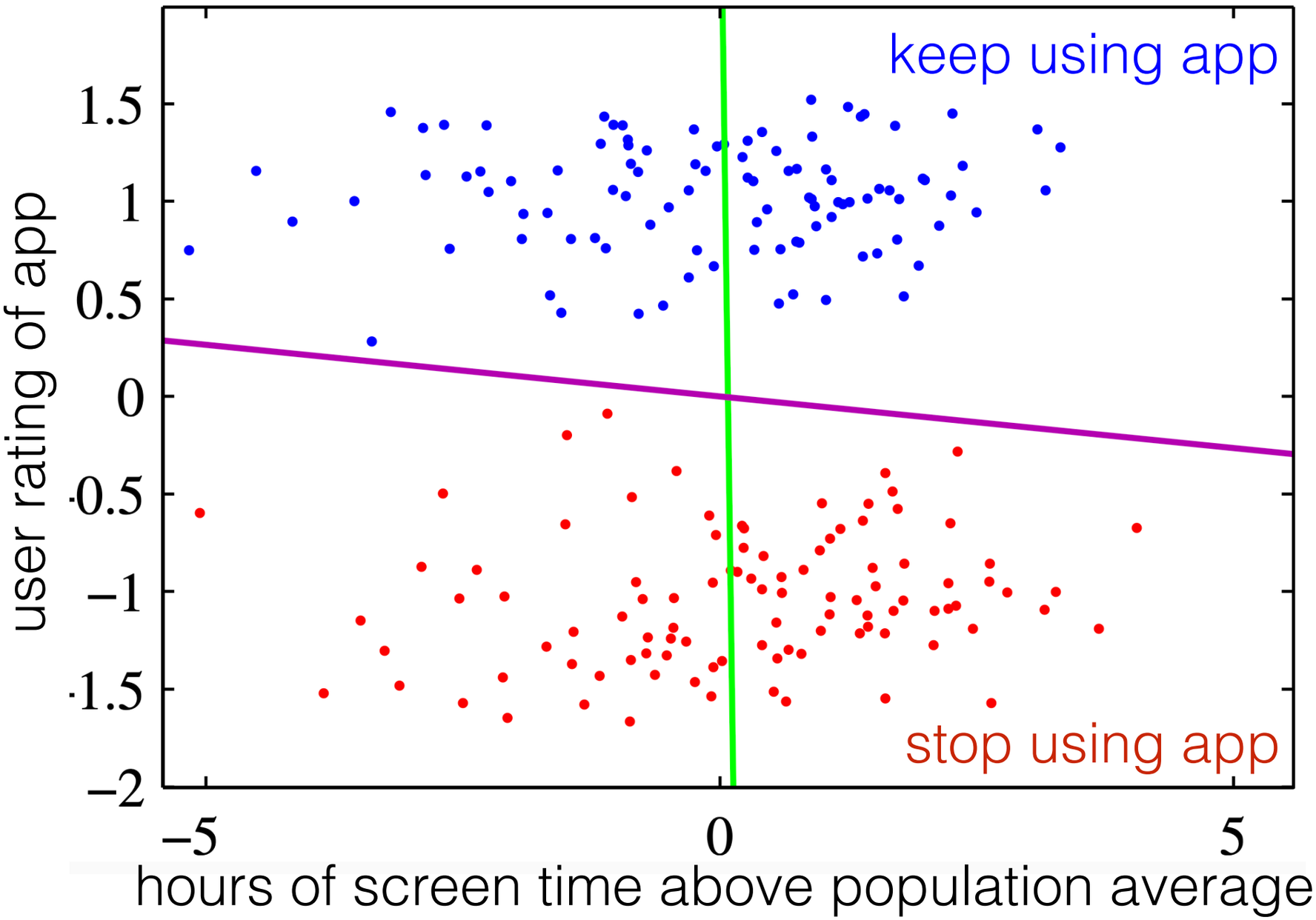
Challenges of PCA

Challenges of PCA

- Proposal: reduce the dimensionality of the feature space with PCA before running supervised learning

Challenges of PCA

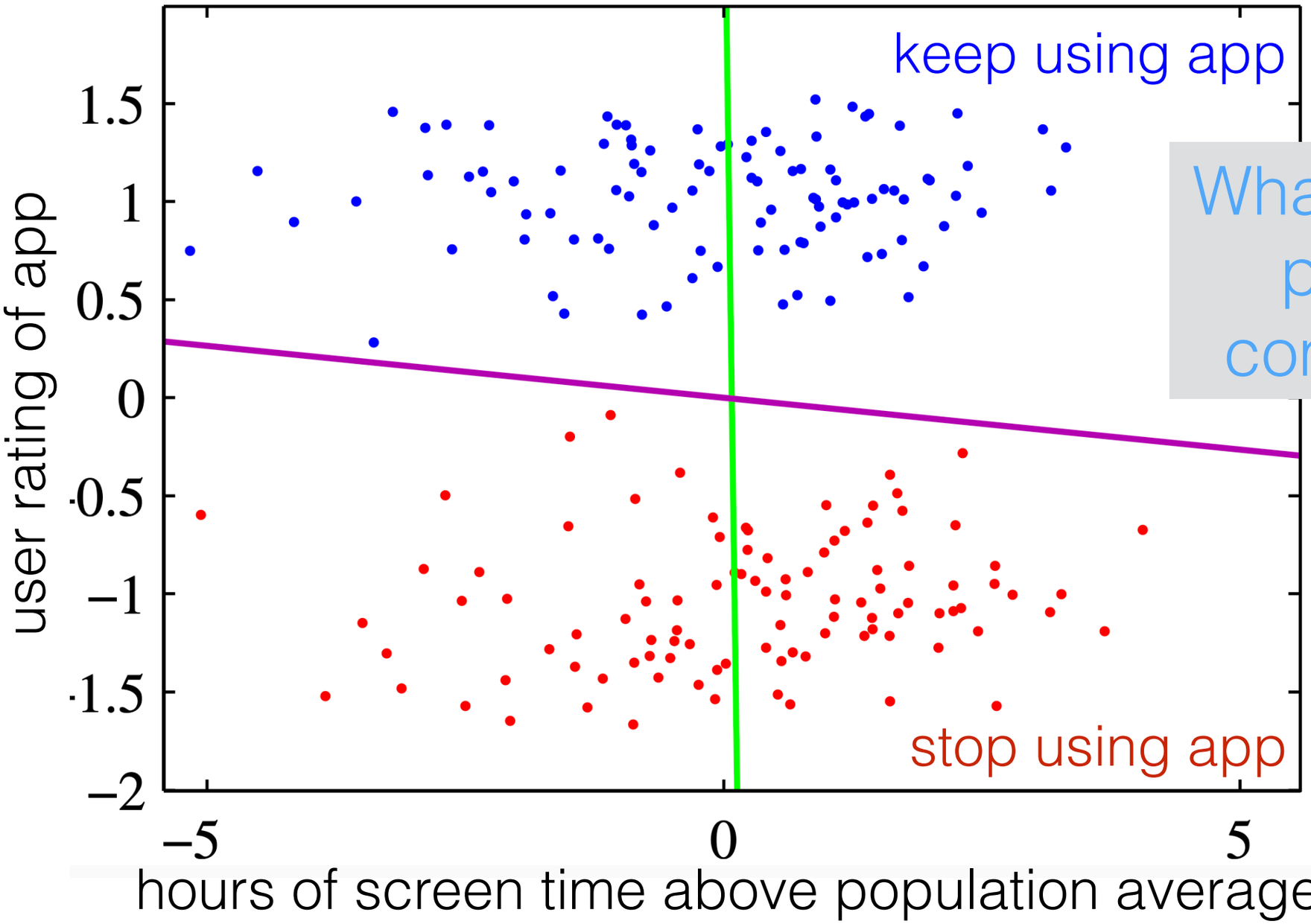
- Proposal: reduce the dimensionality of the feature space with PCA before running supervised learning



[adapted from Bishop 2006, Fig 12.7]

Challenges of PCA

- Proposal: reduce the dimensionality of the feature space with PCA before running supervised learning

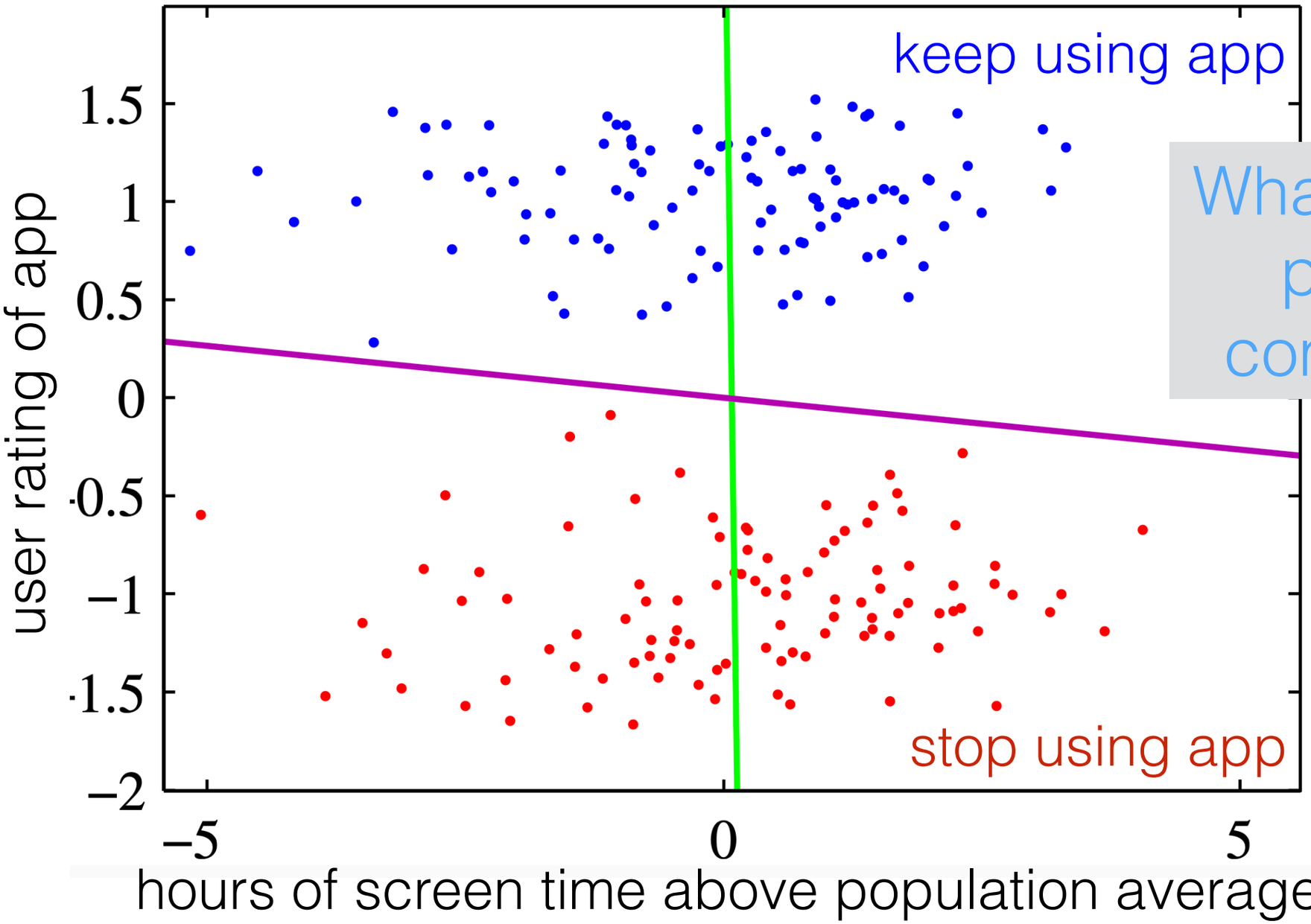


What's the first principal component?

[adapted from Bishop 2006, Fig 12.7]

Challenges of PCA

- Proposal: reduce the dimensionality of the feature space with PCA before running supervised learning



What's the first principal component?

[adapted from Bishop 2006, Fig 12.7]

- This pre-processing assumes directions of highest feature variance are the meaningful ones for prediction

Challenges of PCA

Challenges of PCA

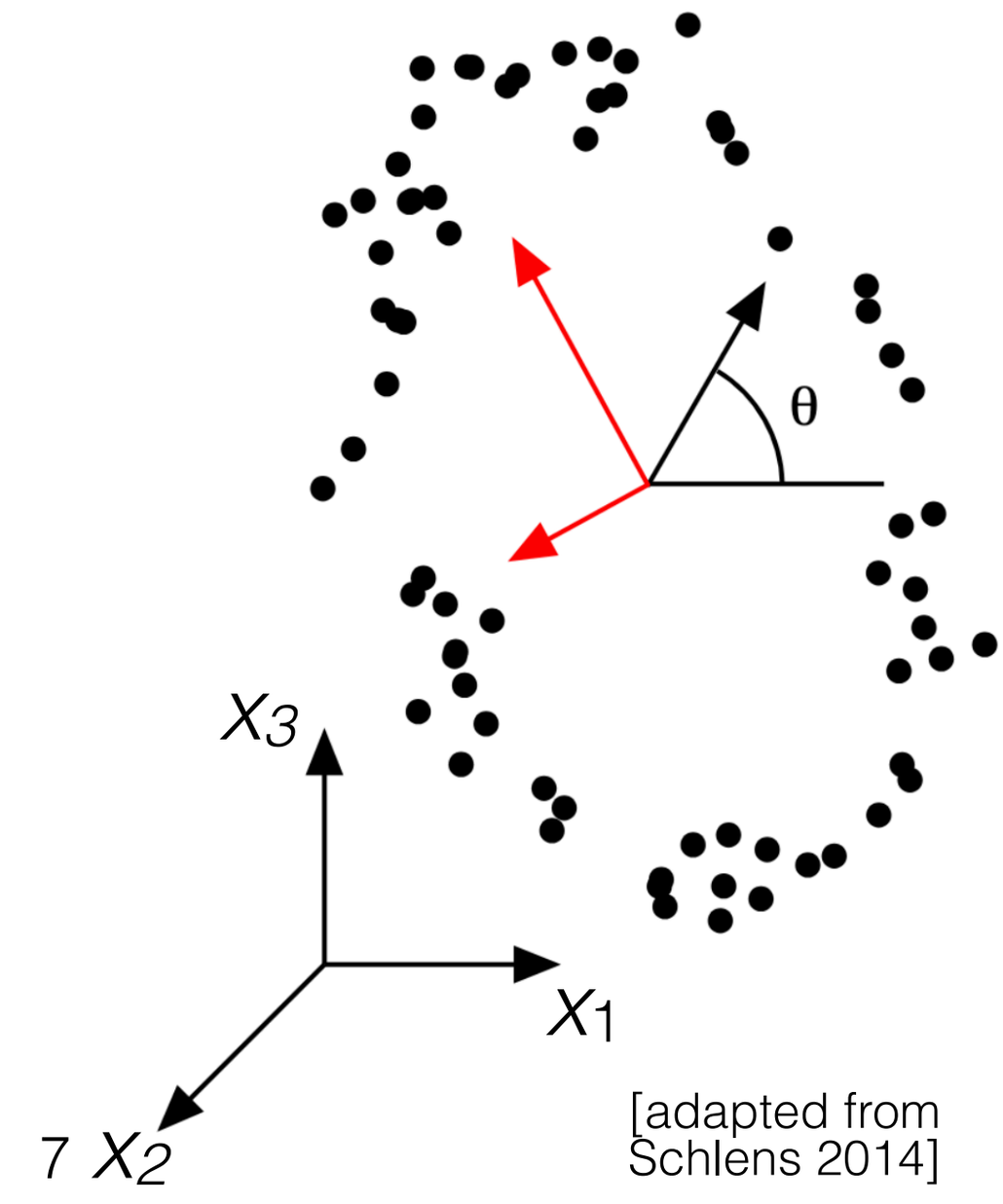
- PCA finds an orthogonal basis in the original feature space

Challenges of PCA

- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)

Challenges of PCA

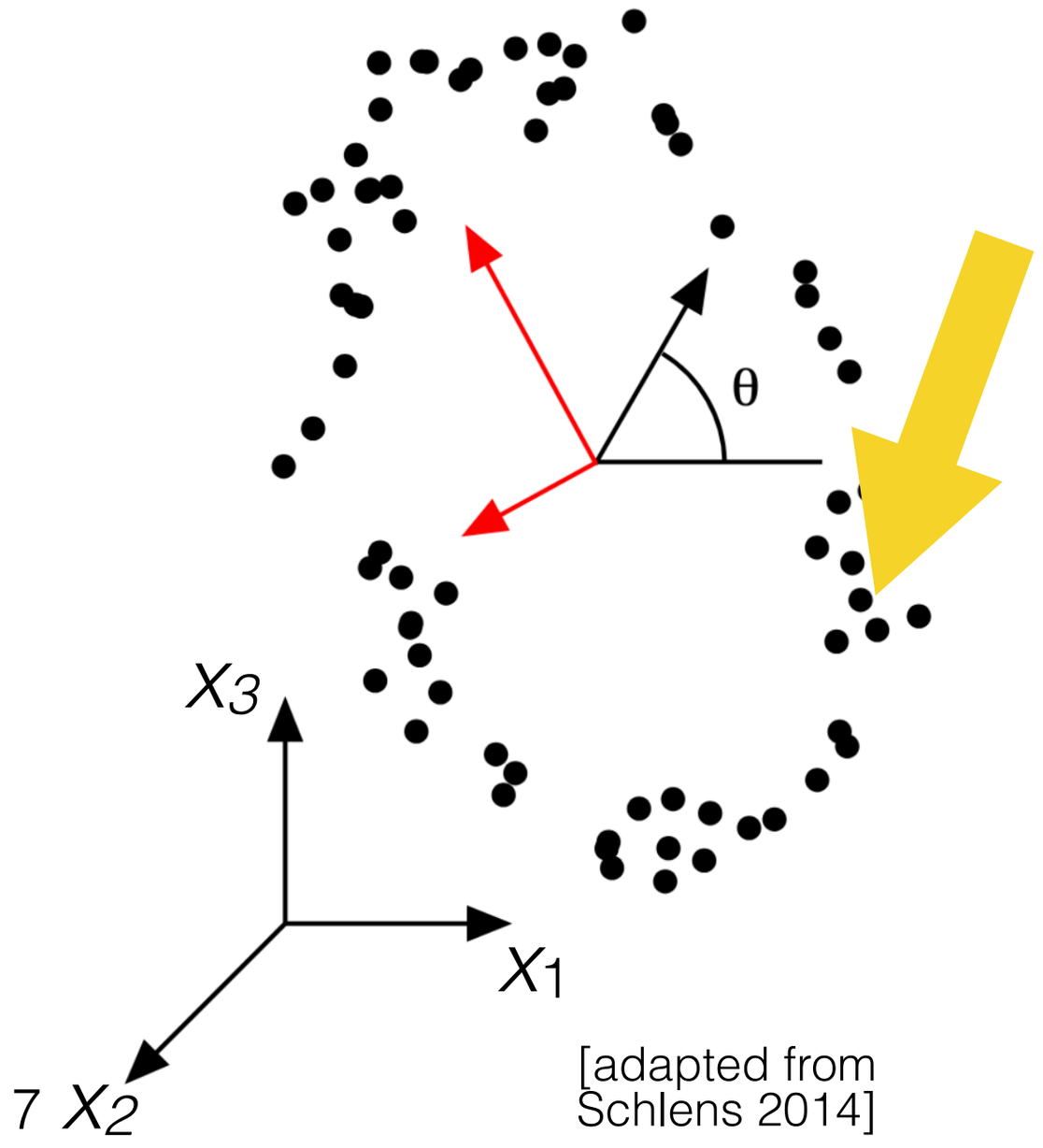
- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



[adapted from Schlens 2014]

