6.7900: Machine Learning
Lecture 18

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 Times

I. GPs for regression: model and inference

II. Examples and common types of missing data

Today

I. How to proceed when data is missing

II. Dimensionality reduction

III. Principal components analysis (PCA)

Recap: missing data

Recap: missing data

- Some examples:

Recap: missing data

- Some examples:
  - Data corruption

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. “What is your income?”

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. “What is your income?”
  - Scientific phenomena that are hard to observe

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. “What is your income?”
  - Scientific phenomena that are hard to observe
  - Data collected in different ways / combining datasets

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. “What is your income?”
  - Scientific phenomena that are hard to observe
  - Data collected in different ways / combining datasets
- Types of missing data

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. “What is your income?”
  - Scientific phenomena that are hard to observe
  - Data collected in different ways / combining datasets
- Types of missing data
  - Let $M_{nd} = 1$ if feature d in data point n is missing, else 0

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. “What is your income?”
  - Scientific phenomena that are hard to observe
  - Data collected in different ways / combining datasets
- Types of missing data
  - Let $M_{nd} = 1$ if feature d in data point n is missing, else 0
  - Assume Y and X_{obs} observed; X_{mis} missing

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. “What is your income?”
  - Scientific phenomena that are hard to observe
  - Data collected in different ways / combining datasets
- Types of missing data
  - Let $M_{nd} = 1$ if feature d in data point n is missing, else 0
  - Assume Y and X_{obs} observed; X_{mis} missing
  - Missing completely at random (MCAR)

$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. "What is your income?"
  - Scientific phenomena that are hard to observe
  - Data collected in different ways / combining datasets
- Types of missing data
  - Let M_{nd} = 1 if feature *d* in data point *n* is missing, else 0
  - Assume Y and X_{obs} observed; X_{mis} missing
  - Missing completely at random (MCAR)
$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$
  - Missing at random (MAR)
$$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M|X_{\text{obs}}, Y)$$

Recap: missing data

- Some examples:
  - Data corruption
  - Survey non-response
    - E.g. “What is your income?”
  - Scientific phenomena that are hard to observe
  - Data collected in different ways / combining datasets
- Types of missing data
  - Let $M_{nd} = 1$ if feature d in data point n is missing, else 0
  - Assume Y and X_{obs} observed; X_{mis} missing
  - Missing completely at random (MCAR)
    $$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M)$$
  - Missing at random (MAR)
    $$p(M|X_{\text{obs}}, X_{\text{mis}}, Y) = p(M|X_{\text{obs}}, Y)$$
  - Missing not at random (MNAR or NMAR)
    - Missingness can depend on something besides the observed data

What to do?

- So what are our practical options?

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
    - Can visualize patterns of missingness

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data
  - Pro: Very easy & can be default in software packages(!)

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data
  - Pro: Very easy & can be default in software packages(!)
  - Con: Losing data (e.g. D=100 and each data point misses one feature. 99% observed, but lose all data.)

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data
  - Pro: Very easy & can be default in software packages(!)
  - Con: Losing data (e.g. D=100 and each data point misses one feature. 99% observed, but lose all data.)
  - Con: Losing ability to make predictions on some test data

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data
  - Pro: Very easy & can be default in software packages(!)
  - Con: Losing data (e.g. D=100 and each data point misses one feature. 99% observed, but lose all data.)
  - Con: Losing ability to make predictions on some test data
  - Con: If not MCAR, could introduce bias

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data
  - Pro: Very easy & can be default in software packages(!)
  - Con: Losing data (e.g. D=100 and each data point misses one feature. 99% observed, but lose all data.)
  - Con: Losing ability to make predictions on some test data
  - Con: If not MCAR, could introduce bias
    - Can reweight observed data (cf. covariate shift). This has all the usual challenges of reweighting schemes.

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data
  - Pro: Very easy & can be default in software packages(!)
  - Con: Losing data (e.g. D=100 and each data point misses one feature. 99% observed, but lose all data.)
  - Con: Losing ability to make predictions on some test data
  - Con: If not MCAR, could introduce bias
    - Can reweight observed data (cf. covariate shift). This has all the usual challenges of reweighting schemes.
- Option: Information in the missingness

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data
  - Pro: Very easy & can be default in software packages(!)
  - Con: Losing data (e.g. D=100 and each data point misses one feature. 99% observed, but lose all data.)
  - Con: Losing ability to make predictions on some test data
  - Con: If not MCAR, could introduce bias
    - Can reweight observed data (cf. covariate shift). This has all the usual challenges of reweighting schemes.
- Option: Information in the missingness
  - If feature is already categorical, add a “missing” category

What to do?