Challenges of PCA

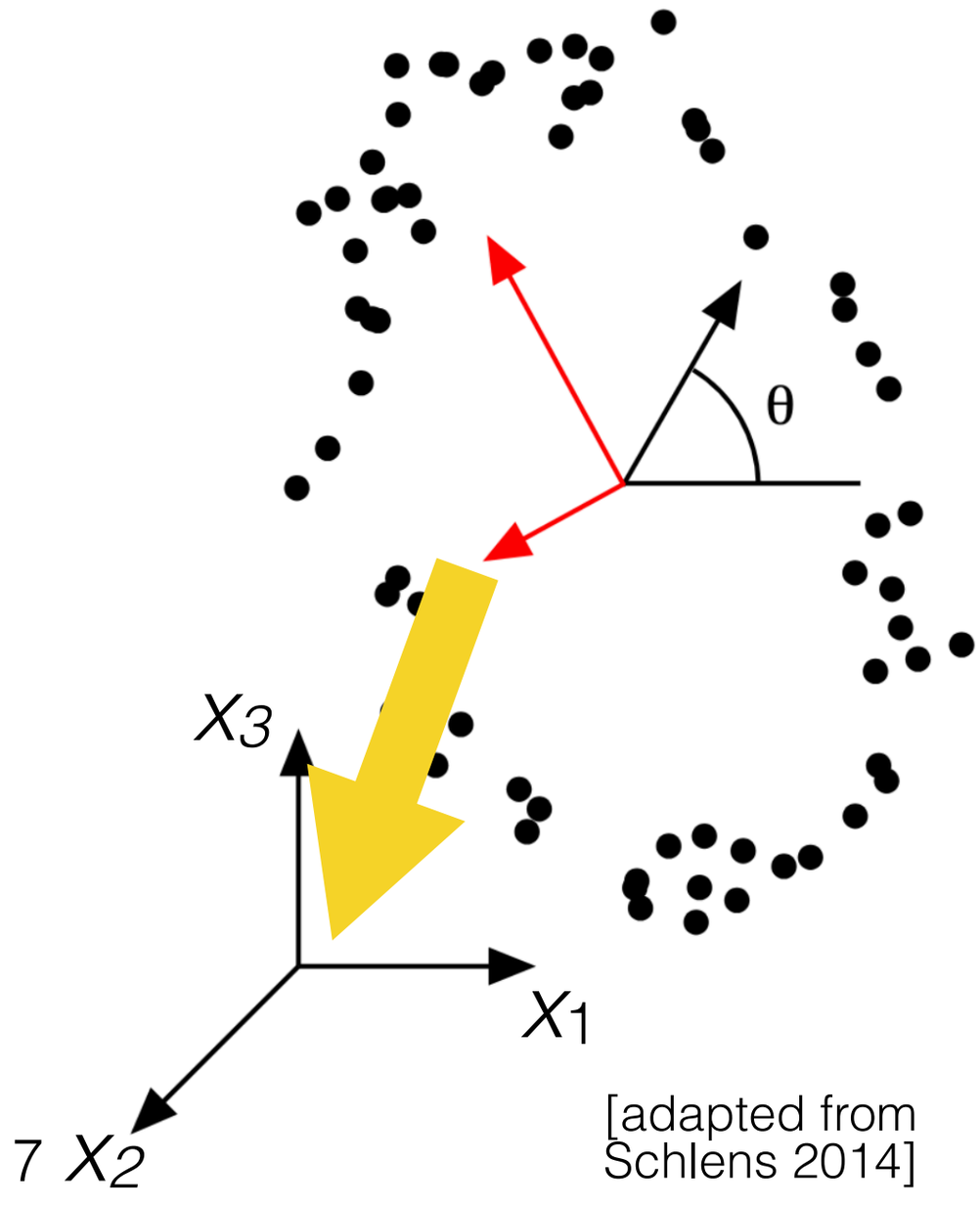
- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



[adapted from Schlens 2014]

Challenges of PCA

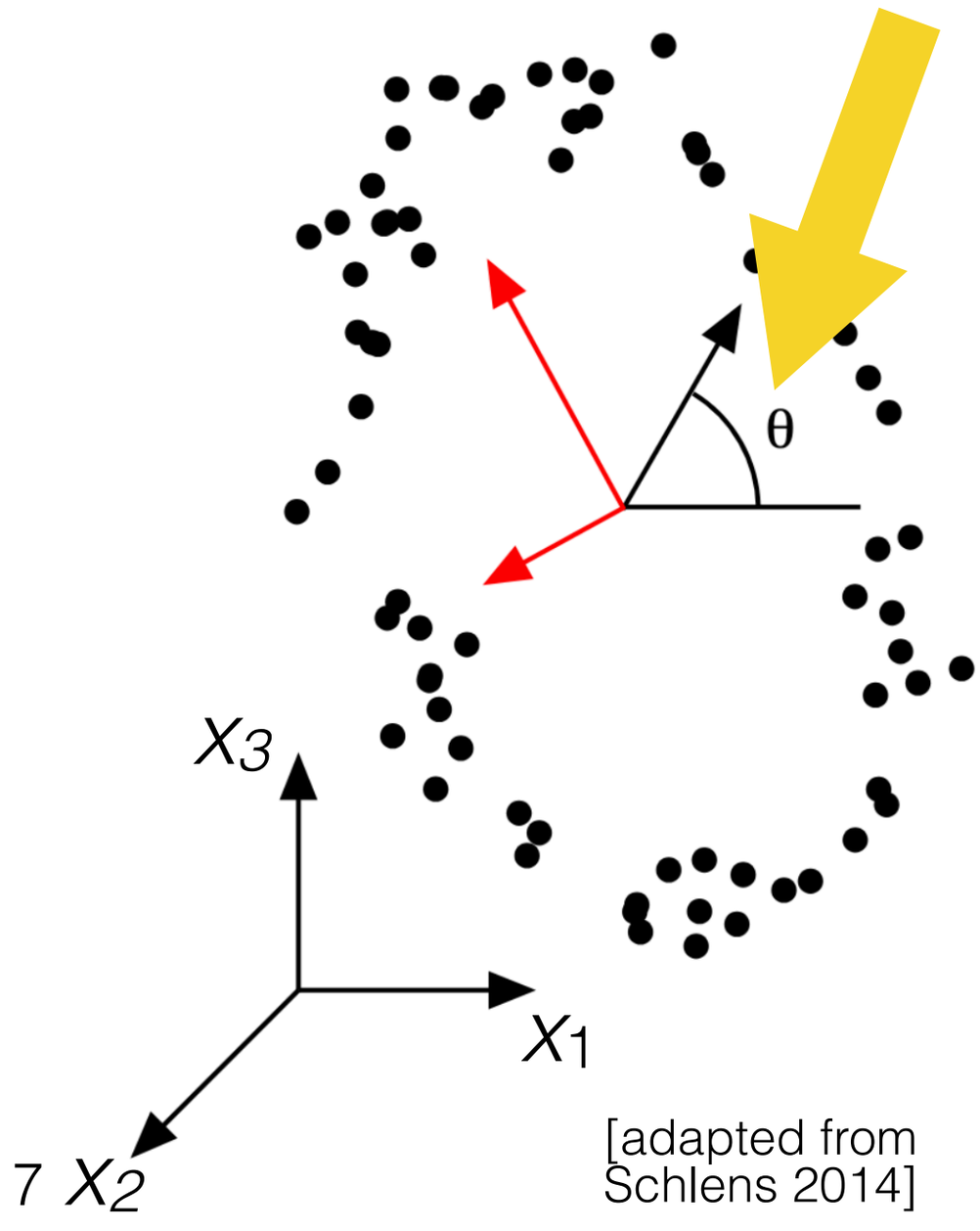
- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



[adapted from Schlens 2014]

Challenges of PCA

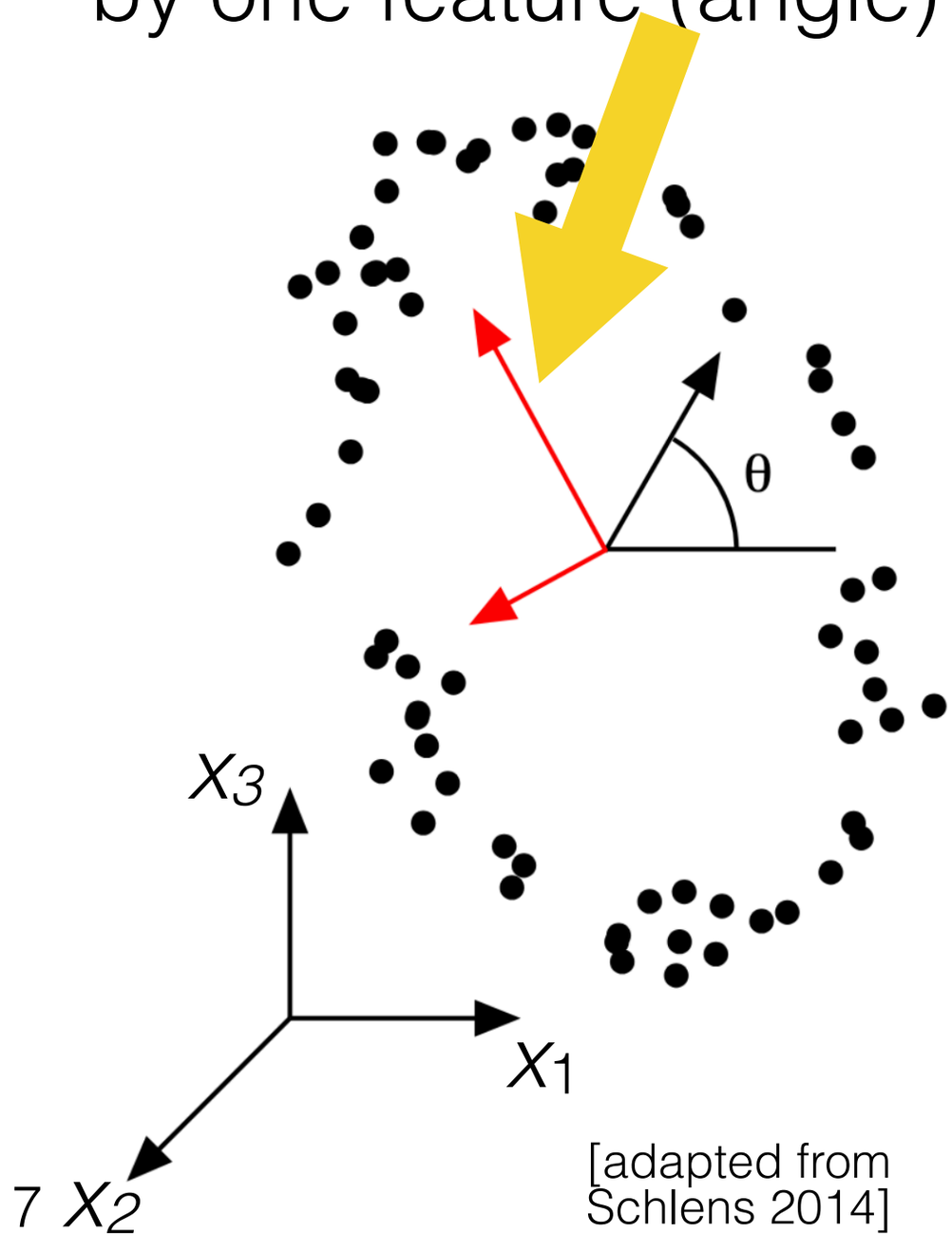
- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



[adapted from Schlens 2014]

Challenges of PCA

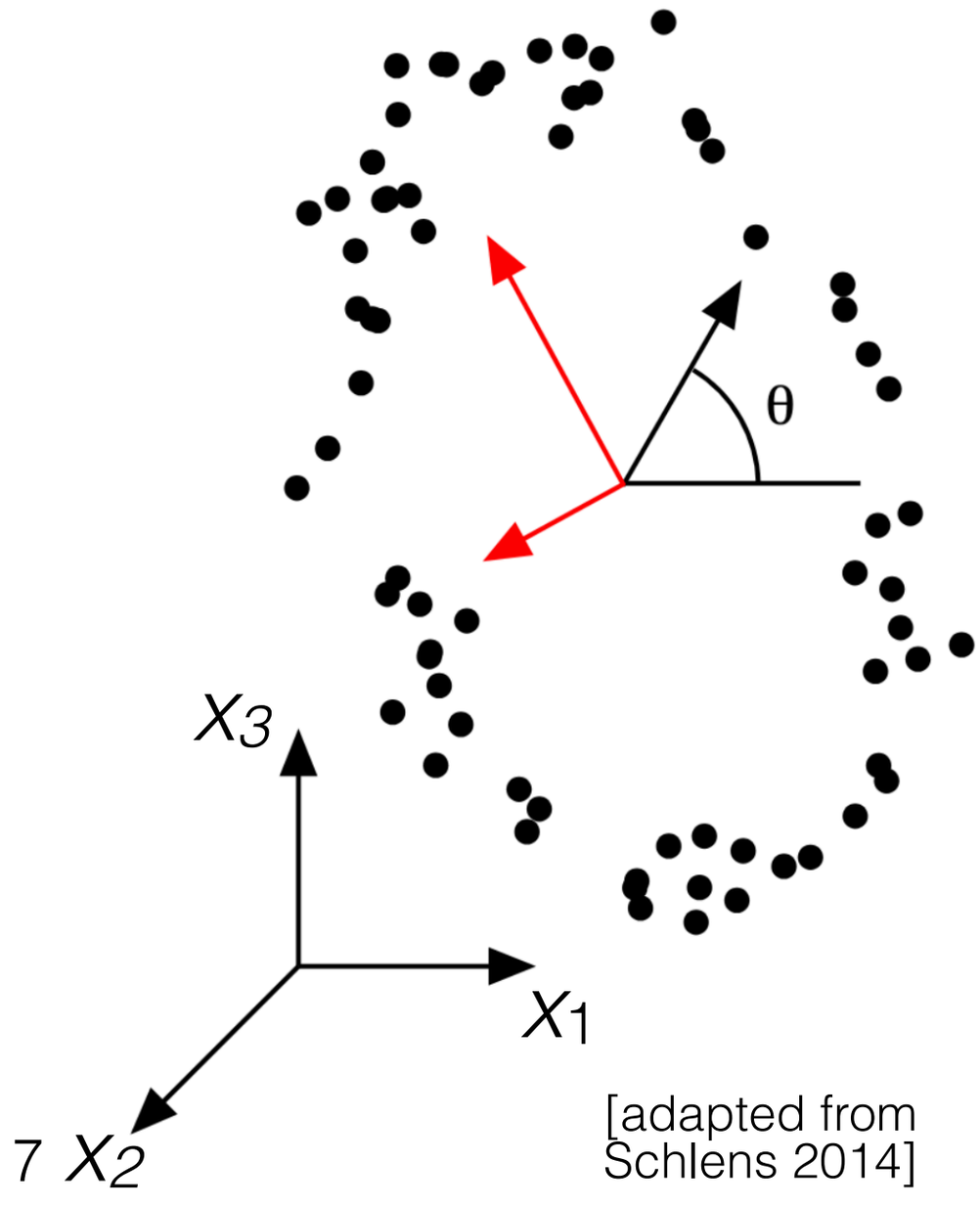
- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



[adapted from Schlens 2014]

Challenges of PCA

- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)

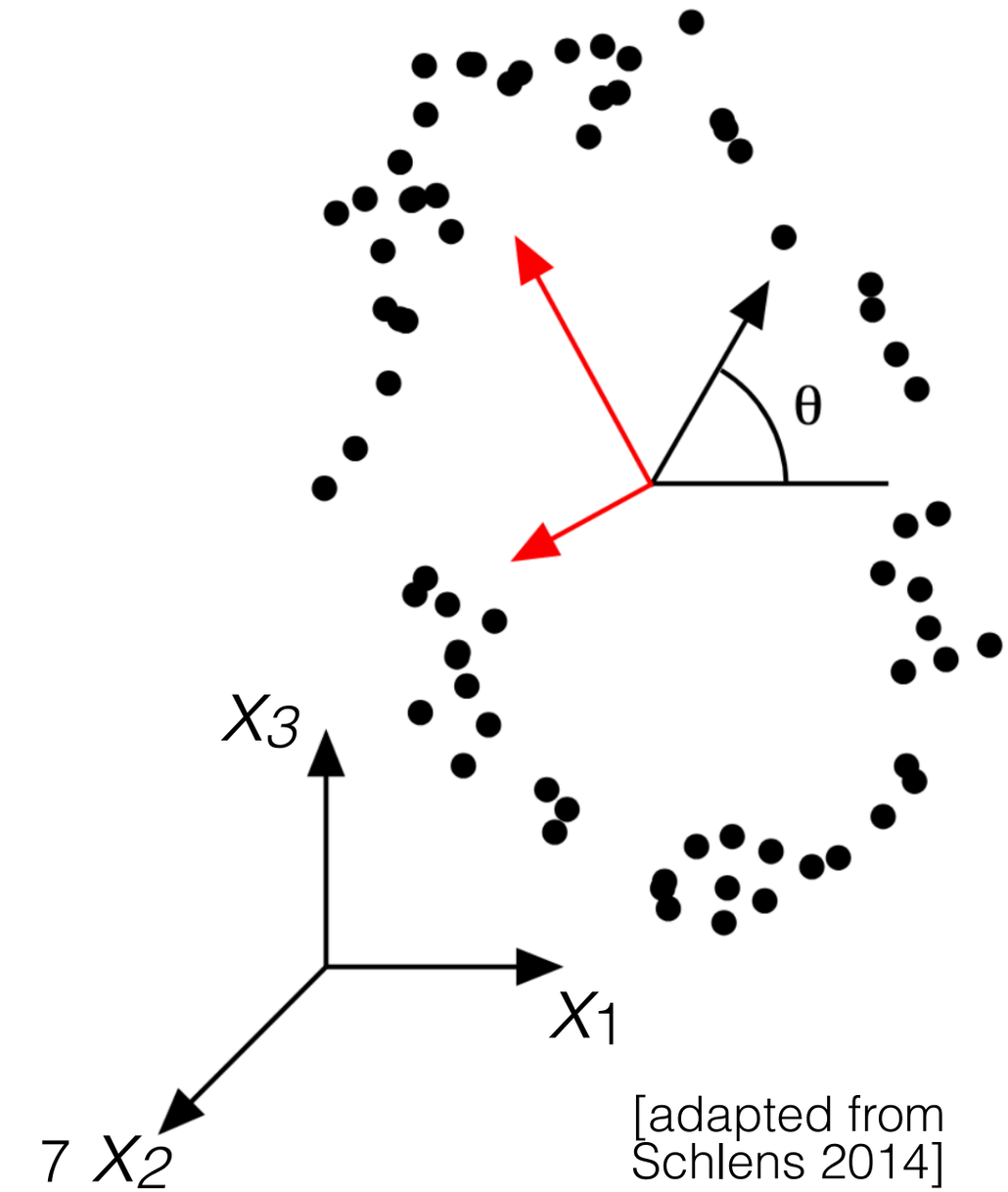


[adapted from Schlens 2014]