Why do anything? To run standard ML algorithms; for quality of results

- So what are our practical options?
- Always a great idea to start by visualizing the data
  - Can visualize patterns of missingness
  - Might help suggest the mechanism: e.g. weather data instruments malfunction during extreme storms [Tierney 2024]
- Option: throw out data points or features with missing data
  - Pro: Very easy & can be default in software packages(!)
  - Con: Losing data (e.g. D=100 and each data point misses one feature. 99% observed, but lose all data.)
  - Con: Losing ability to make predictions on some test data
  - Con: If not MCAR, could introduce bias
    - Can reweight observed data (cf. covariate shift). This has all the usual challenges of reweighting schemes.
- Option: Information in the missingness
  - If feature is already categorical, add a “missing” category
  - More generally, can introduce a “missingness” feature

What to do?

What to do?

- Option: Single imputation,
  i.e. replace missing value

What to do?

- Option: Single imputation, i.e. replace missing value
  - E.g. use mean or median of observed values (cf. MCAR)

What to do?

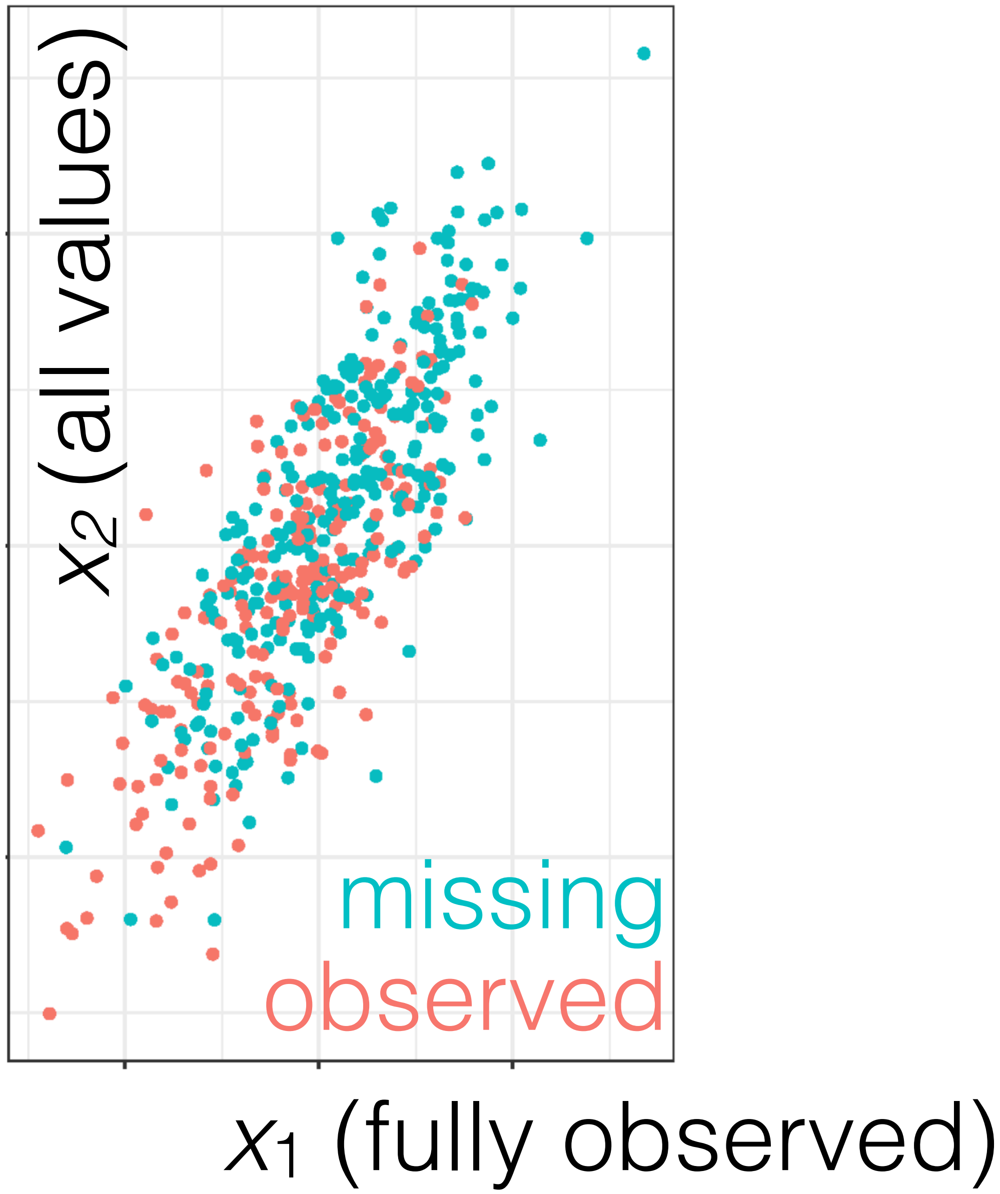
- Option: Single imputation, i.e. replace missing value
    - E.g. use mean or median of observed values (cf. MCAR)
    - E.g. use random sample from observed values (cf. MCAR)