- Suppose all the activity in a town is on two main streets

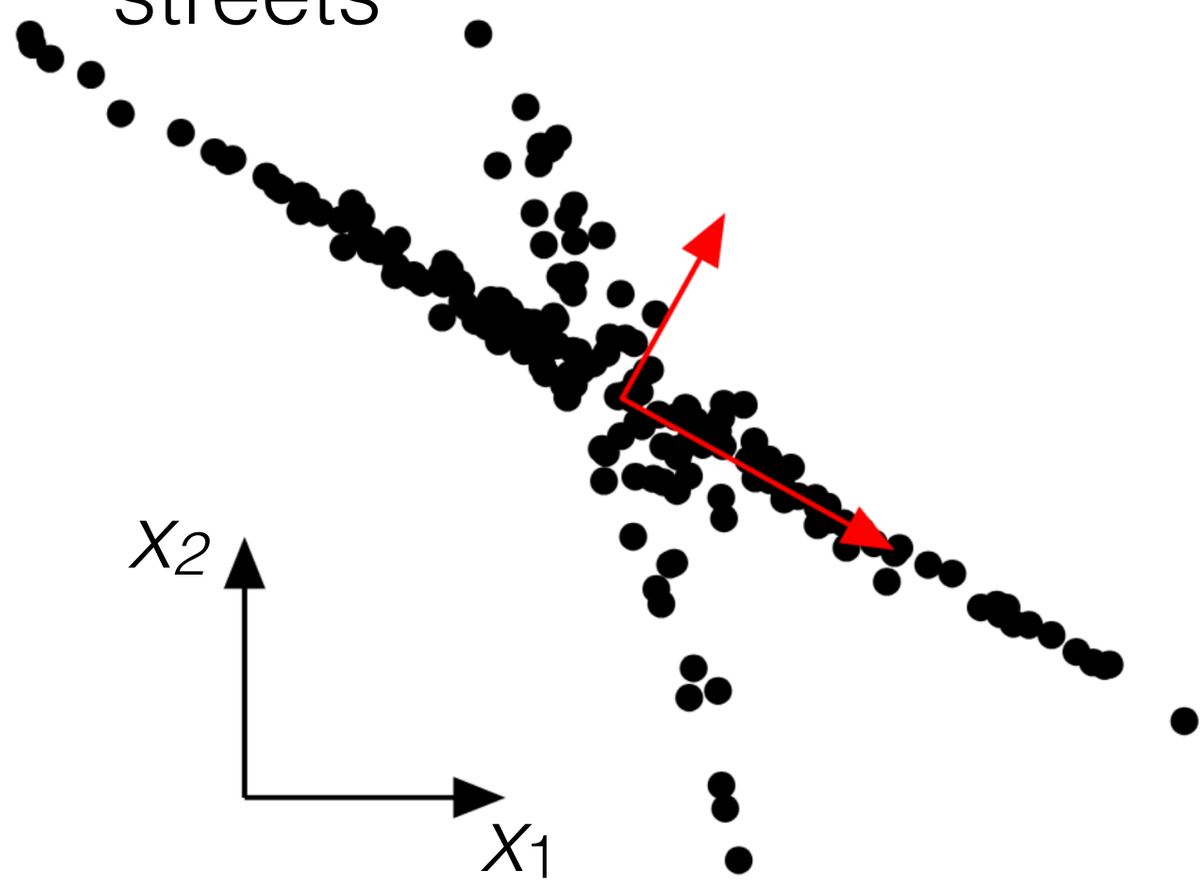
Challenges of PCA

- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



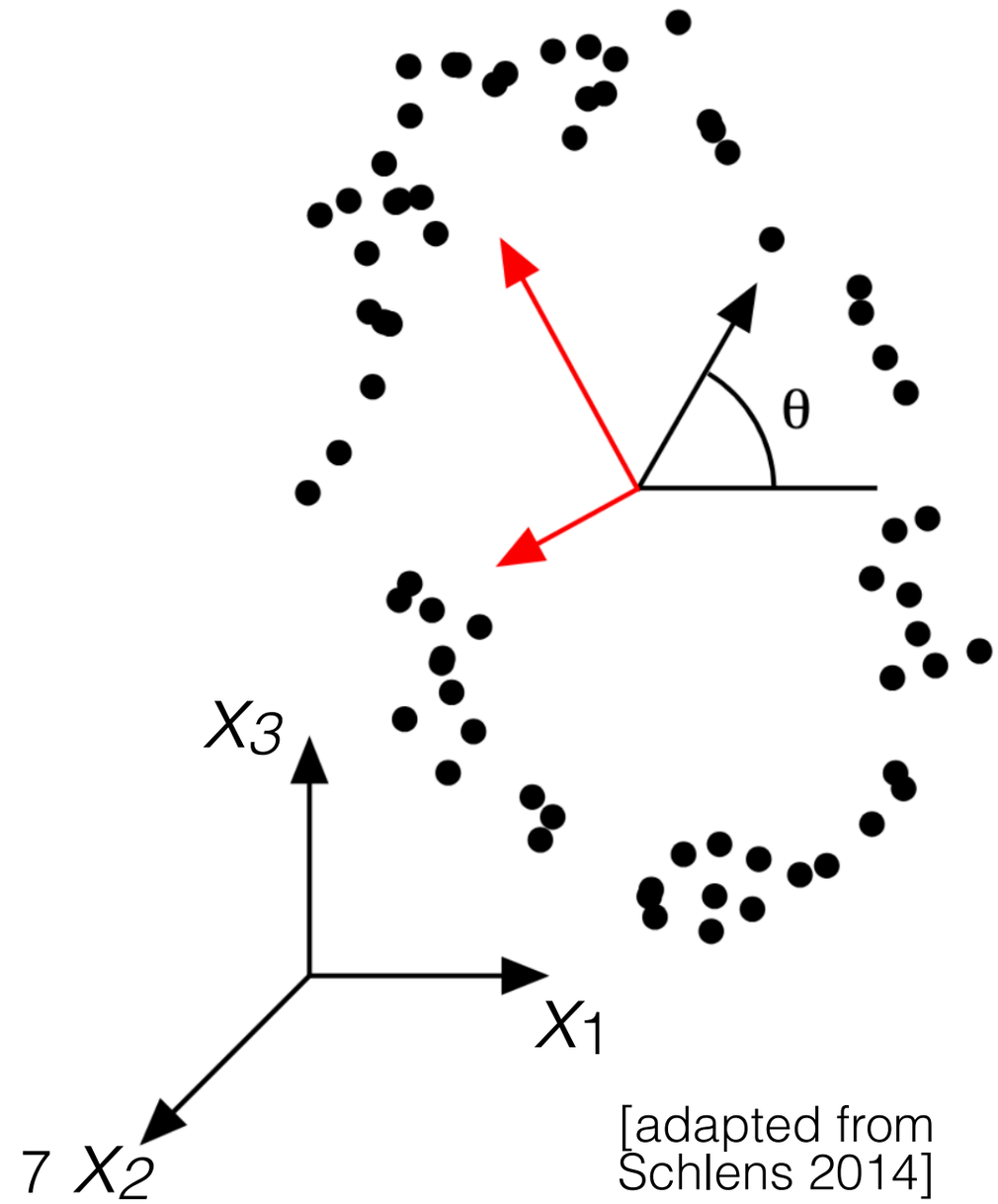
[adapted from Schlens 2014]

- Suppose all the activity in a town is on two main streets



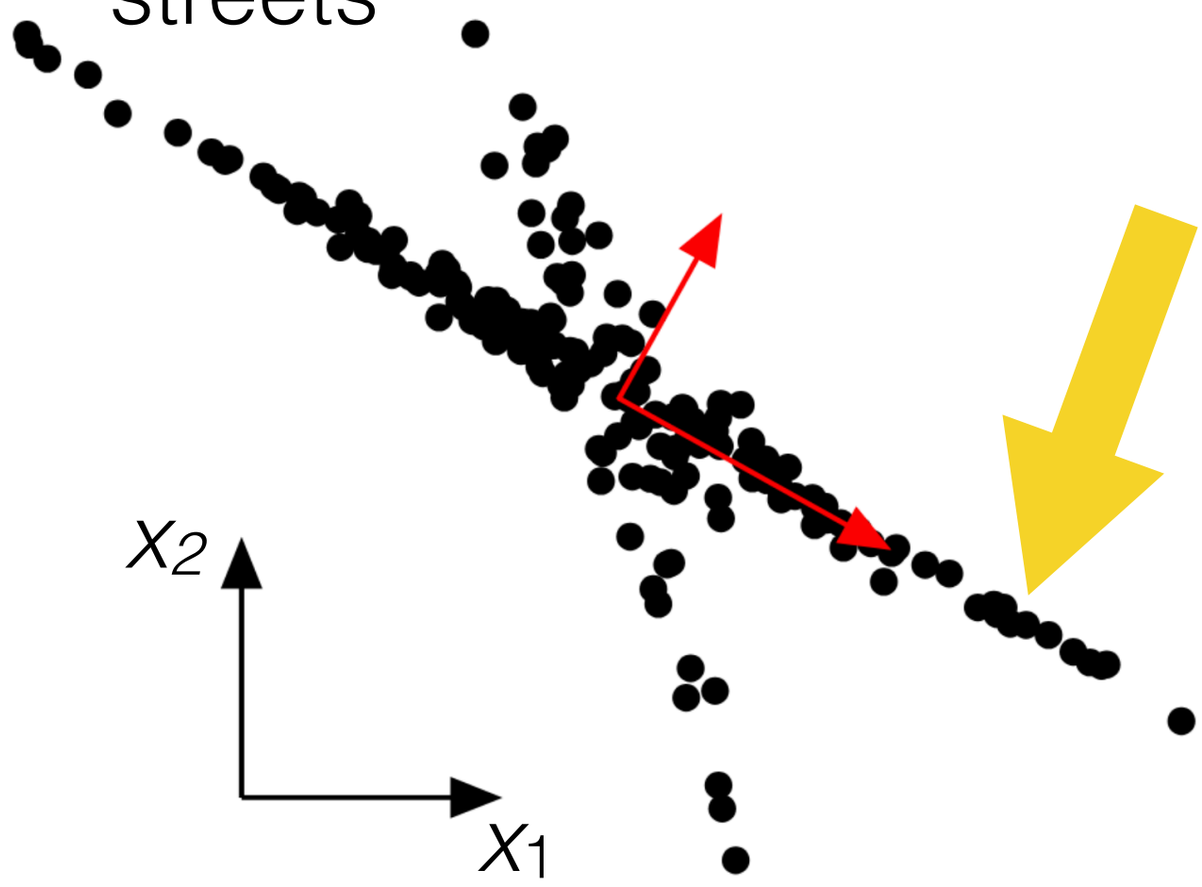
Challenges of PCA

- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



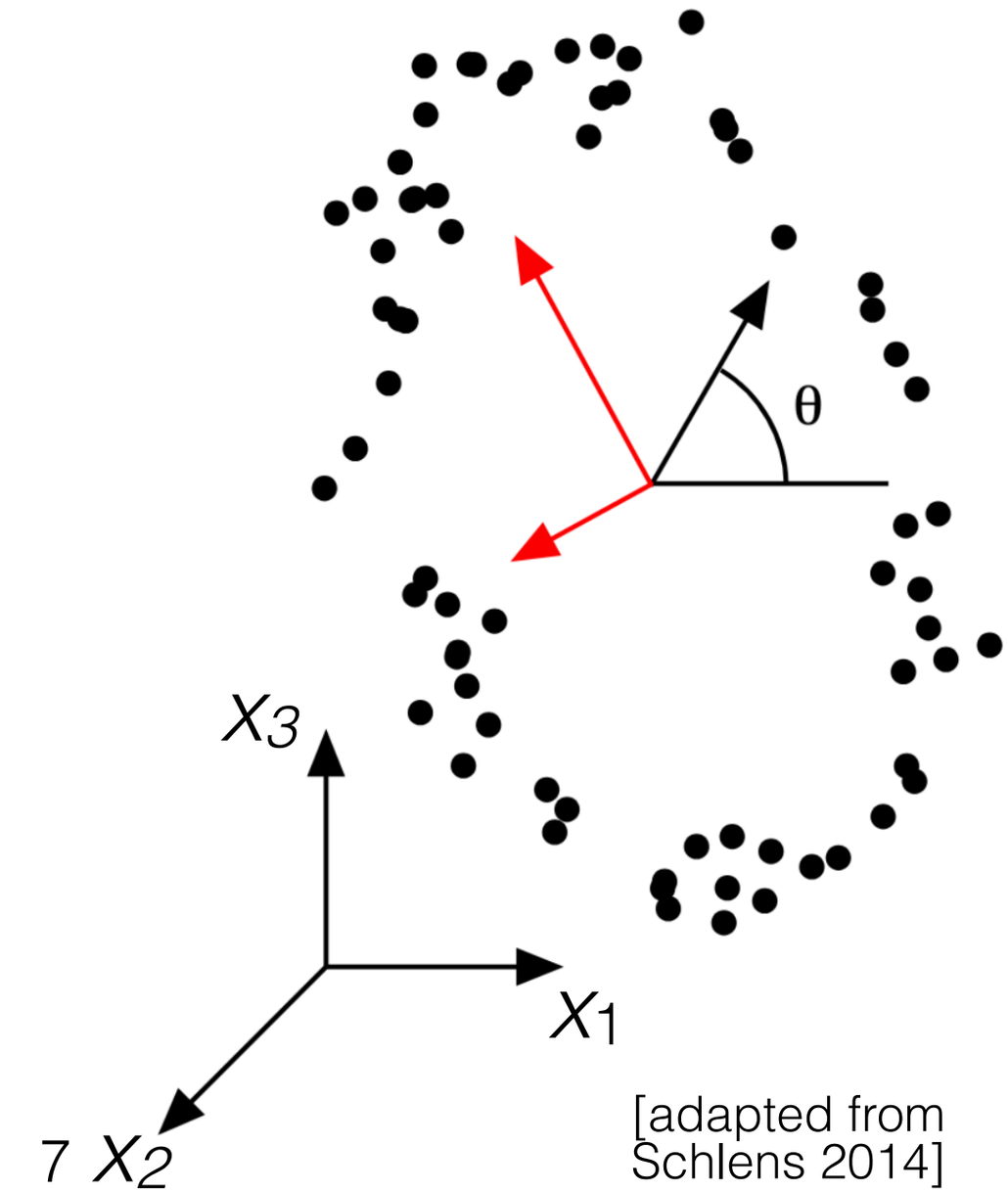
[adapted from Schlens 2014]

- Suppose all the activity in a town is on two main streets



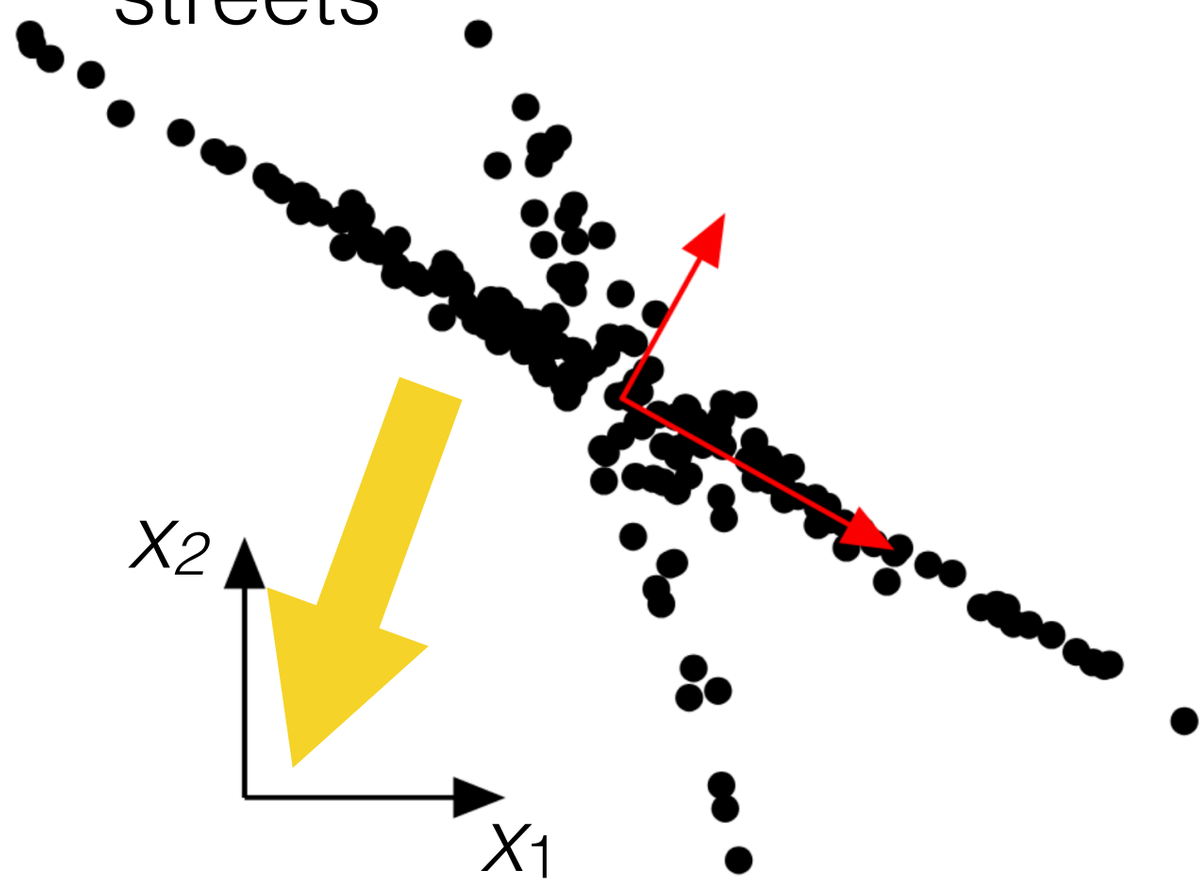
Challenges of PCA

- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



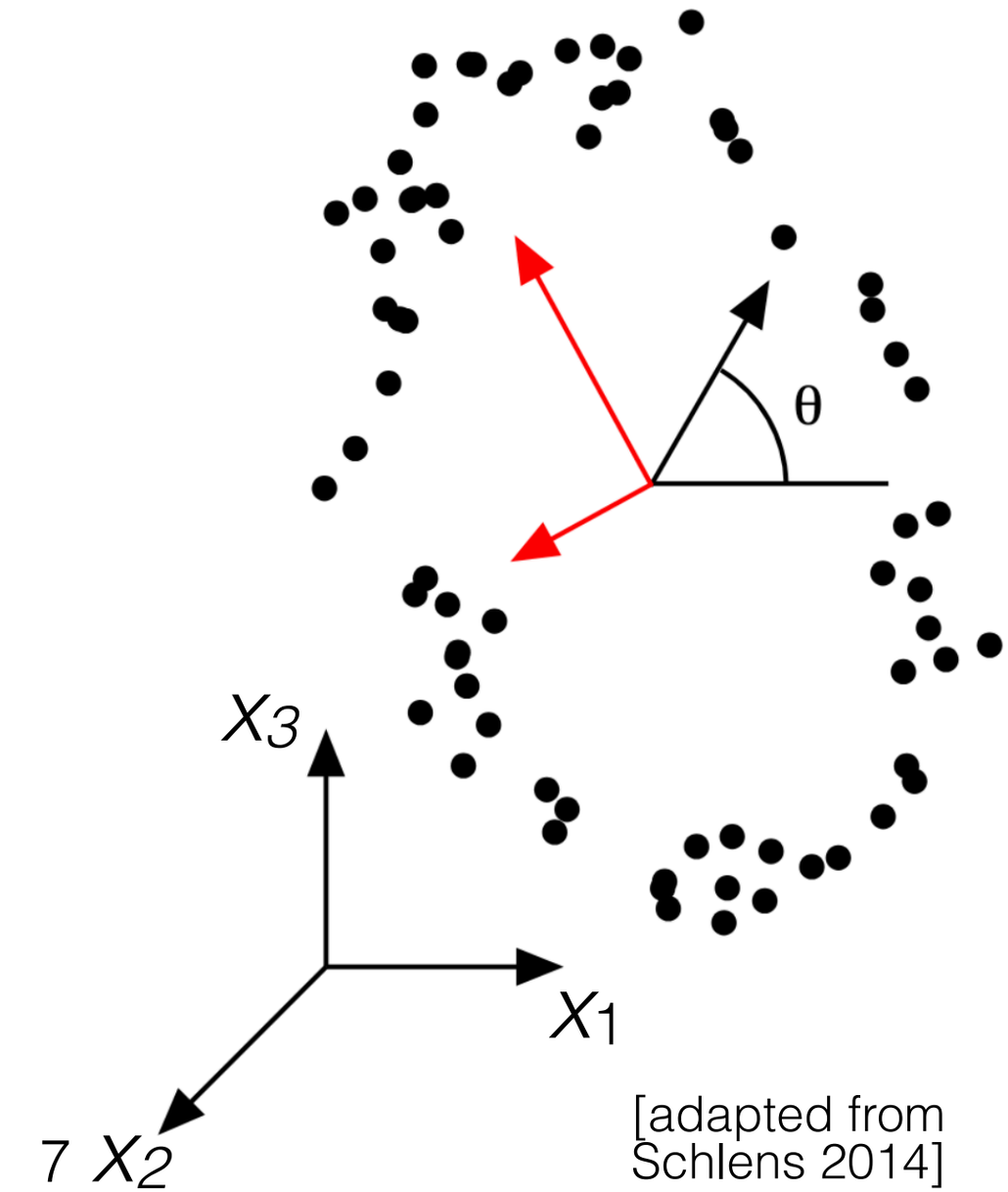
[adapted from Schlens 2014]

- Suppose all the activity in a town is on two main streets



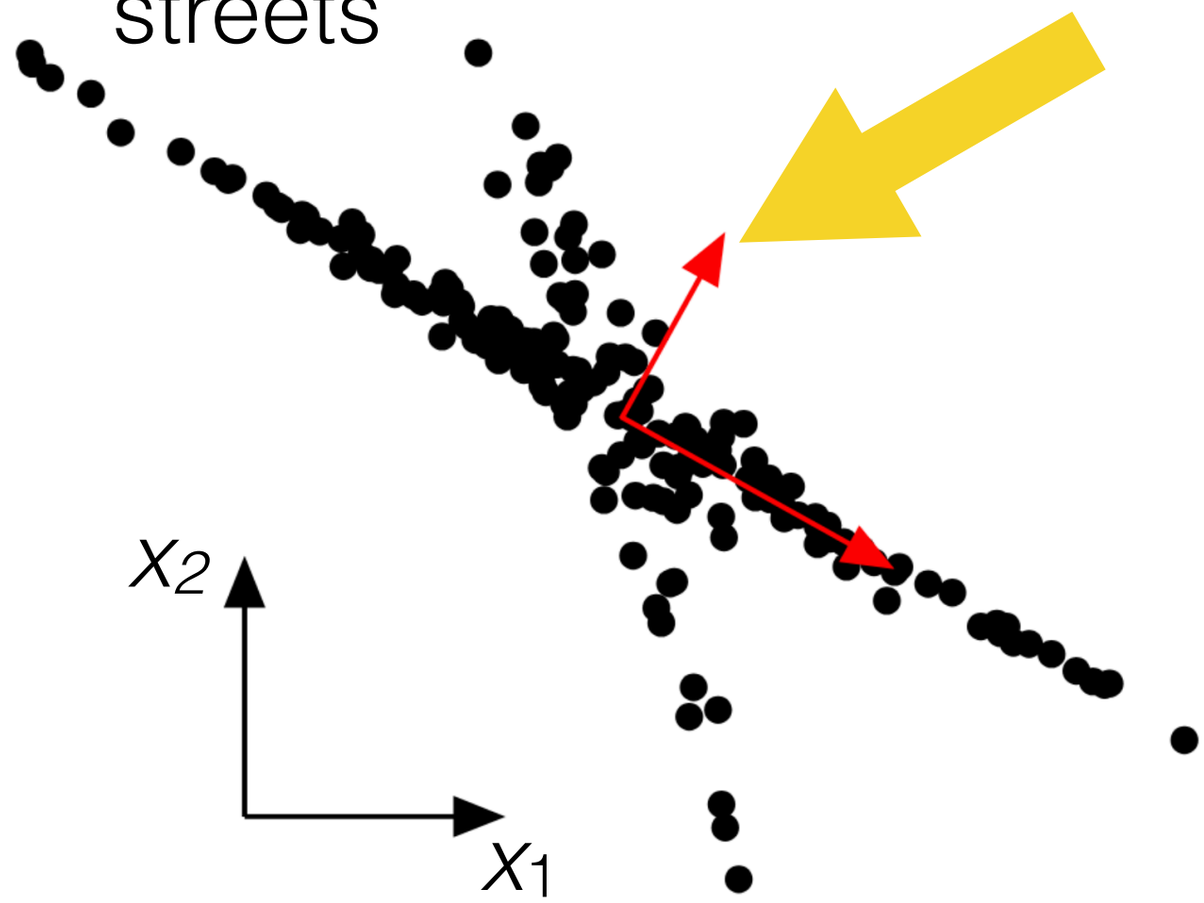
Challenges of PCA

- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



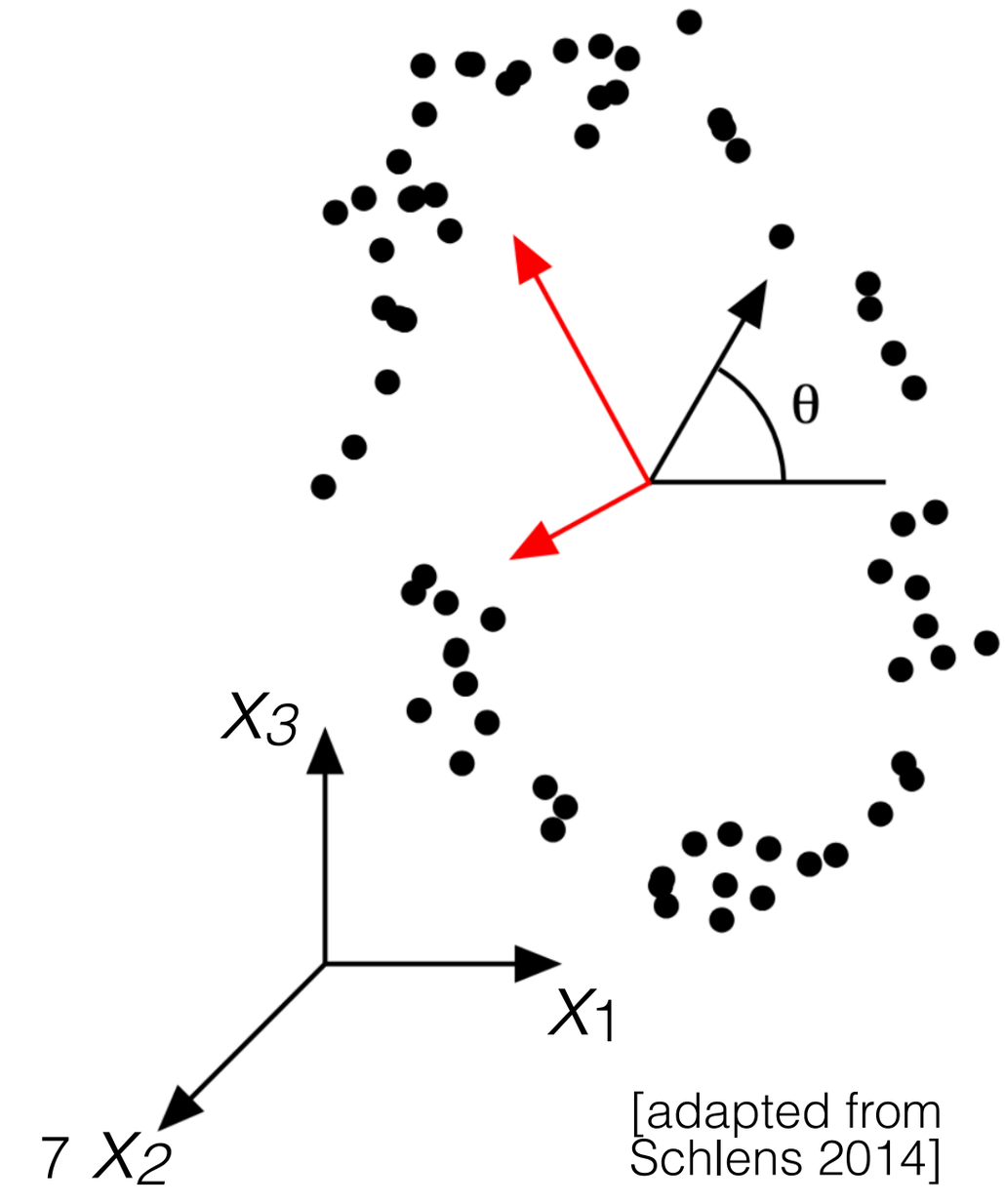
[adapted from Schlens 2014]

- Suppose all the activity in a town is on two main streets



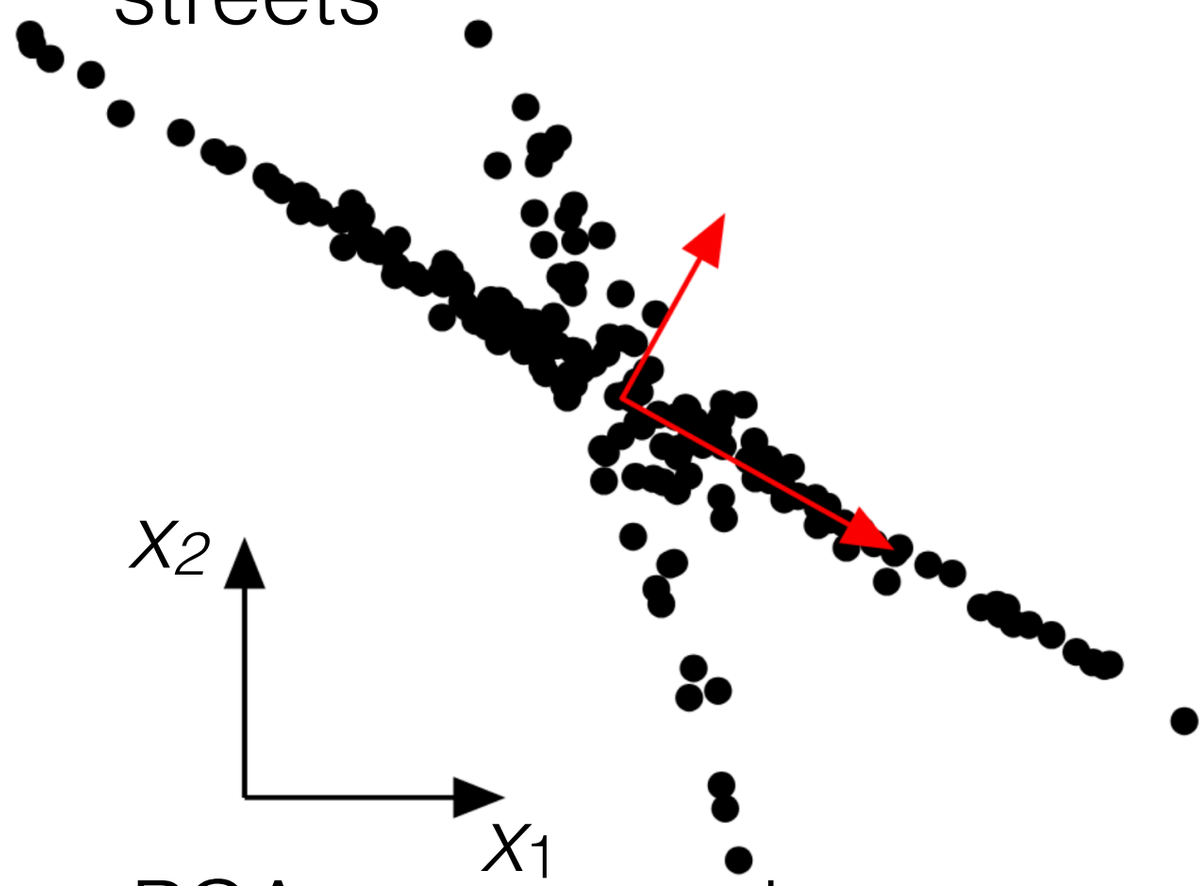
Challenges of PCA

- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



[adapted from Schlens 2014]

- Suppose all the activity in a town is on two main streets

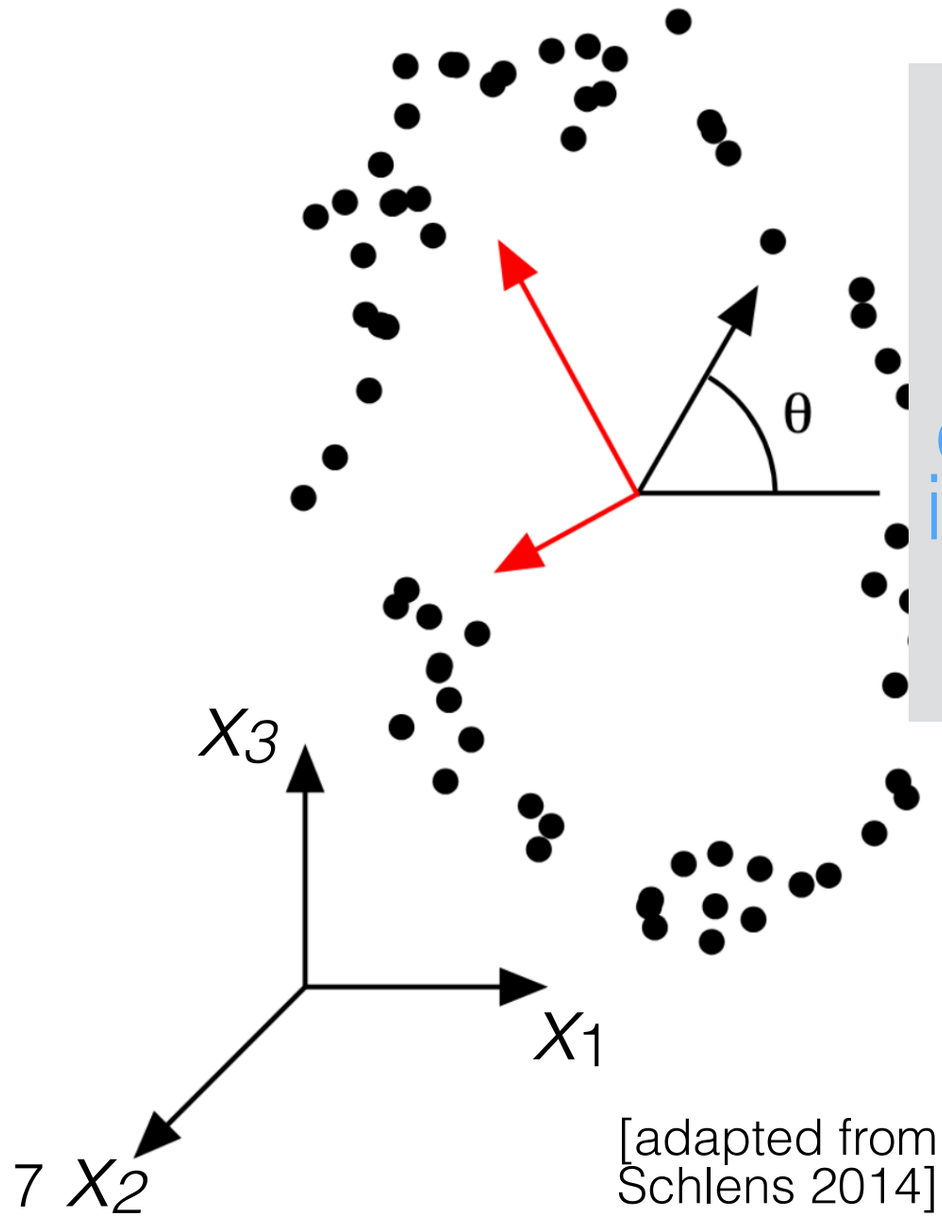


"[U]sing PCA we can make a more substantial normalization of the data to give it zero mean and unit covariance, so that different variables become decorrelated."

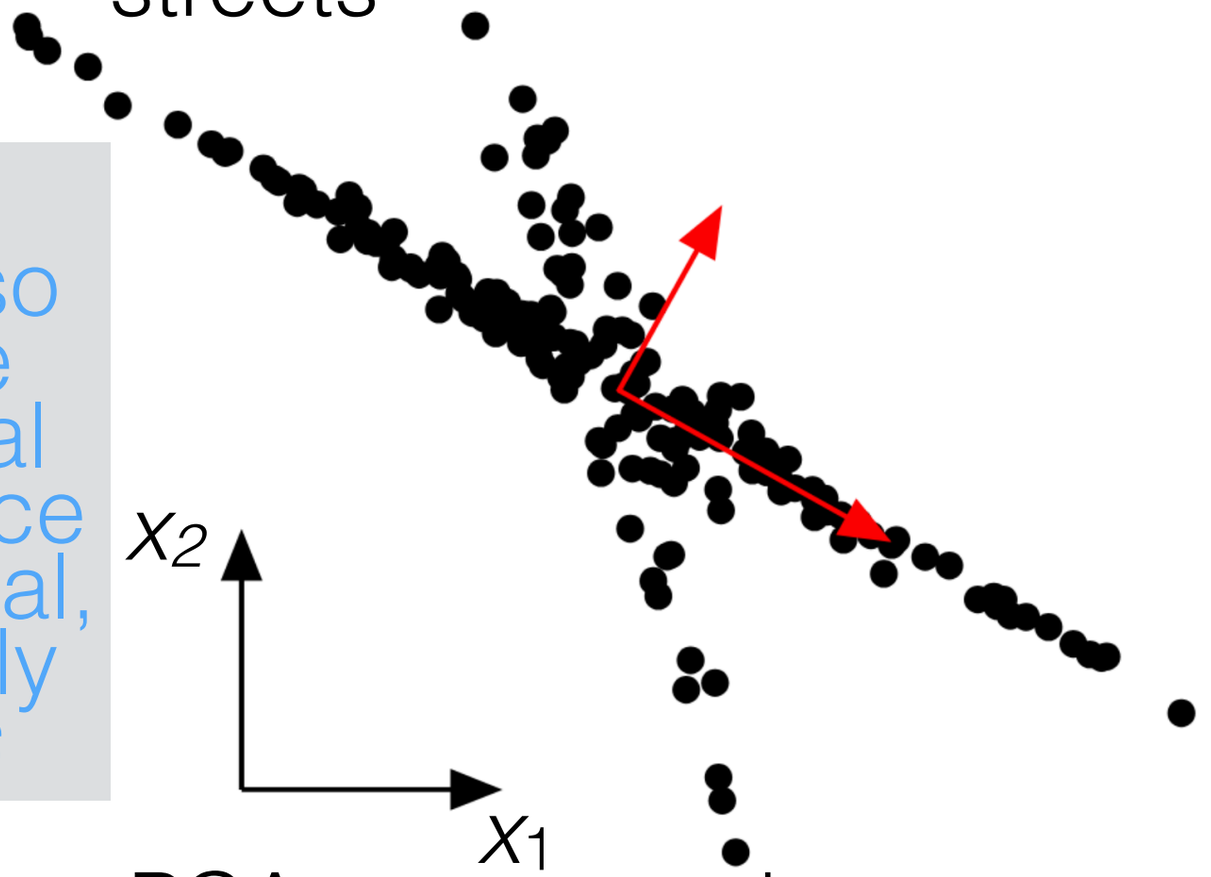
[Bishop 2006, 12.1]

Challenges of PCA

- PCA finds an orthogonal basis in the original feature space
- Motion of a car on a ferris wheel can be described by one feature (angle)



- Suppose all the activity in a town is on two main streets



PCA rotates so that the empirical covariance is diagonal, but it only *rotates*

[adapted from Schlens 2014]

"[U]sing PCA we can make a more substantial normalization of the data to give it zero mean and unit covariance, so that different variables become decorrelated."

[Bishop 2006, 12.1]

Another option: t-SNE

- t-distributed stochastic neighbor embedding (t-SNE)

Another option: t-SNE

- t-distributed stochastic neighbor embedding (t-SNE)
- Non-linear (vs. linear PCA)

Another option: t-SNE

- t-distributed stochastic neighbor embedding (t-SNE)
- Non-linear (vs. linear PCA)
- Designed for visualization (mapping to 2D, cf. other goals)

Another option: t-SNE

- t-distributed stochastic neighbor embedding (t-SNE)
- Non-linear (vs. linear PCA)
- Designed for visualization (mapping to 2D, cf. other goals)
- Local rather
  than global

Another option: t-SNE

- t-distributed stochastic neighbor embedding (t-SNE)
- Non-linear (vs. linear PCA)
- Designed for visualization (mapping to 2D, cf. other goals)
- Local rather than global
  - PCA gives you a map based on all training points

Another option: t-SNE

- t-distributed stochastic neighbor embedding (t-SNE)
- Non-linear (vs. linear PCA)
- Designed for visualization (mapping to 2D, cf. other goals)
- Local rather than global
  - PCA gives you a map based on all training points
- PCA's map applies to new points

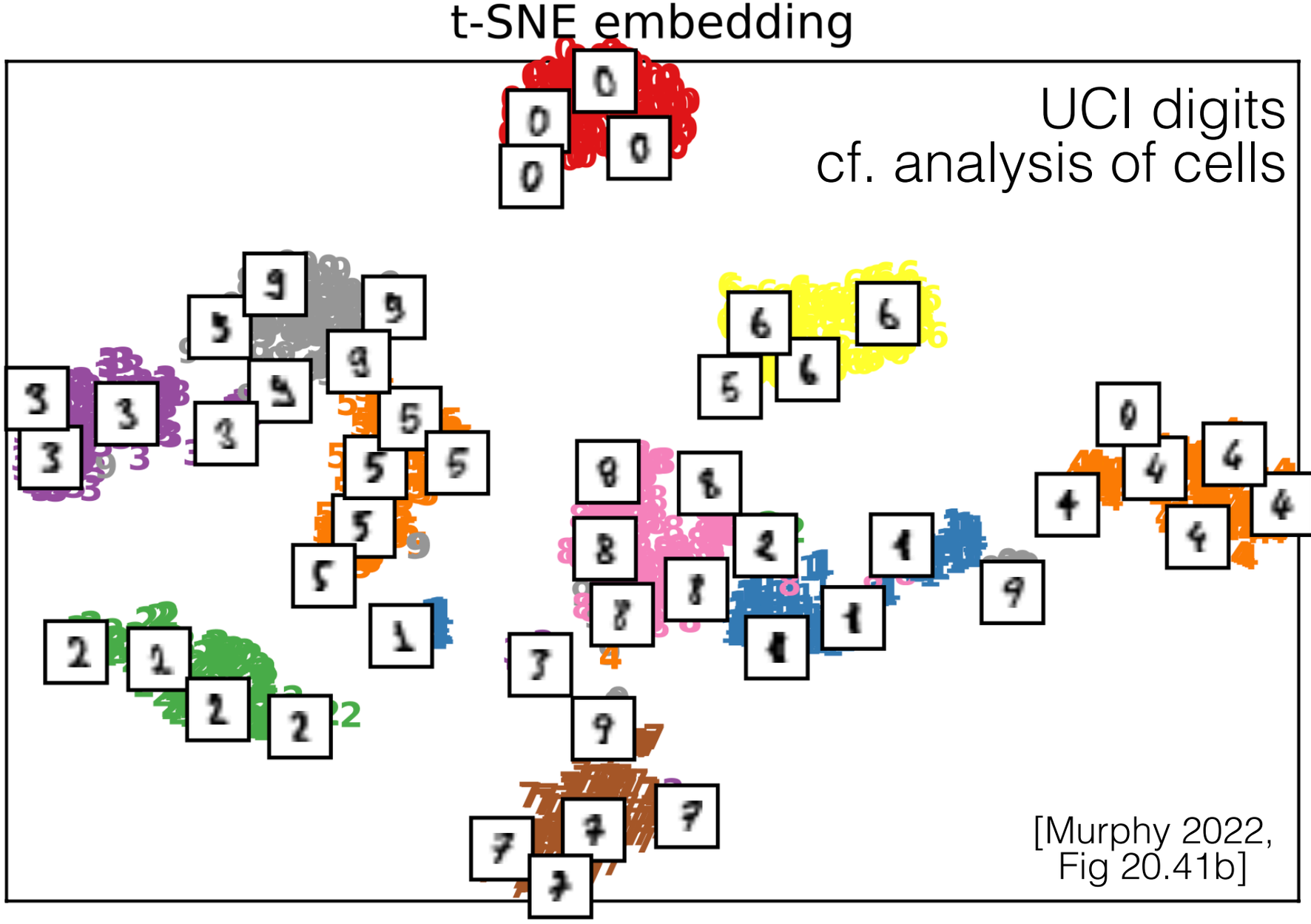
Another option: t-SNE

- t-distributed stochastic neighbor embedding (t-SNE)
- Non-linear (vs. linear PCA)
- Designed for visualization (mapping to 2D, cf. other goals)
- Local rather than global
  - PCA gives you a map based on all training points
- PCA's map applies to new points

UCI digits
cf. analysis of cells

Another option: t-SNE

- t-distributed stochastic neighbor embedding (t-SNE)
- Non-linear (vs. linear PCA)
- Designed for visualization (mapping to 2D, cf. other goals)
- Local rather than global
  - PCA gives you a map based on all training points
- PCA's map applies to new points



SNE & t-SNE

SNE & t-SNE

- Keep distances from original space in 2D?