What to do?

- Option: Single imputation, i.e. replace missing value
    - E.g. use mean or median of observed values (cf. MCAR)
    - E.g. use random sample from observed values (cf. MCAR)
    - E.g. regression estimate of missing features from other features (cf. MAR)

What to do?

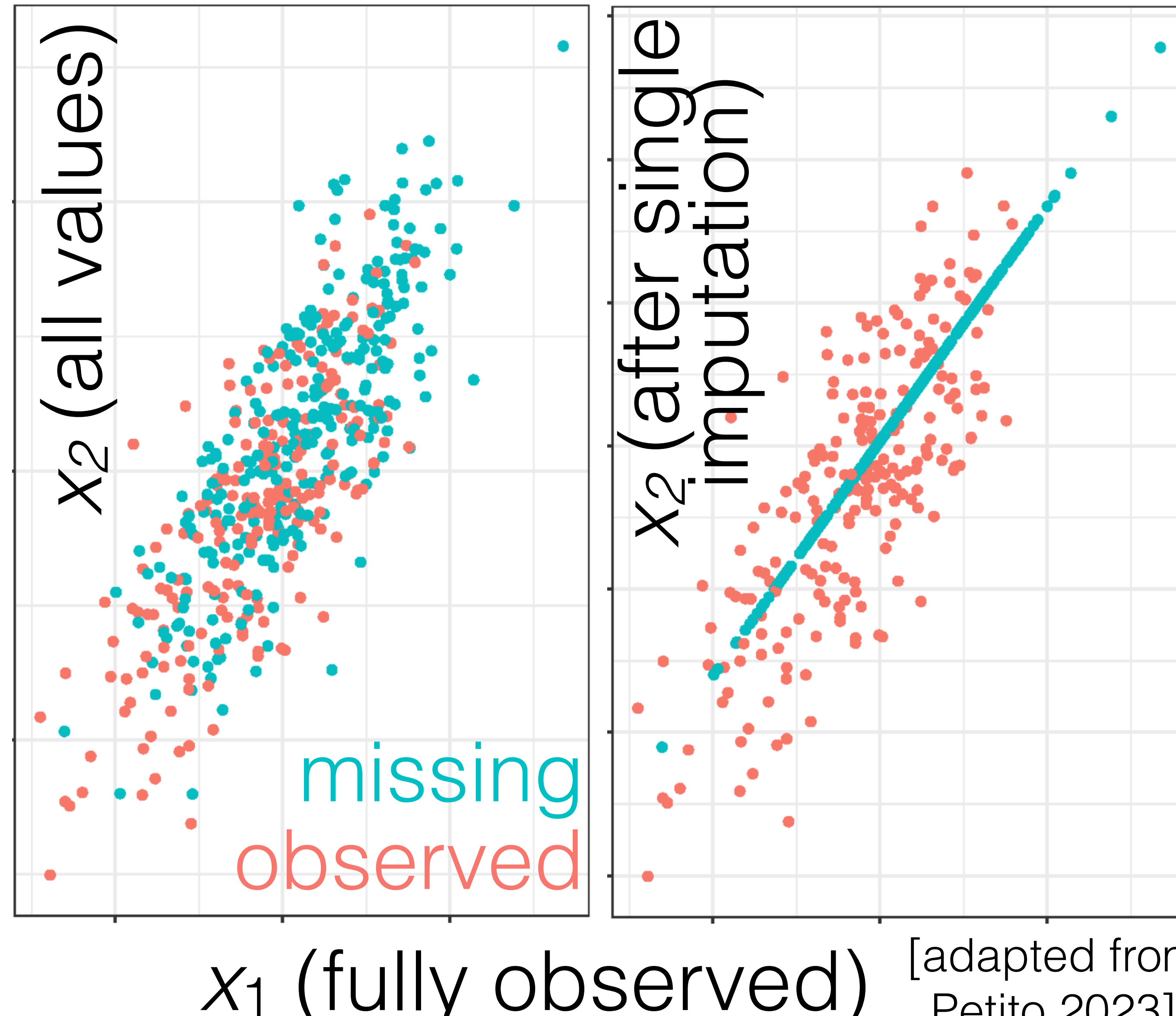
- Option: Single imputation, i.e. replace missing value
  - E.g. use mean or median of observed values (cf. MCAR)
  - E.g. use random sample from observed values (cf. MCAR)
  - E.g. regression estimate of missing features from other features (cf. MAR)



[adapted from Petito 2023]

What to do?

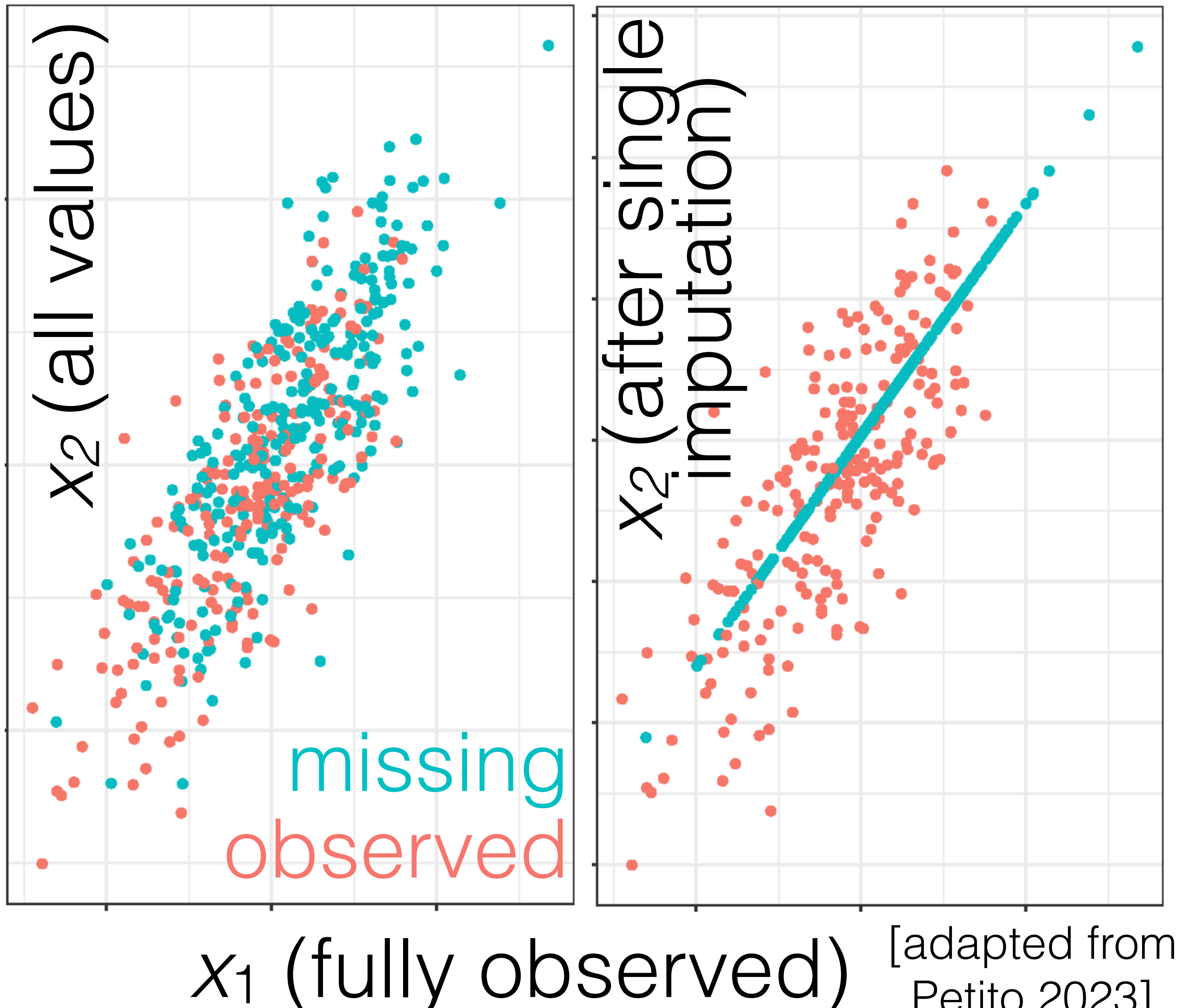
- Option: Single imputation, i.e. replace missing value
  - E.g. use mean or median of observed values (cf. MCAR)
  - E.g. use random sample from observed values (cf. MCAR)
  - E.g. regression estimate of missing features from other features (cf. MAR)