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D
  - Similarity of each point to *n* in the original space, as a probability distribution. (Note: hyperparam for each *n*)

$$p_{m|n} := \frac{\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(n')}\|^2)}$$

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D
  - Similarity of each point to *n* in the original space, as a probability distribution. (Note: hyperparam for each *n*)

$$p_{m|n} := \frac{\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(n')}\|^2)}$$

  - Similarity of each point to *n* in 2D space (no hyperparam)

$$q_{m|n} := \frac{\exp(-\|z^{(n)} - z^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\|z^{(n)} - z^{(n')}\|^2)}$$

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D
  - Similarity of each point to *n* in the original space, as a probability distribution. (Note: hyperparam for each *n*)

$$p_{m|n} := \frac{\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(n')}\|^2)}$$

  - Similarity of each point to *n* in 2D space (no hyperparam)

$$q_{m|n} := \frac{\exp(-\|z^{(n)} - z^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\|z^{(n)} - z^{(n')}\|^2)}$$

  - Make all *N* distribs close: $\min_{\{z_n\}_{n=1}^N}\sum_{n=1}^N D_{KL}(p_{\cdot|n}\|q_{\cdot|n})$

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D
  - Similarity of each point to *n* in the original space, as a probability distribution. (Note: hyperparam for each *n*)

$$p_{m|n} := \frac{\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(n')}\|^2)}$$

  - Similarity of each point to *n* in 2D space (no hyperparam)

$$q_{m|n} := \frac{\exp(-\|z^{(n)} - z^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\|z^{(n)} - z^{(n')}\|^2)}$$

  - Make all *N* distribs close: $\min_{\{z_n\}_{n=1}^N}\sum_{n=1}^N D_{KL}(p_{\cdot|n}\|q_{\cdot|n})$
  - Kullback-Leibler divergence: asymmetric
    $D_{KL}(p_{\cdot|n}\|q_{\cdot|n}) = \sum_{m:m\neq n} p_{m|n}\log\frac{p_{m|n}}{q_{m|n}}$

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D
  - Similarity of each point to *n* in the original space, as a probability distribution. (Note: hyperparam for each *n*)

$$p_{m|n} := \frac{\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(n')}\|^2)}$$

  - Similarity of each point to *n* in 2D space (no hyperparam)

$$q_{m|n} := \frac{\exp(-\|z^{(n)} - z^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\|z^{(n)} - z^{(n')}\|^2)}$$

  - Make all *N* distribs close: $\min_{\{z_n\}_{n=1}^N}\sum_{n=1}^N D_{KL}(p_{\cdot|n}\|q_{\cdot|n})$
  - Kullback-Leibler divergence: asymmetric $D_{KL}(p_{\cdot|n}\|q_{\cdot|n}) = \sum_{m:m\neq n} p_{m|n}\log\frac{p_{m|n}}{q_{m|n}}$

when *p* large, need *q* large

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D
  - Similarity of each point to *n* in the original space, as a probability distribution. (Note: hyperparam for each *n*)

$$p_{m|n} := \frac{\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(n')}\|^2)}$$

  - Similarity of each point to *n* in 2D space (no hyperparam)

$$q_{m|n} := \frac{\exp(-\|z^{(n)} - z^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\|z^{(n)} - z^{(n')}\|^2)}$$

  - Make all *N* distribs close: $\min_{\{z_n\}_{n=1}^N}\sum_{n=1}^N D_{KL}(p_{\cdot|n}\|q_{\cdot|n})$
  - Kullback-Leibler divergence: asymmetric $D_{KL}(p_{\cdot|n}\|q_{\cdot|n}) = \sum_{m:m\neq n} p_{m|n}\log\frac{p_{m|n}}{q_{m|n}}$ when *p* large, need *q* large
  - Optimization is definitely not convex: practical concern

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D
  - Similarity of each point to *n* in the original space, as a probability distribution. (Note: hyperparam for each *n*)

$$p_{m|n} := \frac{\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(n')}\|^2)}$$

  - Similarity of each point to *n* in 2D space (no hyperparam)

$$q_{m|n} := \frac{\exp(-\|z^{(n)} - z^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\|z^{(n)} - z^{(n')}\|^2)}$$

  - Make all *N* distribs close: $\min_{\{z_n\}_{n=1}^N}\sum_{n=1}^N D_{KL}(p_{\cdot|n}\|q_{\cdot|n})$
  - Kullback-Leibler divergence: asymmetric $D_{KL}(p_{\cdot|n}\|q_{\cdot|n}) = \sum_{m:m\neq n} p_{m|n}\log\frac{p_{m|n}}{q_{m|n}}$ — when *p* large, need *q* large
  - Optimization is definitely not convex: practical concern
  - User sets "perplexity" (~controls # neighbors) $\Rightarrow \{\sigma_n^2\}_{n=1}^N$

SNE & t-SNE

- Keep distances from original space in 2D? Problem: everything far in high D
- SNE: try to keep *neighbors* from high dimensions in 2D
  - Similarity of each point to *n* in the original space, as a probability distribution. (Note: hyperparam for each *n*)

$$p_{m|n} := \frac{\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\frac{1}{2\sigma_n^2}\|x^{(n)} - x^{(n')}\|^2)}$$

  - Similarity of each point to *n* in 2D space (no hyperparam)

$$q_{m|n} := \frac{\exp(-\|z^{(n)} - z^{(m)}\|^2)}{\sum_{n':n'\neq n}\exp(-\|z^{(n)} - z^{(n')}\|^2)}$$

  - Make all *N* distribs close: $\min_{\{z_n\}_{n=1}^N}\sum_{n=1}^N D_{KL}(p_{\cdot|n}\|q_{\cdot|n})$
  - Kullback-Leibler divergence: asymmetric $D_{KL}(p_{\cdot|n}\|q_{\cdot|n}) = \sum_{m:m\neq n} p_{m|n}\log\frac{p_{m|n}}{q_{m|n}}$ when *p* large, need *q* large
  - Optimization is definitely not convex: practical concern
  - User sets "perplexity" (~controls # neighbors) $\Rightarrow \{\sigma_n^2\}_{n=1}^N$
- t-SNE: Practical behavior is better with a heavier-tailed distribution (student's-t/Cauchy instead of Gaussian)

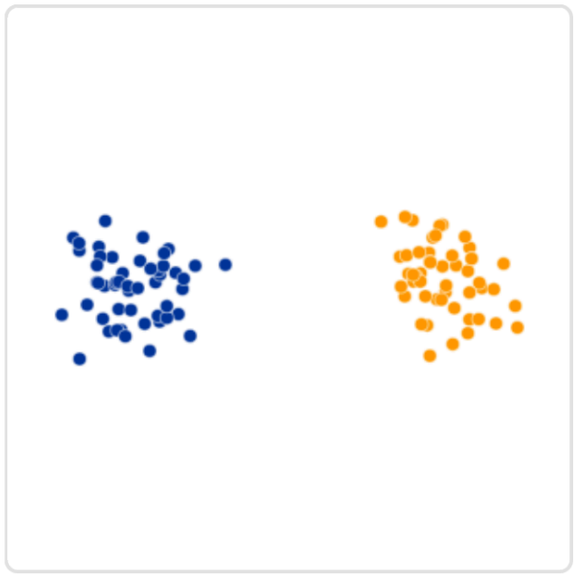
Challenges of t-SNE

- The choice of perplexity matters

[Largely borrowed from Wattenberg et al 2016 "How to use t-SNE effectively." Check it out!]

Challenges of t-SNE

- The choice of perplexity matters
- Data 2D here



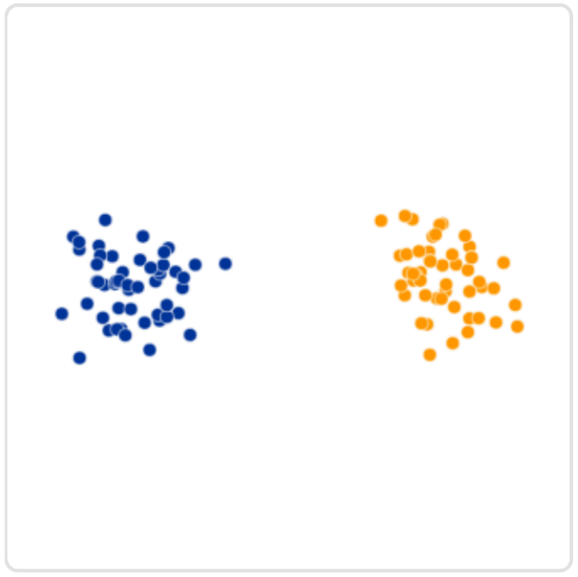
Original

[Largely borrowed from Wattenberg et al 2016 "How to use t-SNE effectively." Check it out!]

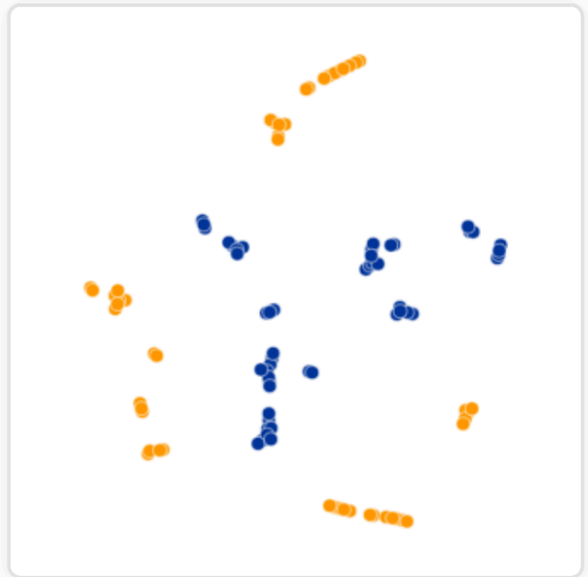
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 "How to use t-SNE effectively." Check it out!]

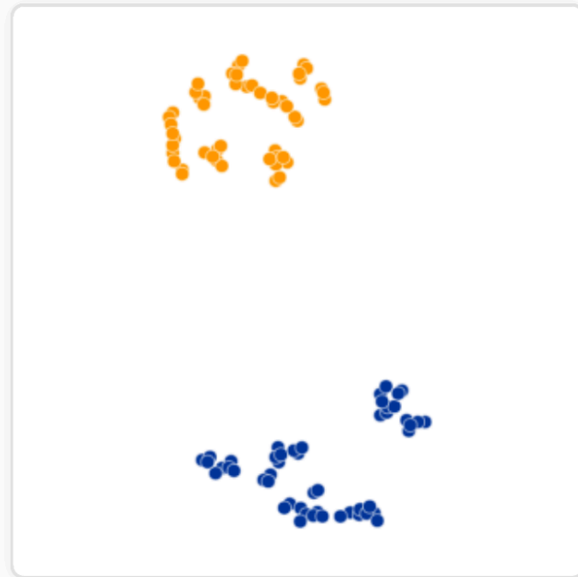
- The choice of perplexity matters
- Data 2D here



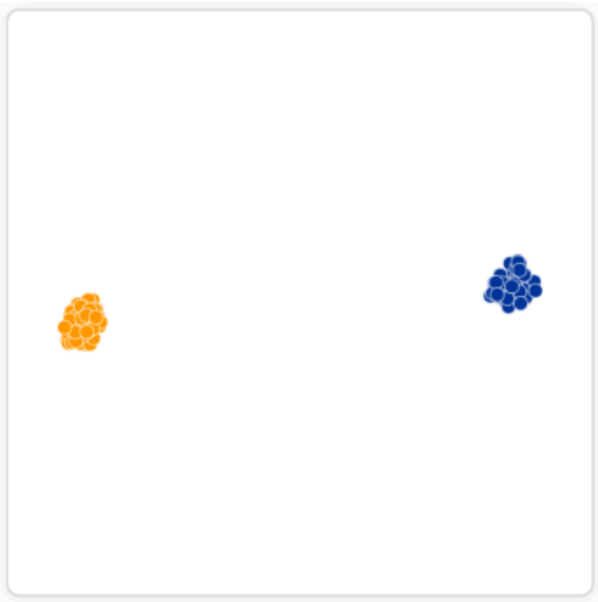
Original



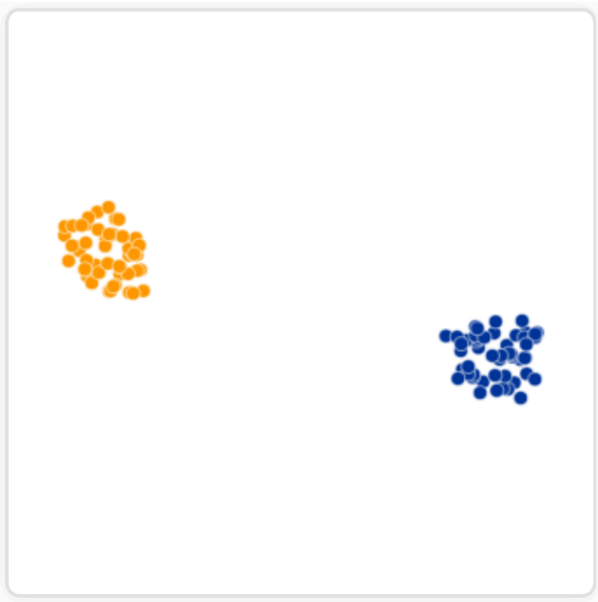
Perplexity: 2
Step: 5,000



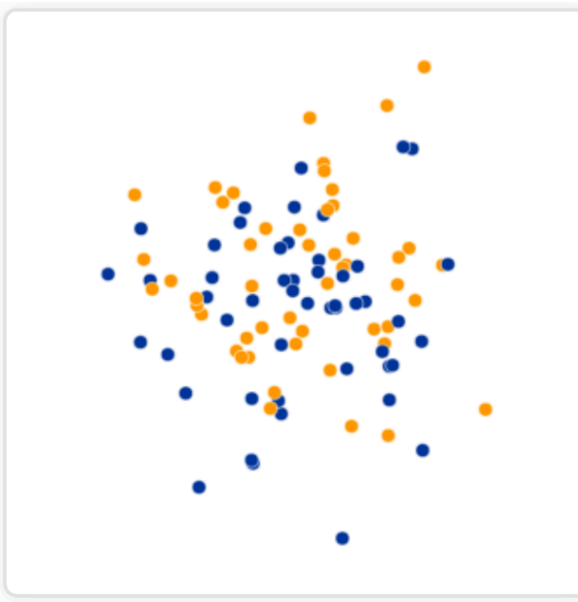
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

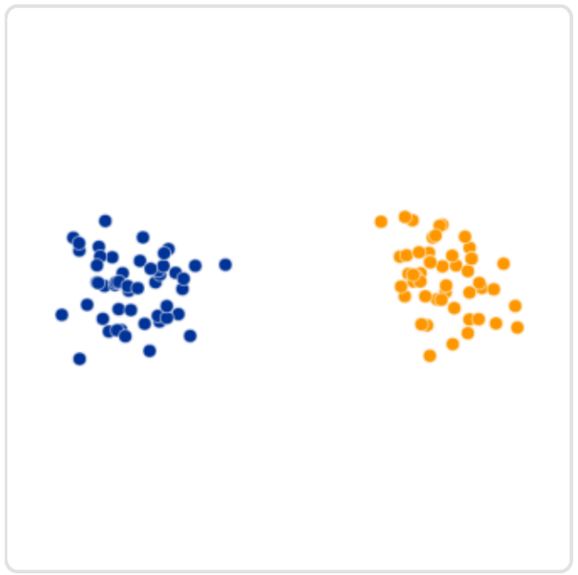


Perplexity: 100
Step: 5,000

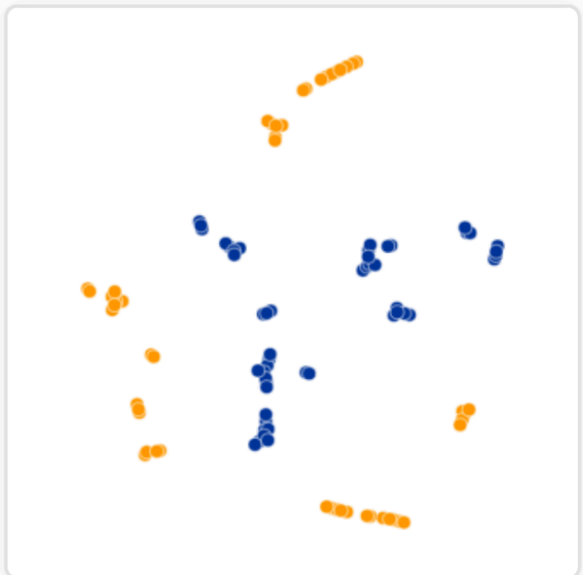
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 "How to use t-SNE effectively." Check it out!]

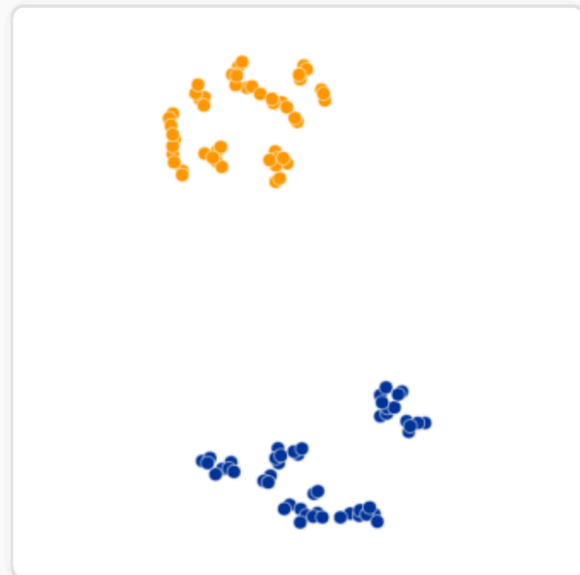
- The choice of perplexity matters
- Data 2D here
- t-SNE paper suggests perplexity between 5 and 50



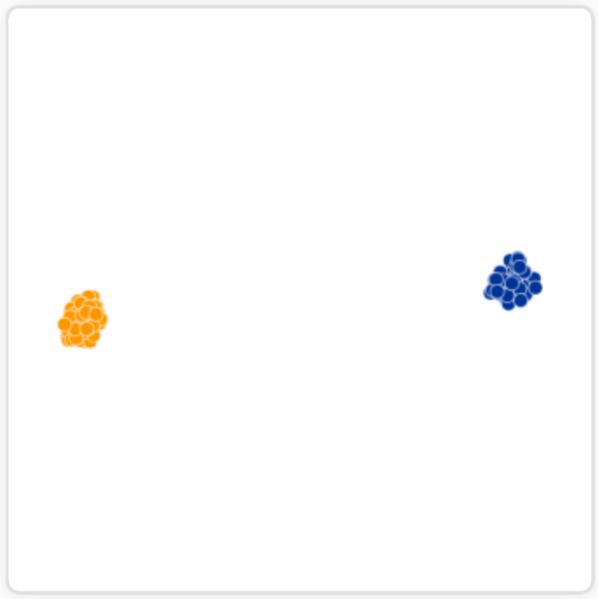
Original



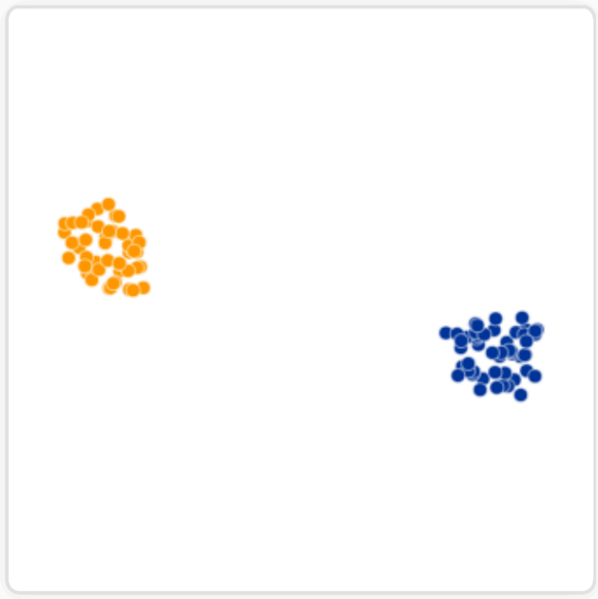
Perplexity: 2
Step: 5,000



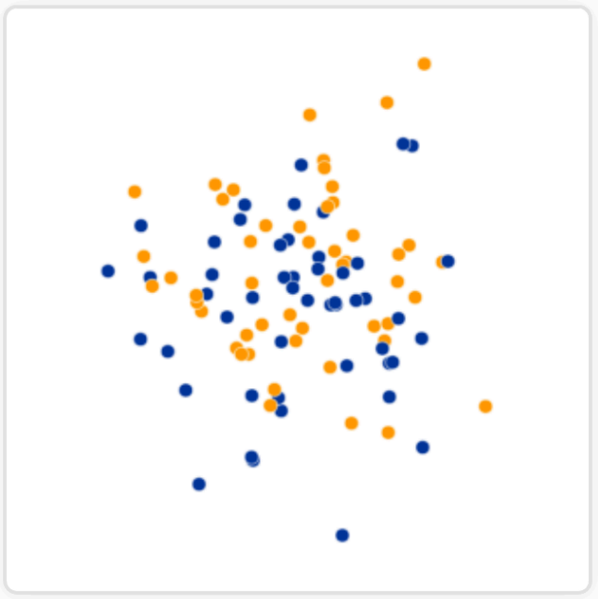
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

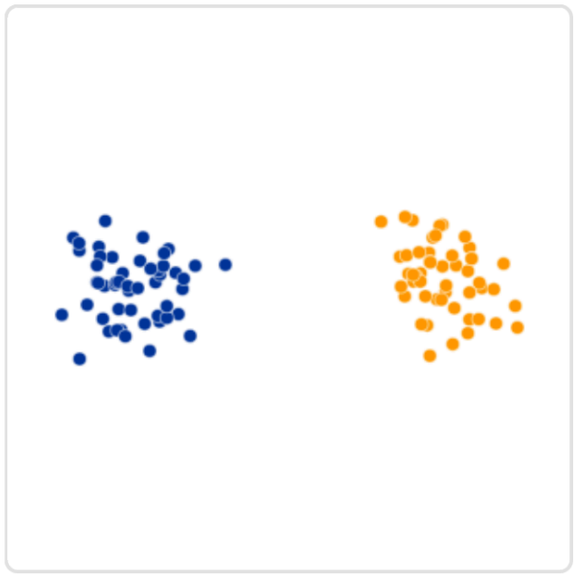


Perplexity: 100
Step: 5,000

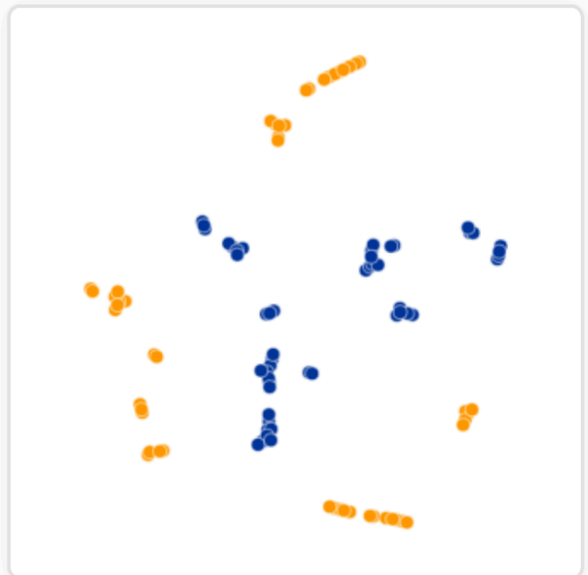
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 "How to use t-SNE effectively." Check it out!]