[adapted from Petito 2023]

What to do?

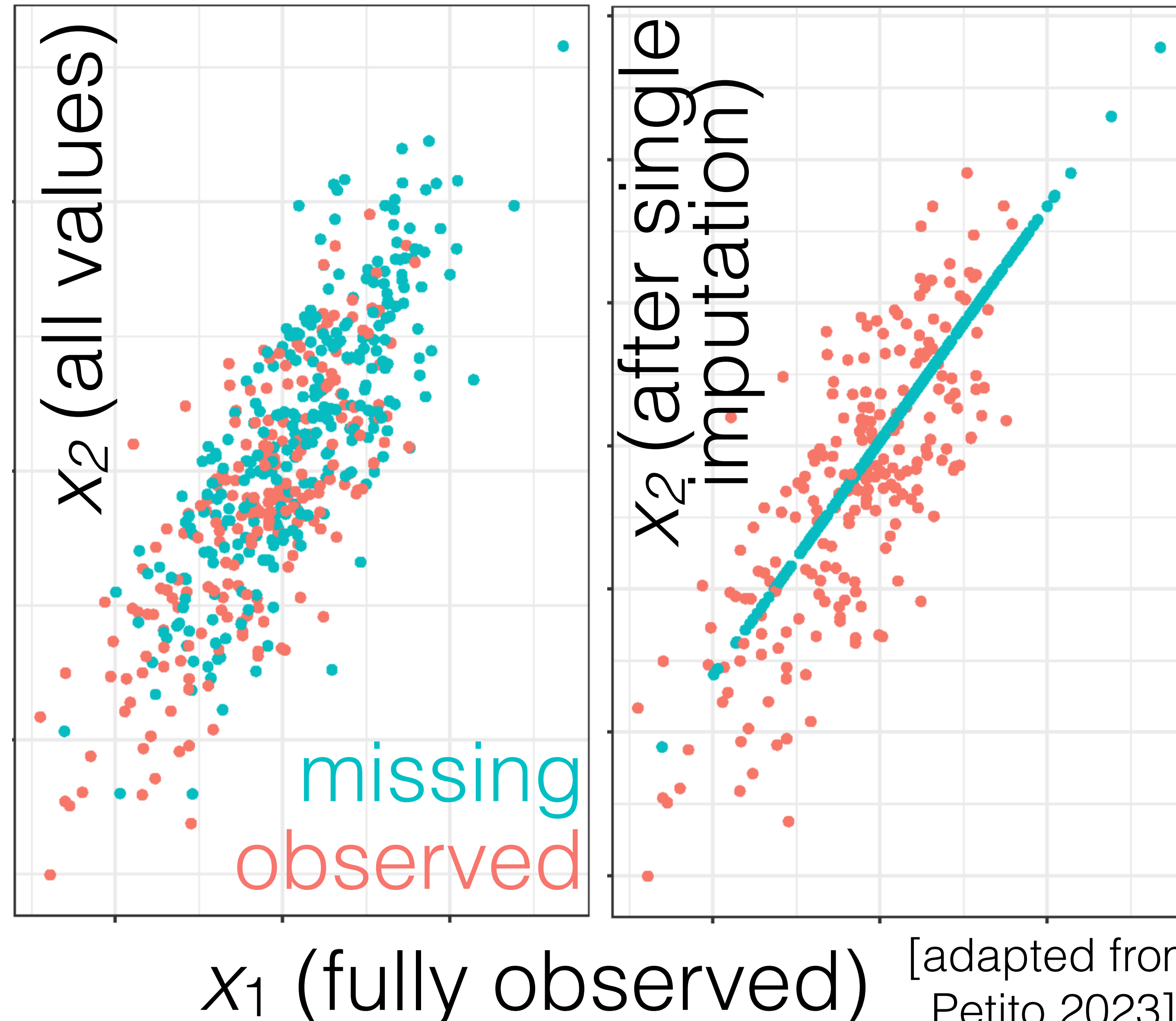
- Option: Single imputation, i.e. replace missing value
    - E.g. use mean or median of observed values (cf. MCAR)
    - E.g. use random sample from observed values (cf. MCAR)
    - E.g. regression estimate of missing features from other features (cf. MAR)
- Con of methods above: no uncertainty in missing values



[adapted from Petito 2023]

What to do?

- Option: Single imputation, i.e. replace missing value
  - E.g. use mean or median of observed values (cf. MCAR)
  - E.g. use random sample from observed values (cf. MCAR)
  - E.g. regression estimate of missing features from other features (cf. MAR)

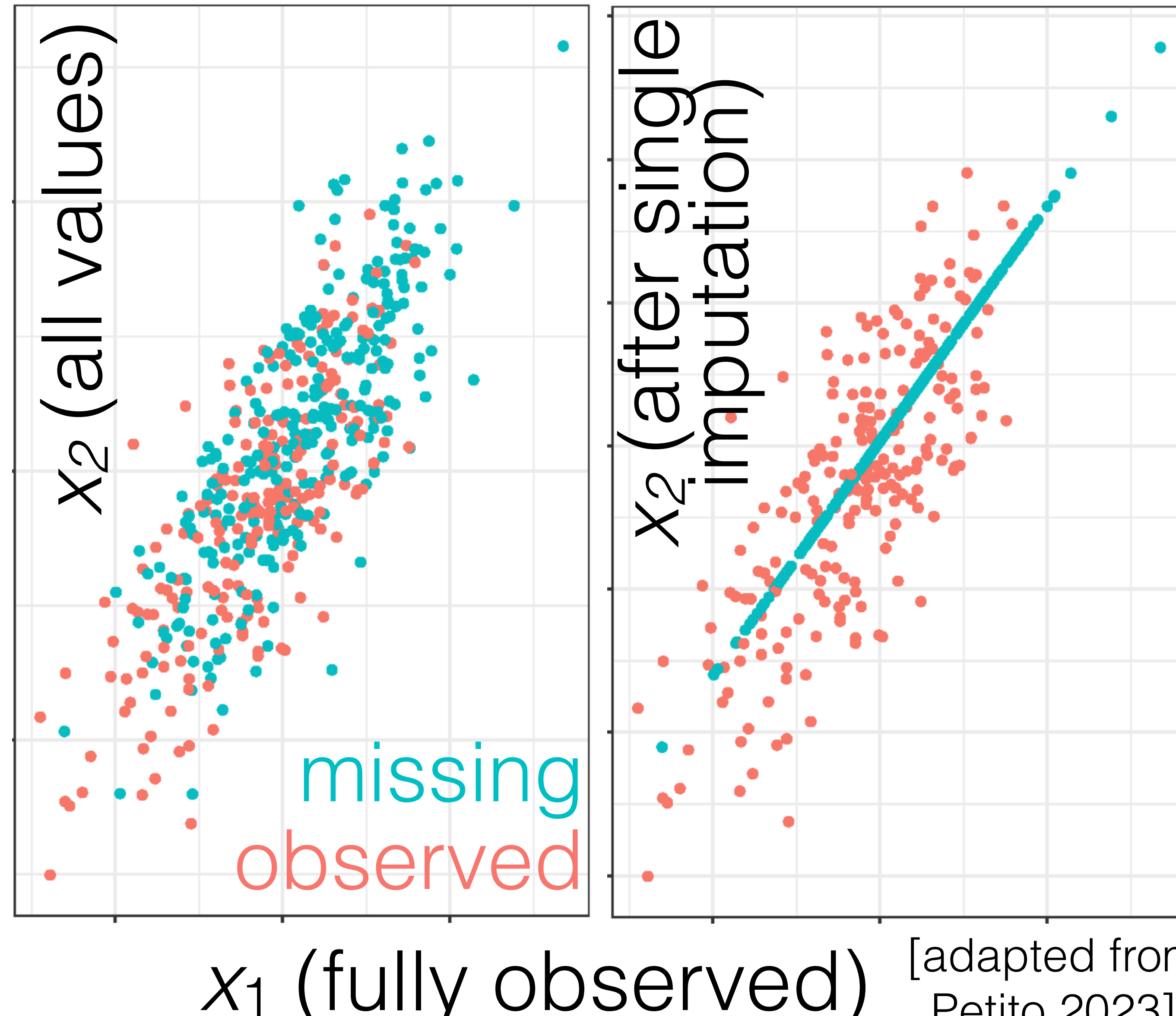


[adapted from Petito 2023]

- Con of methods above: no uncertainty in missing values
- Option: multiple imputation

What to do?

- Option: Single imputation, i.e. replace missing value
  - E.g. use mean or median of observed values (cf. MCAR)
  - E.g. use random sample from observed values (cf. MCAR)
  - E.g. regression estimate of missing features from other features (cf. MAR)