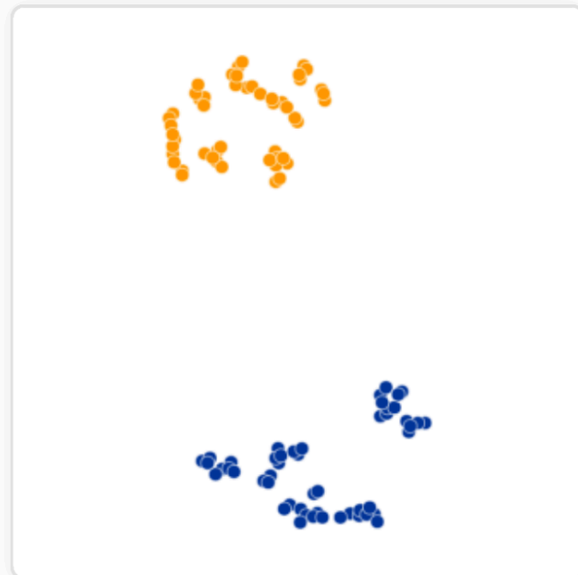
- The choice of perplexity matters
- Data 2D here
- t-SNE paper suggests perplexity between 5 and 50
- Default is often 30



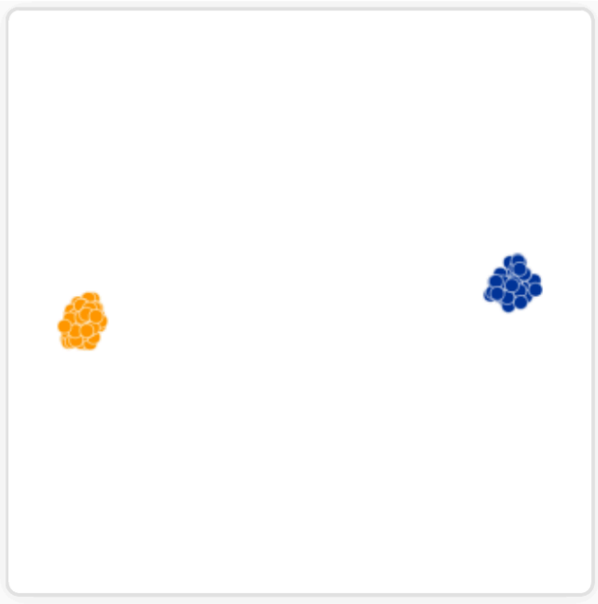
Original



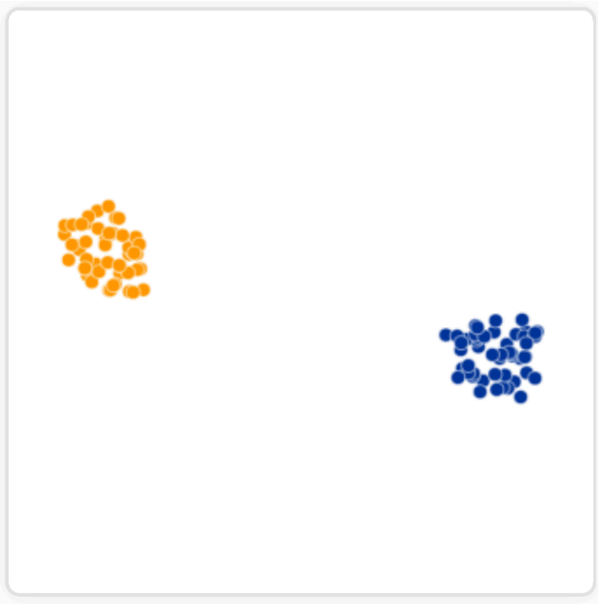
Perplexity: 2
Step: 5,000



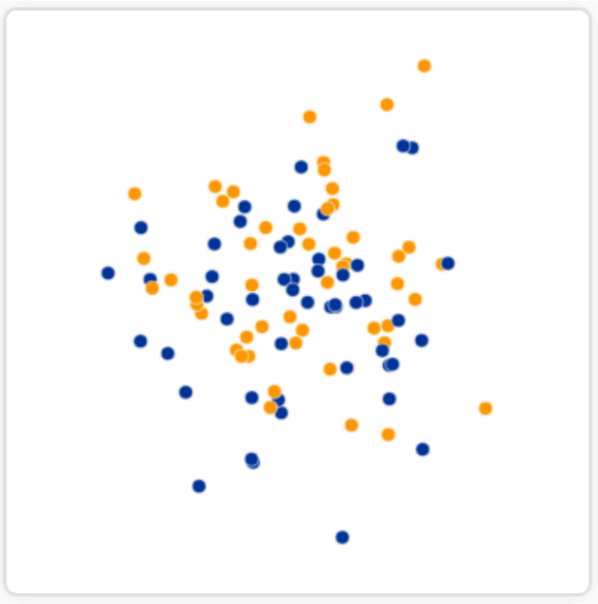
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

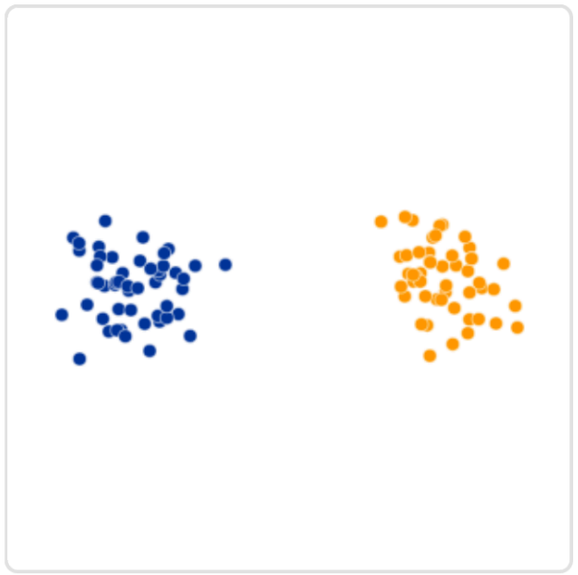


Perplexity: 100
Step: 5,000

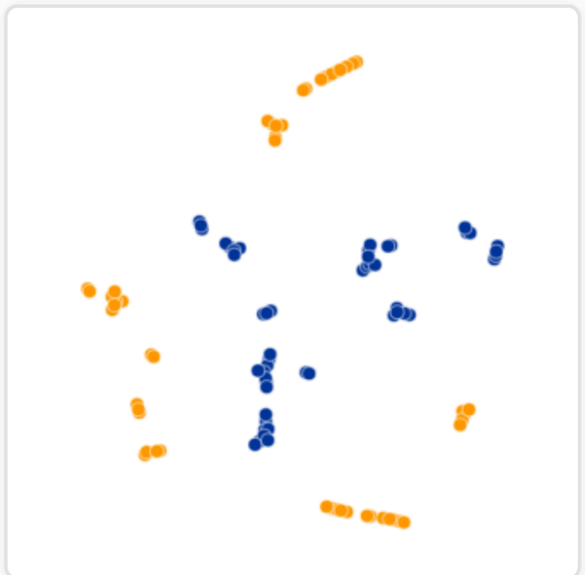
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 "How to use t-SNE effectively." Check it out!]

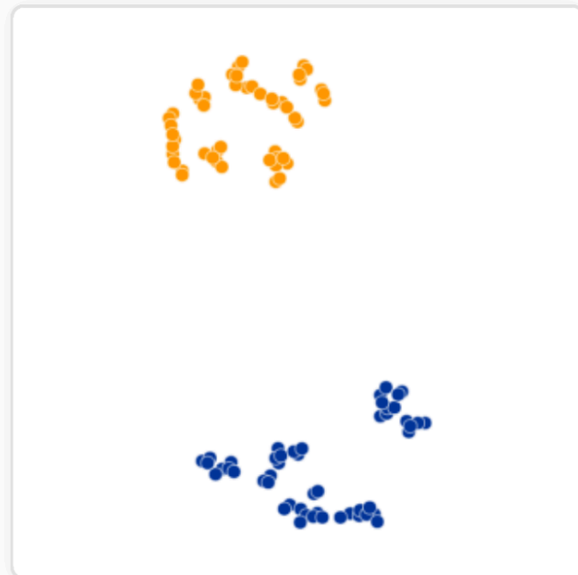
- The choice of perplexity matters
- Data 2D here
- t-SNE paper suggests perplexity between 5 and 50
- Default is often 30
- Have to choose learning rate & # steps (optimization matters)



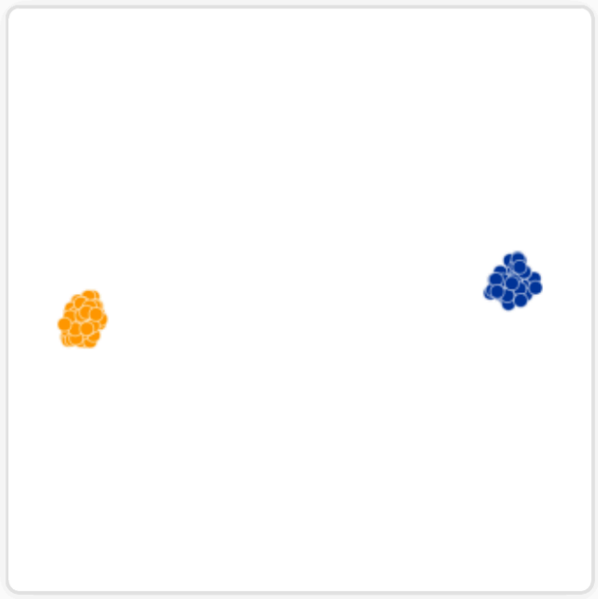
Original



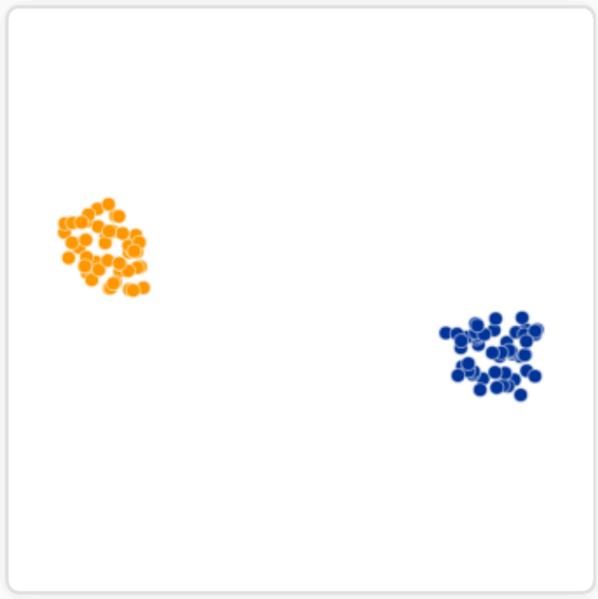
Perplexity: 2
Step: 5,000



Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

Challenges of t-SNE

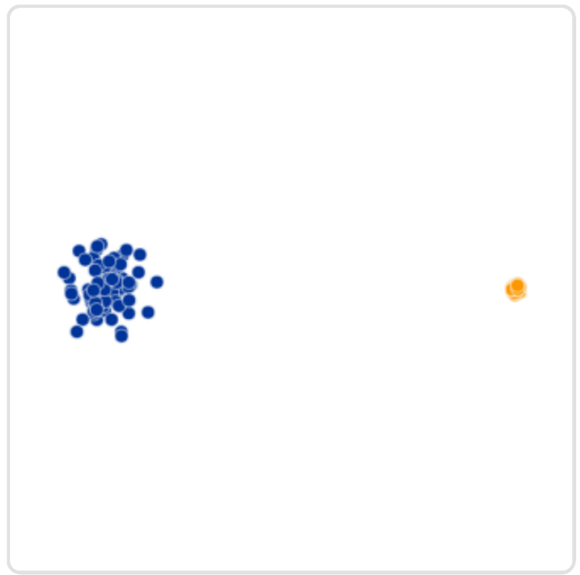
[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

- t-SNE cluster diameters need not reflect original diameters

Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

- t-SNE cluster diameters need not reflect original diameters
- Data 2D here

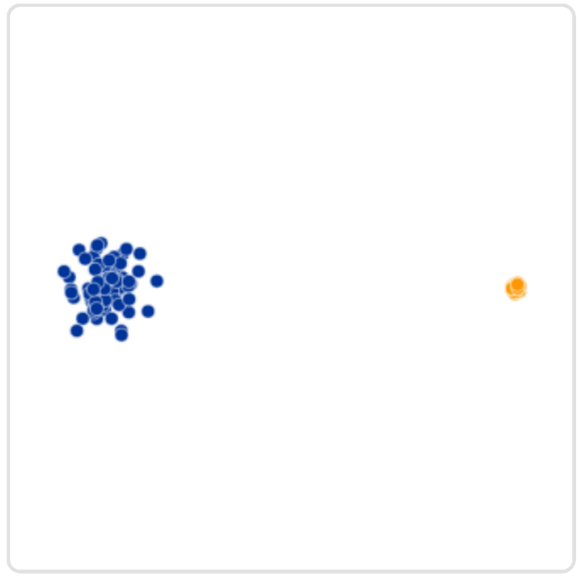


Original

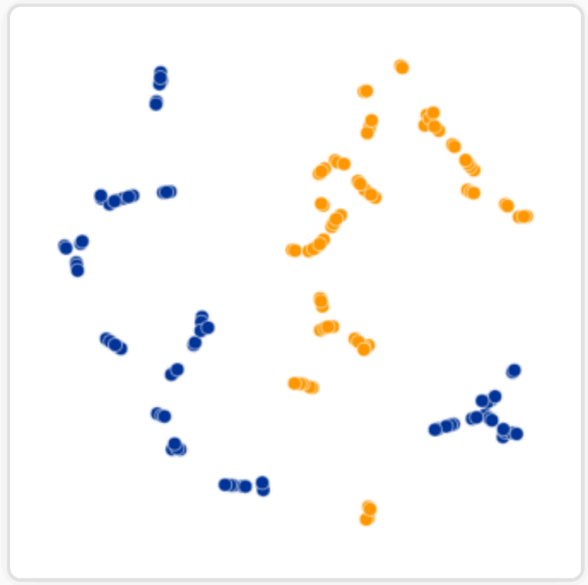
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

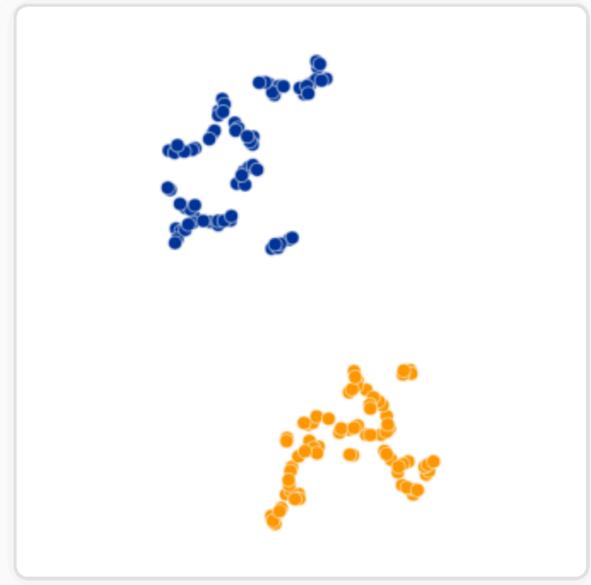
- t-SNE cluster diameters need not reflect original diameters
- Data 2D here



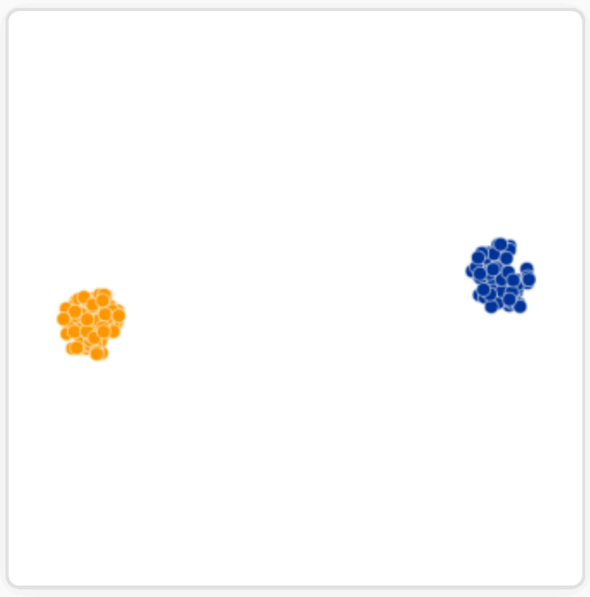
Original



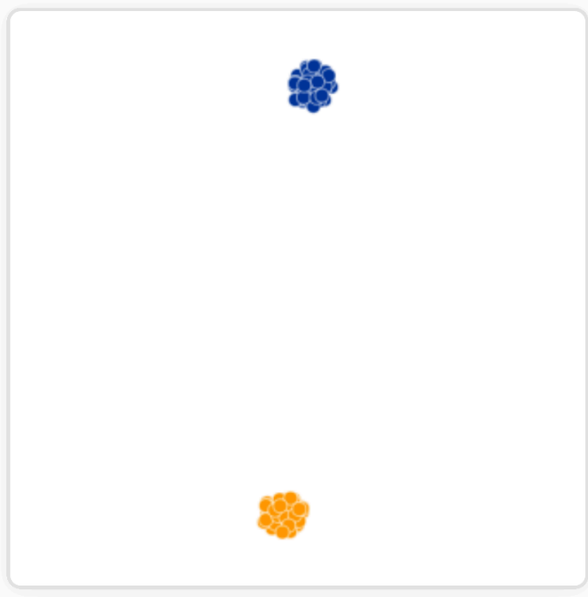
Perplexity: 2
Step: 5,000



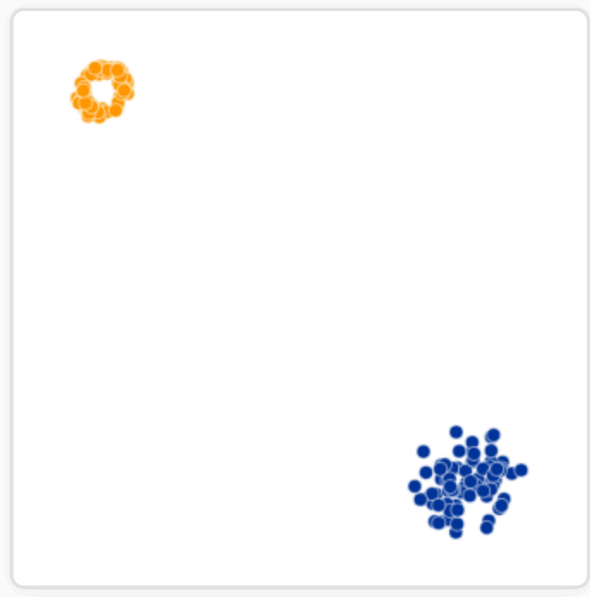
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

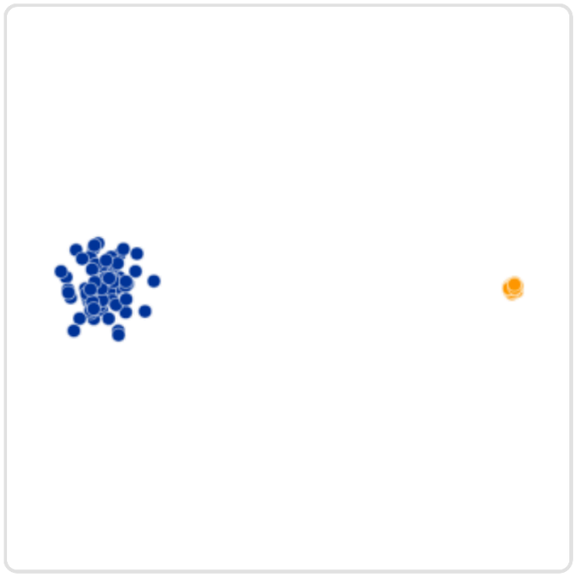


Perplexity: 100
Step: 5,000

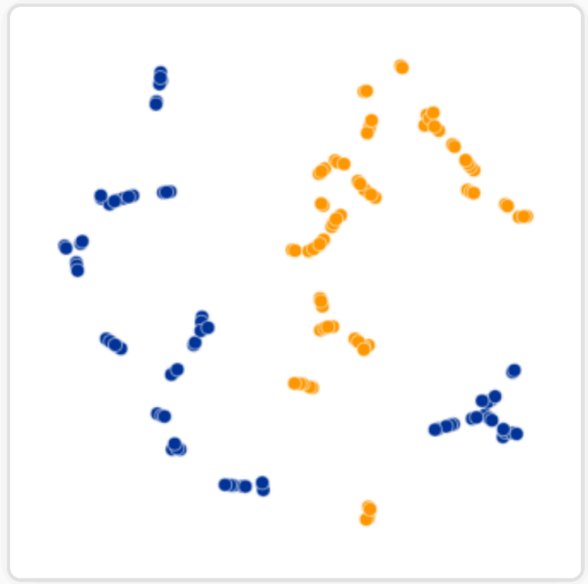
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

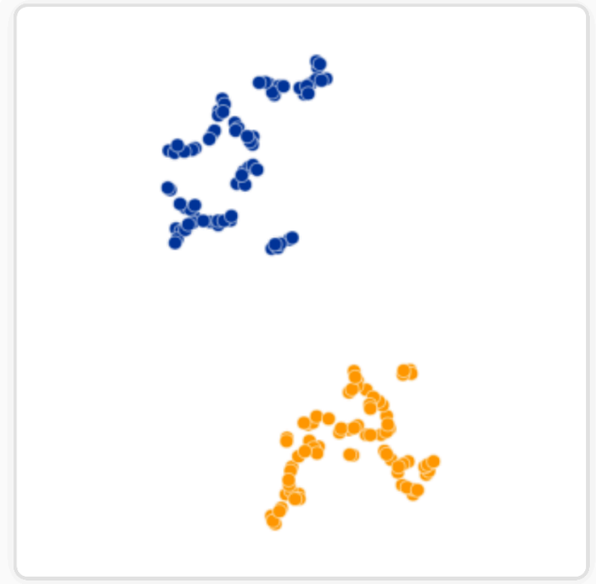
- t-SNE cluster diameters need not reflect original diameters
- Data 2D here
- By design, t-SNE adapts its notion of distance around each point / locally



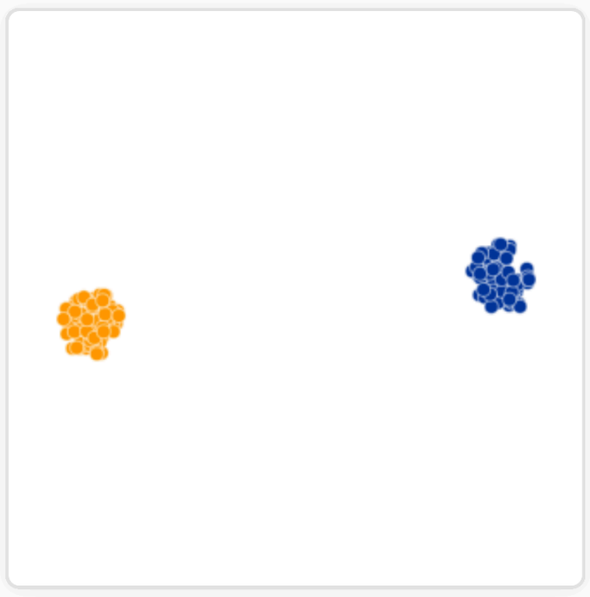
Original



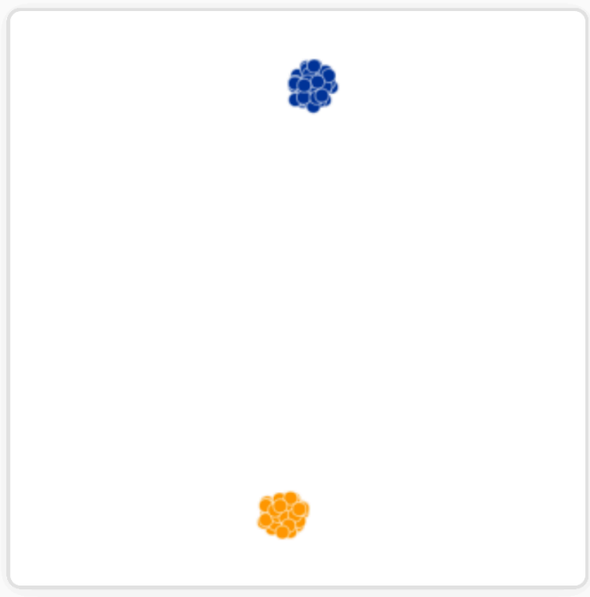
Perplexity: 2
Step: 5,000



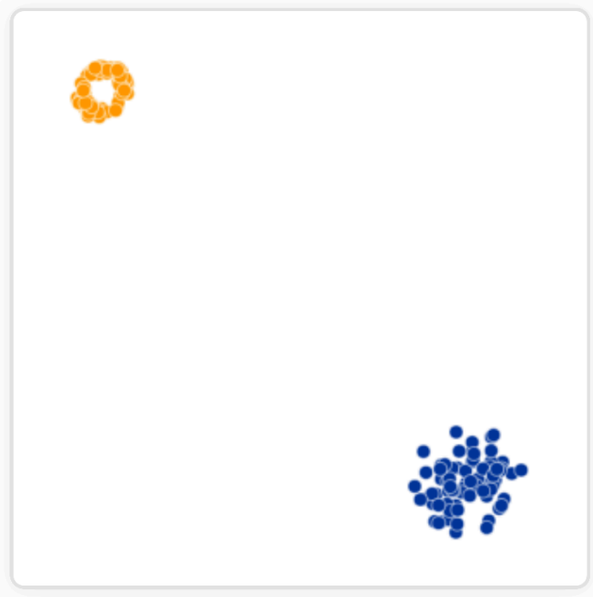
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

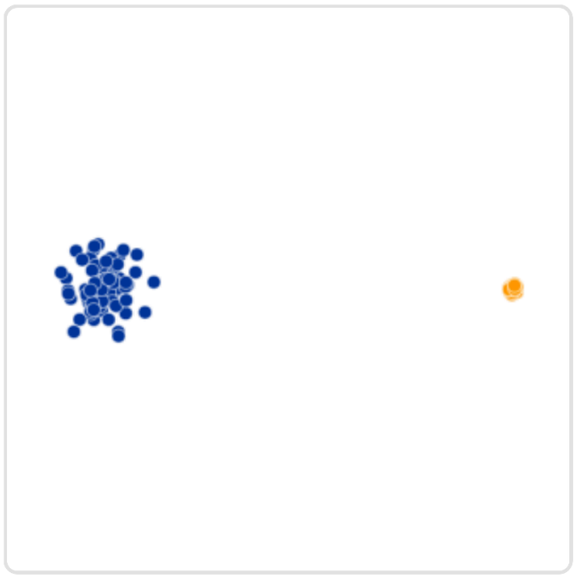


Perplexity: 100
Step: 5,000

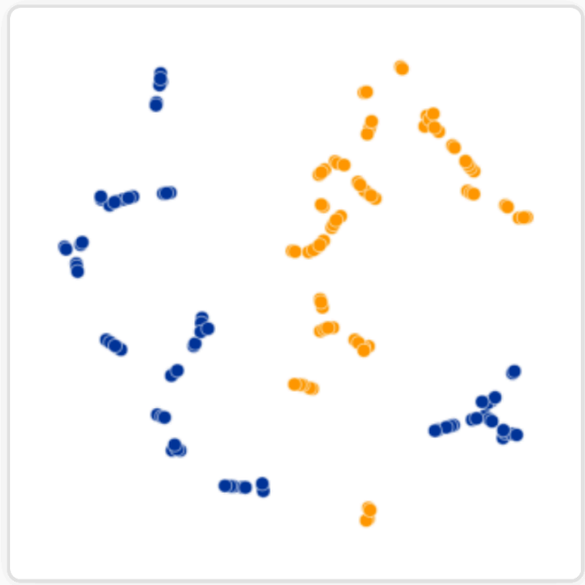
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

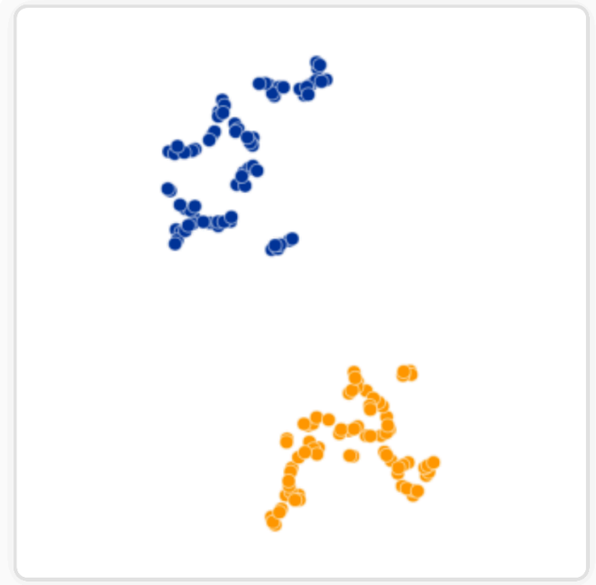
- t-SNE cluster diameters need not reflect original diameters
- Data 2D here
- By design, t-SNE adapts its notion of distance around each point / locally
- Recall that the global perplexity (~controls # of neighbors) $\Rightarrow \{\sigma_n^2\}_{n=1}^N$



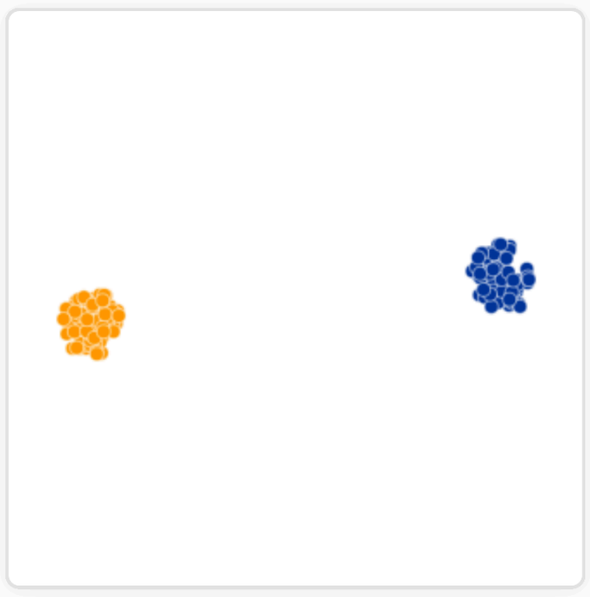
Original



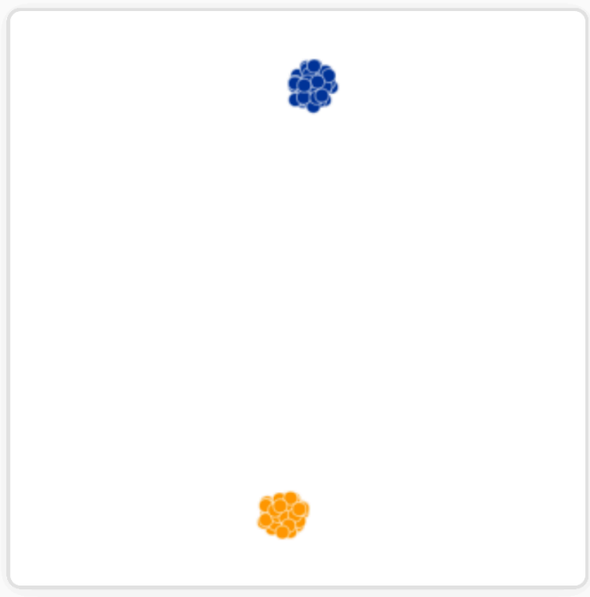
Perplexity: 2
Step: 5,000



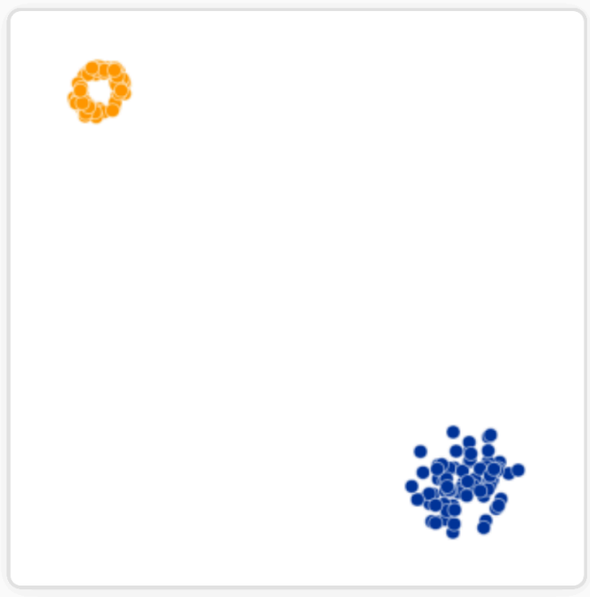
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000



Perplexity: 100
Step: 5,000

Challenges of t-SNE

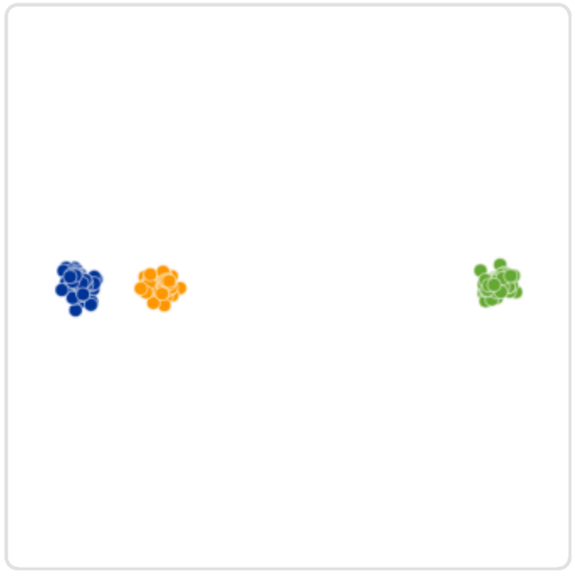
[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

- t-SNE distances between clusters need not reflect originals

Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

- t-SNE distances between clusters need not reflect originals
- Data 2D here

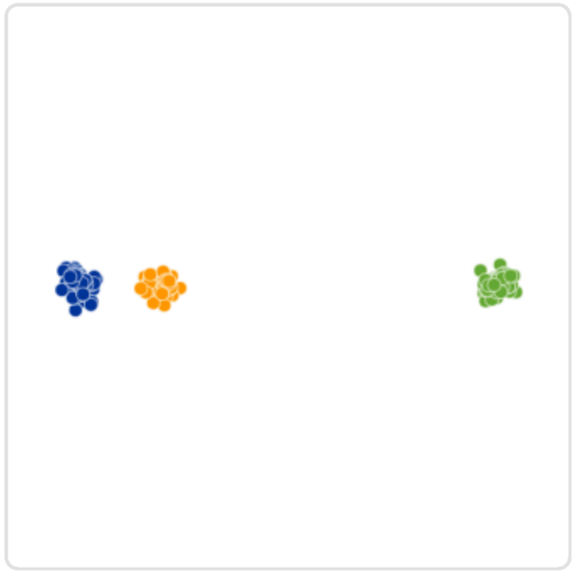


Original

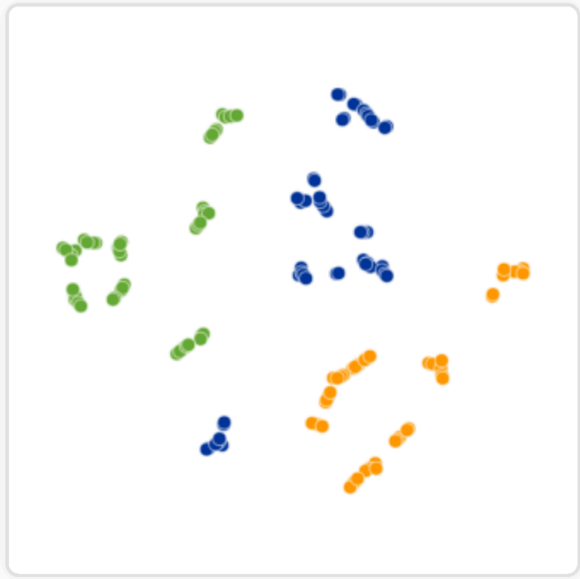
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

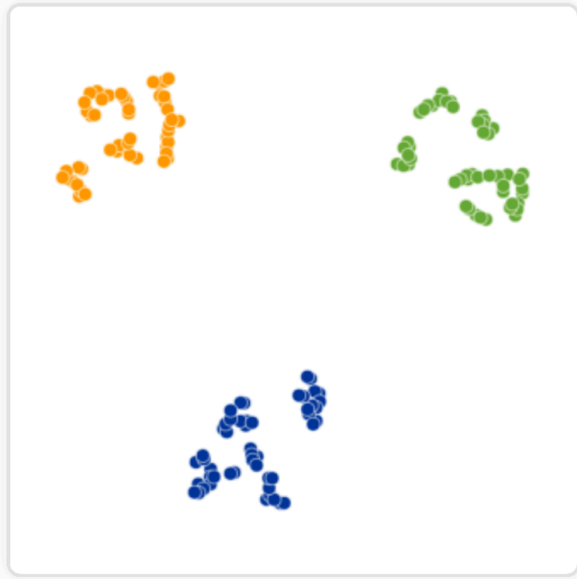
- t-SNE distances between clusters need not reflect originals
- Data 2D here



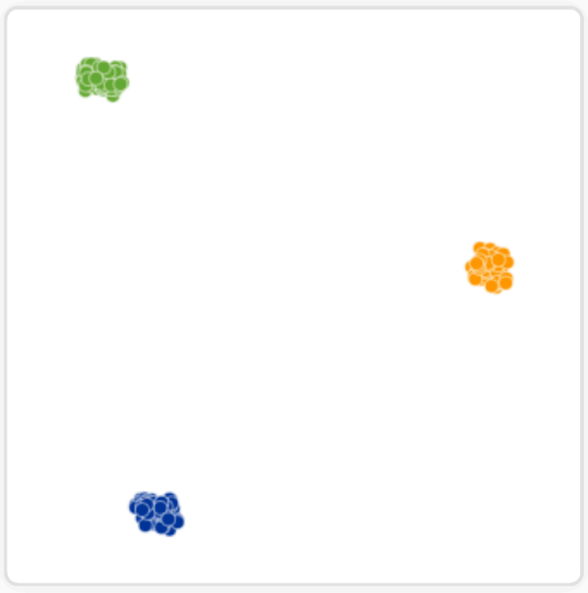
Original



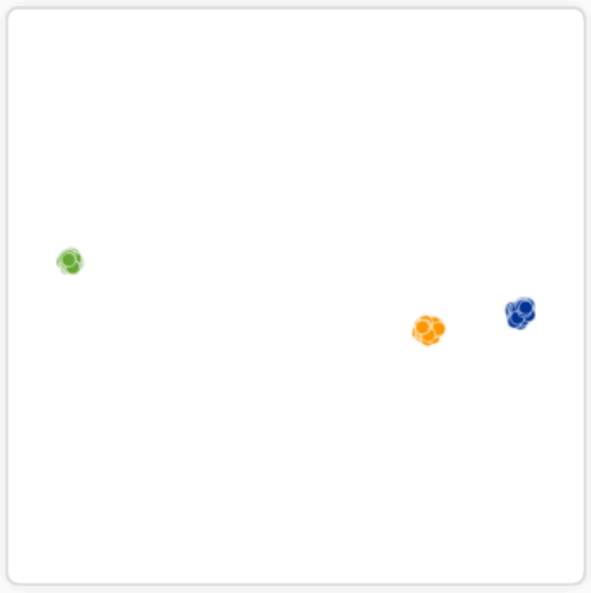
Perplexity: 2
Step: 5,000



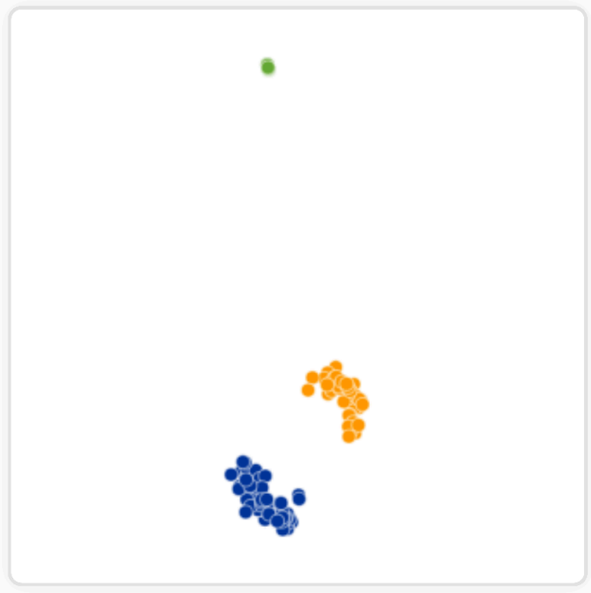
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000