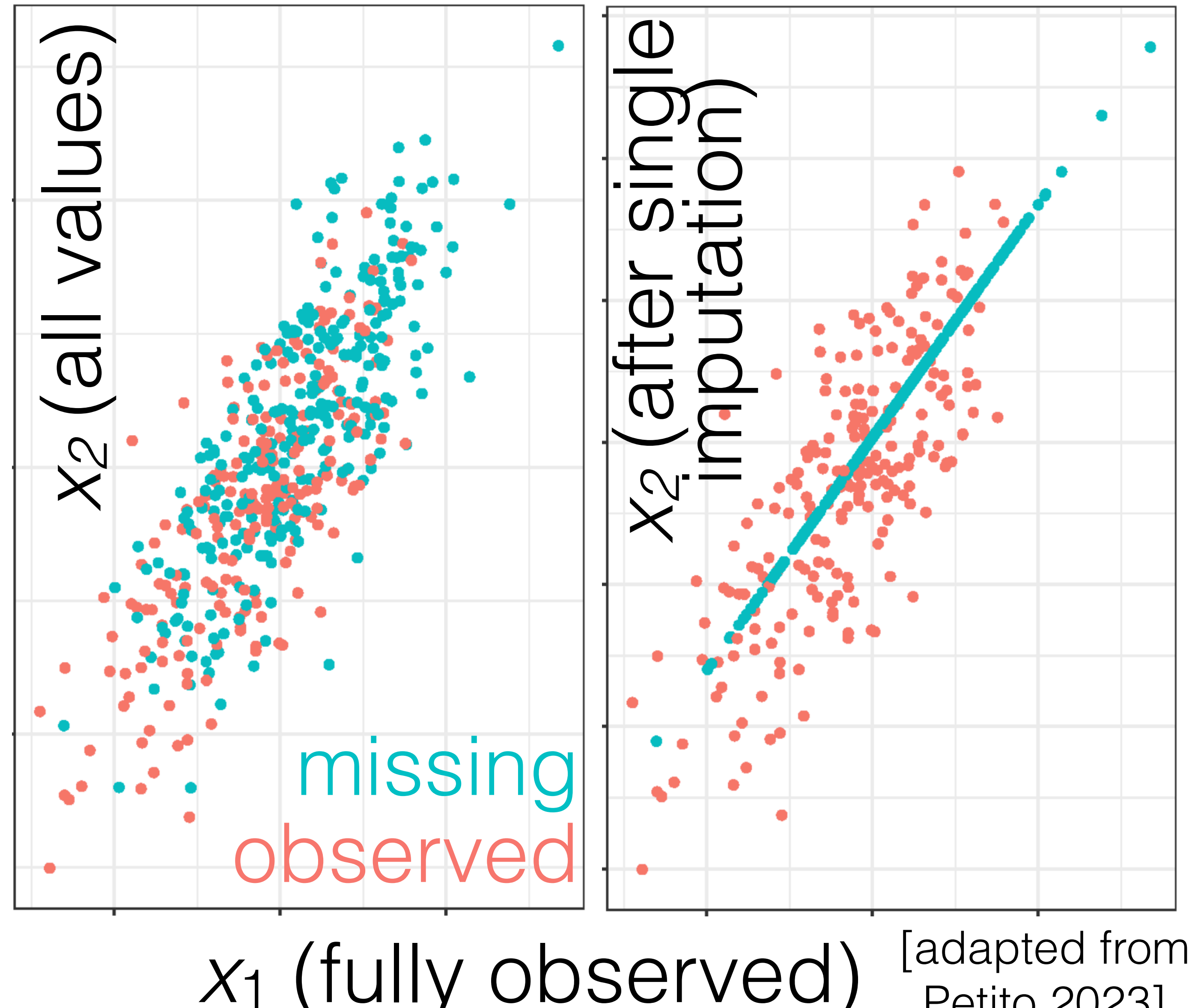


[adapted from Petito 2023]

- Con of methods above: no uncertainty in missing values
- Option: multiple imputation
- Option: full Bayesian model, including over missing values

What to do?

- Option: Single imputation, i.e. replace missing value
  - E.g. use mean or median of observed values (cf. MCAR)
  - E.g. use random sample from observed values (cf. MCAR)
  - E.g. regression estimate of missing features from other features (cf. MAR)



[adapted from Petito 2023]

- Con of methods above: no uncertainty in missing values
- Option: multiple imputation
- Option: full Bayesian model, including over missing values
- Con of methods above: more work for the data analyst

- Sometimes it seems like we don't have enough data

- Sometimes it seems like we don’t have enough data
    - Maybe we really don’t have enough data for what we want to learn, or maybe we can work with what we have

- Sometimes it seems like we don’t have enough data
  - Maybe we really don’t have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data

- Sometimes it seems like we don’t have enough data
  - Maybe we really don’t have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?

- Sometimes it seems like we don’t have enough data
  - Maybe we really don’t have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features

- Sometimes it seems like we don't have enough data
  - Maybe we really don't have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features
  - Common to have many features

- Sometimes it seems like we don’t have enough data
  - Maybe we really don’t have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features
  - Common to have many features
    - Could have data on millions of genetic variants

- Sometimes it seems like we don't have enough data
  - Maybe we really don't have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features
  - Common to have many features
    - Could have data on millions of genetic variants
    - Health records (heart rate, blood pressure, etc etc.)

- Sometimes it seems like we don't have enough data
  - Maybe we really don't have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features
  - Common to have many features
    - Could have data on millions of genetic variants