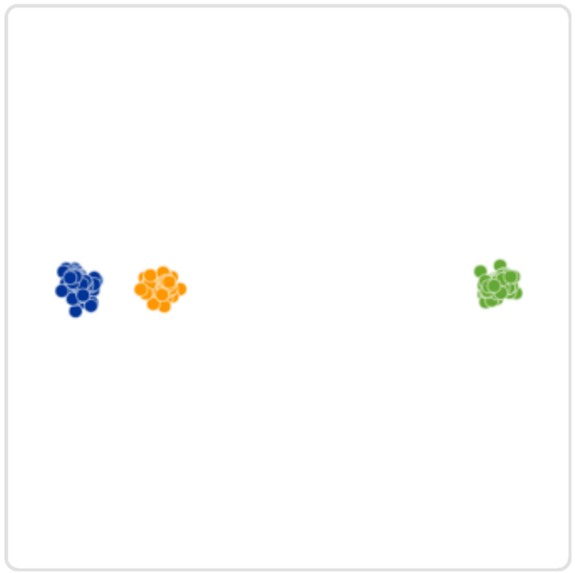


Perplexity: 100
Step: 5,000

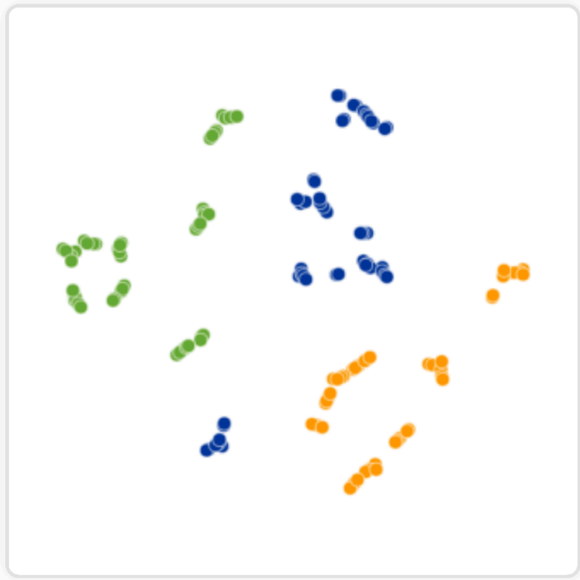
Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

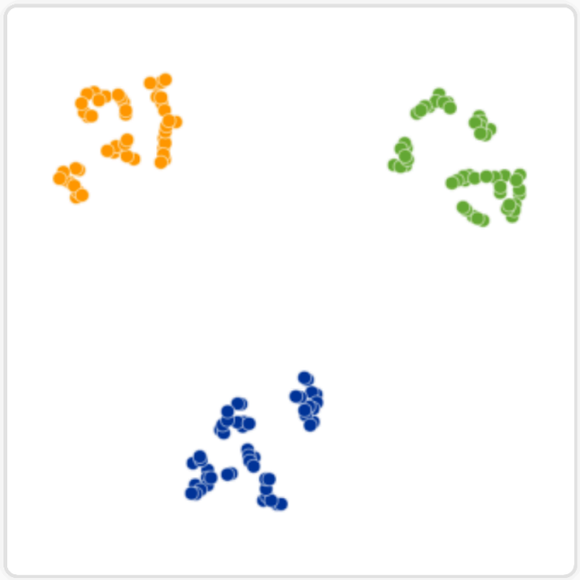
- t-SNE distances between clusters need not reflect originals
- Data 2D here
- Perplexity 50 and above capture the relative distances between clusters; can we count on that in general?



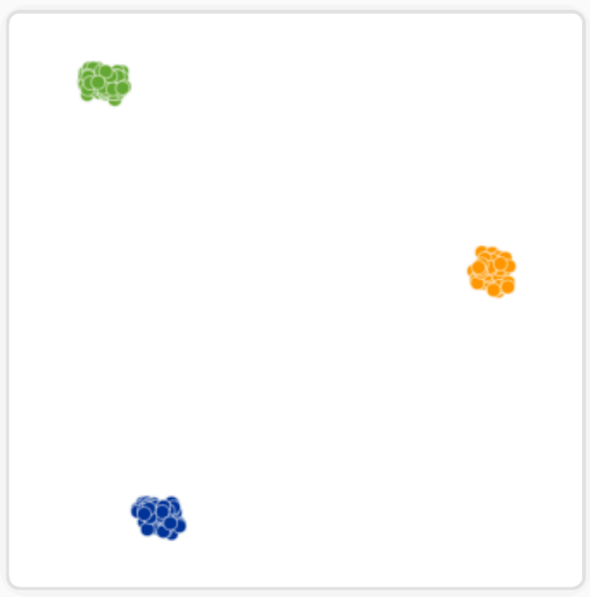
Original



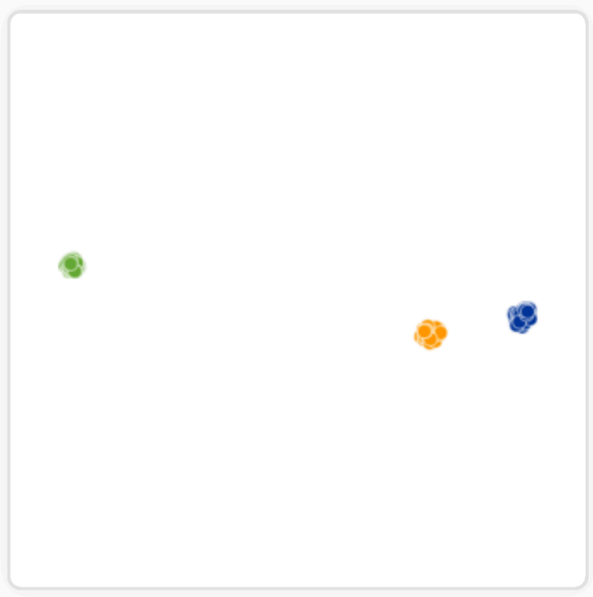
Perplexity: 2
Step: 5,000



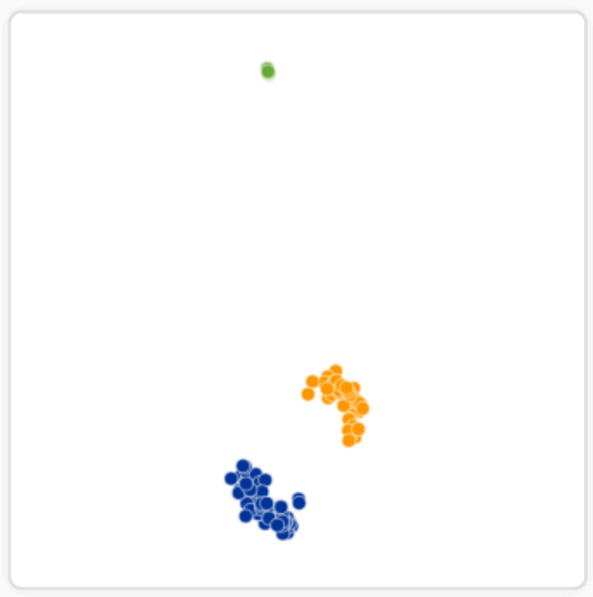
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000



Perplexity: 100
Step: 5,000

Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

- t-SNE distances between clusters need not reflect originals
- Data 2D here

Original

Challenges of t-SNE

[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

- t-SNE distances between clusters need not reflect originals
- Data 2D here
- Same setup but with 200 data points per cluster (previous slide had 50 per cluster)

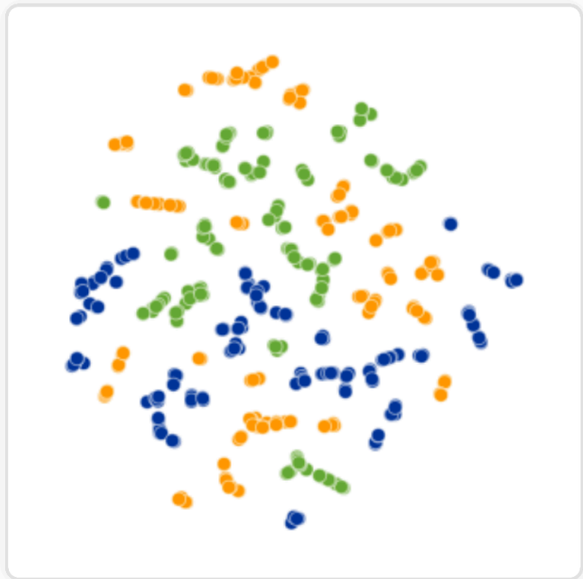
Original

Challenges of t-SNE

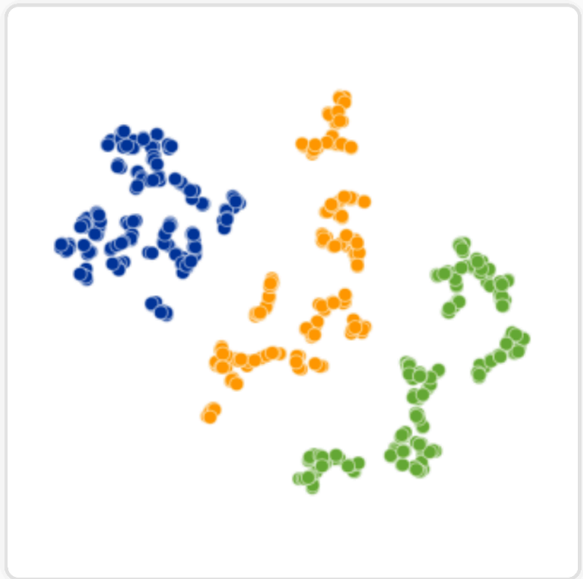
[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

- t-SNE distances between clusters need not reflect originals
- Data 2D here
- Same setup but with 200 data points per cluster (previous slide had 50 per cluster)

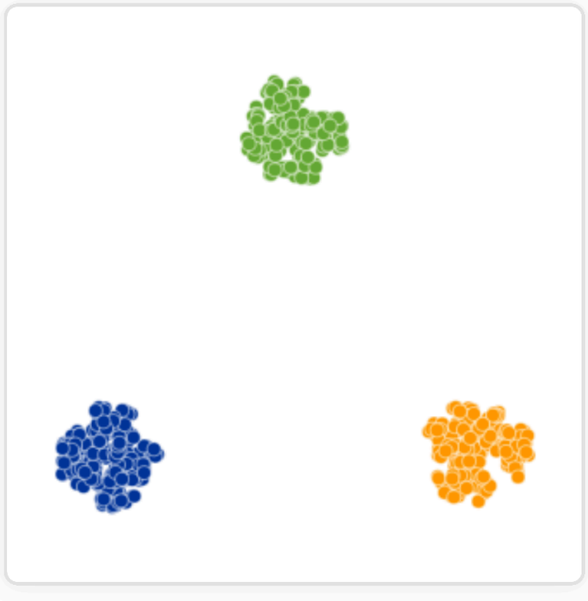
Original



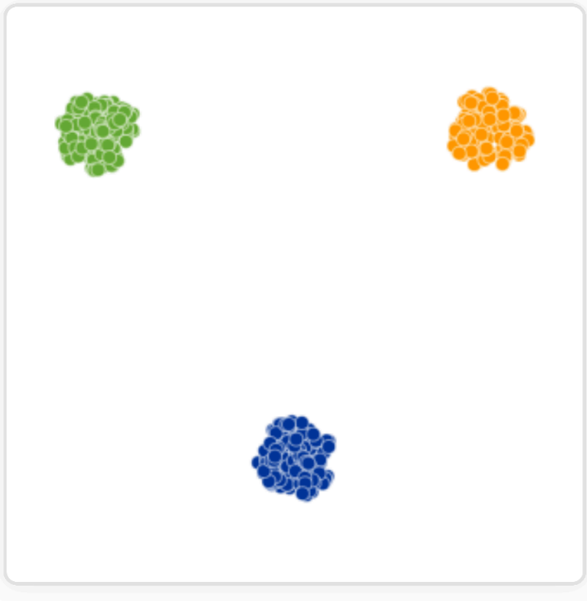
Perplexity: 2
Step: 5,000



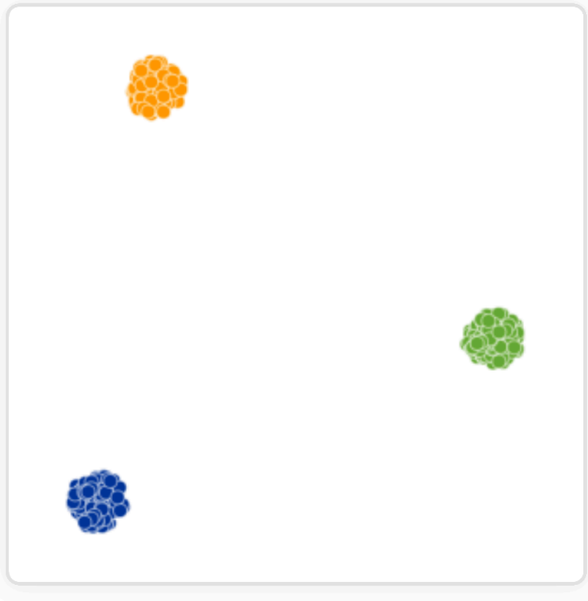
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000



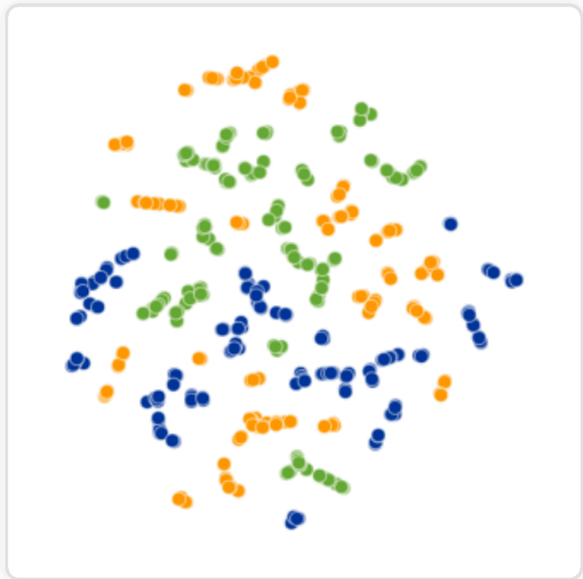
Perplexity: 100
Step: 5,000

Challenges of t-SNE

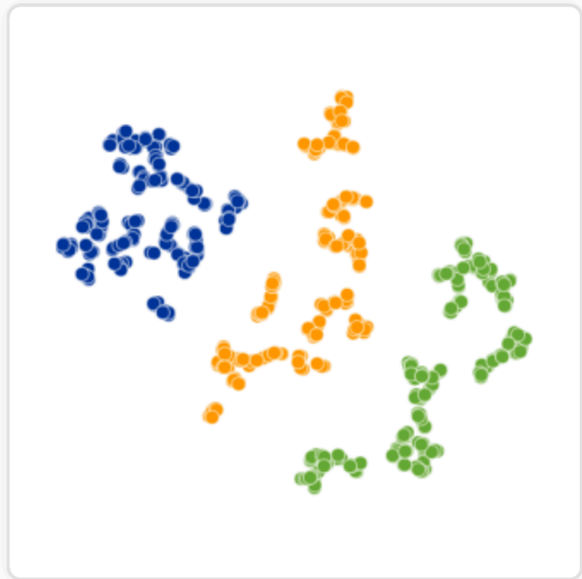
[Largely borrowed from Wattenberg et al 2016 “How to use t-SNE effectively.” Check it out!]

- t-SNE distances between clusters need not reflect originals
- Data 2D here
- Same setup but with 200 data points per cluster (previous slide had 50 per cluster)
- Now none of the plots reflect the distances in the original data

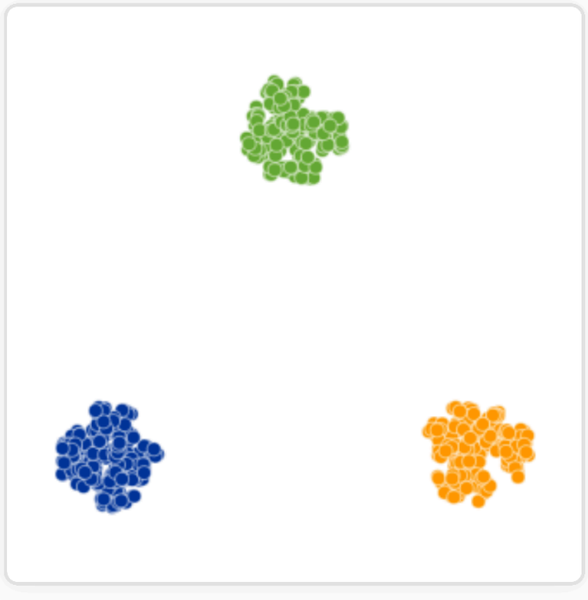
Original



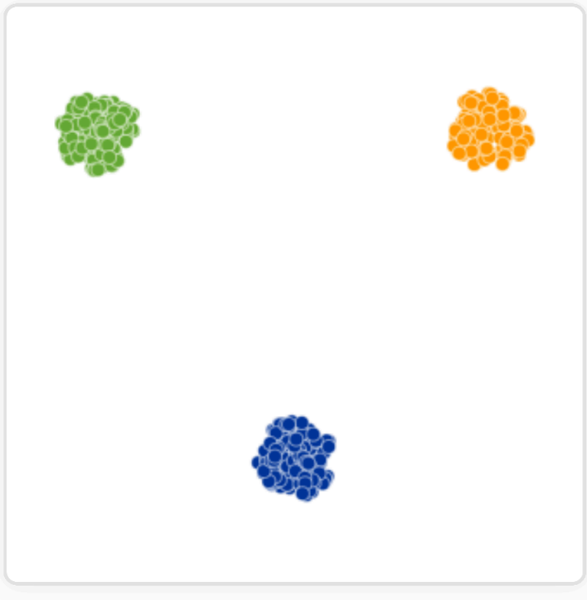
Perplexity: 2
Step: 5,000



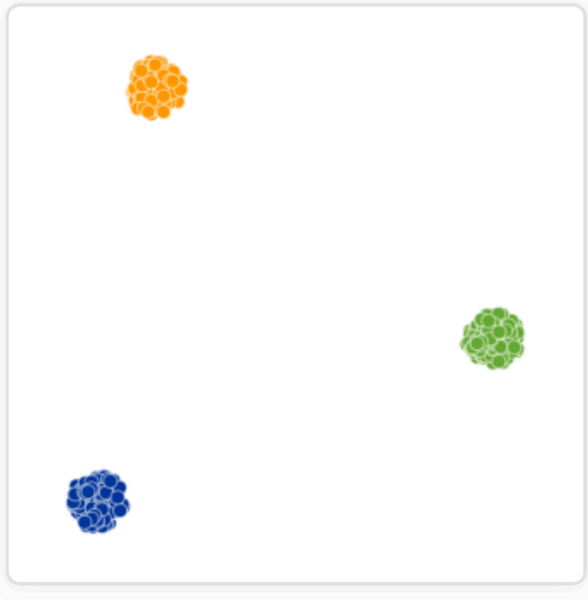
Perplexity: 5
Step: 5,000



Perplexity: 30
Step: 5,000



Perplexity: 50
Step: 5,000



Perplexity: 100
Step: 5,000

References (1/1)

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Murphy, K. P. (2022). Probabilistic machine learning: an introduction. MIT Press.

Shlens, J. (2014). A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100.

Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to use t-SNE effectively. https://distill.pub/2016/misread-tsne/