    - Health records (heart rate, blood pressure, etc etc.)
    - Retailer could sell 10s of millions of products

Dimensionality reduction

- Sometimes it seems like we don’t have enough data
  - Maybe we really don’t have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features
  - Common to have many features
    - Could have data on millions of genetic variants
    - Health records (heart rate, blood pressure, etc etc.)
    - Retailer could sell 10s of millions of products

Dimensionality reduction

- Sometimes it seems like we don’t have enough data
  - Maybe we really don’t have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features
  - Common to have many features
    - Could have data on millions of genetic variants
    - Health records (heart rate, blood pressure, etc etc.)
    - Retailer could sell 10s of millions of products
  - Sometimes can use domain knowledge, but often not

Dimensionality reduction

- Sometimes it seems like we don't have enough data
  - Maybe we really don't have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features
  - Common to have many features
    - Could have data on millions of genetic variants
    - Health records (heart rate, blood pressure, etc etc.)
    - Retailer could sell 10s of millions of products
  - Sometimes can use domain knowledge, but often not
  - Multiple options for algorithmic dimensionality reduction.

Dimensionality reduction

- Sometimes it seems like we don't have enough data
  - Maybe we really don't have enough data for what we want to learn, or maybe we can work with what we have
- Sometimes it seems like we have too much data
  - Visualizing data is always a great idea, but what if I have more than two features?
  - We saw that methods like GP regression (with squared exponential kernel) might struggle with too many features
  - Common to have many features
    - Could have data on millions of genetic variants
    - Health records (heart rate, blood pressure, etc etc.)
    - Retailer could sell 10s of millions of products
  - Sometimes can use domain knowledge, but often not
  - Multiple options for algorithmic dimensionality reduction. "As of April 2022, 32,000-216,000 genetic papers employed PC scatterplots"

[Elhaik 2022; not clear where these numbers come from though]

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)

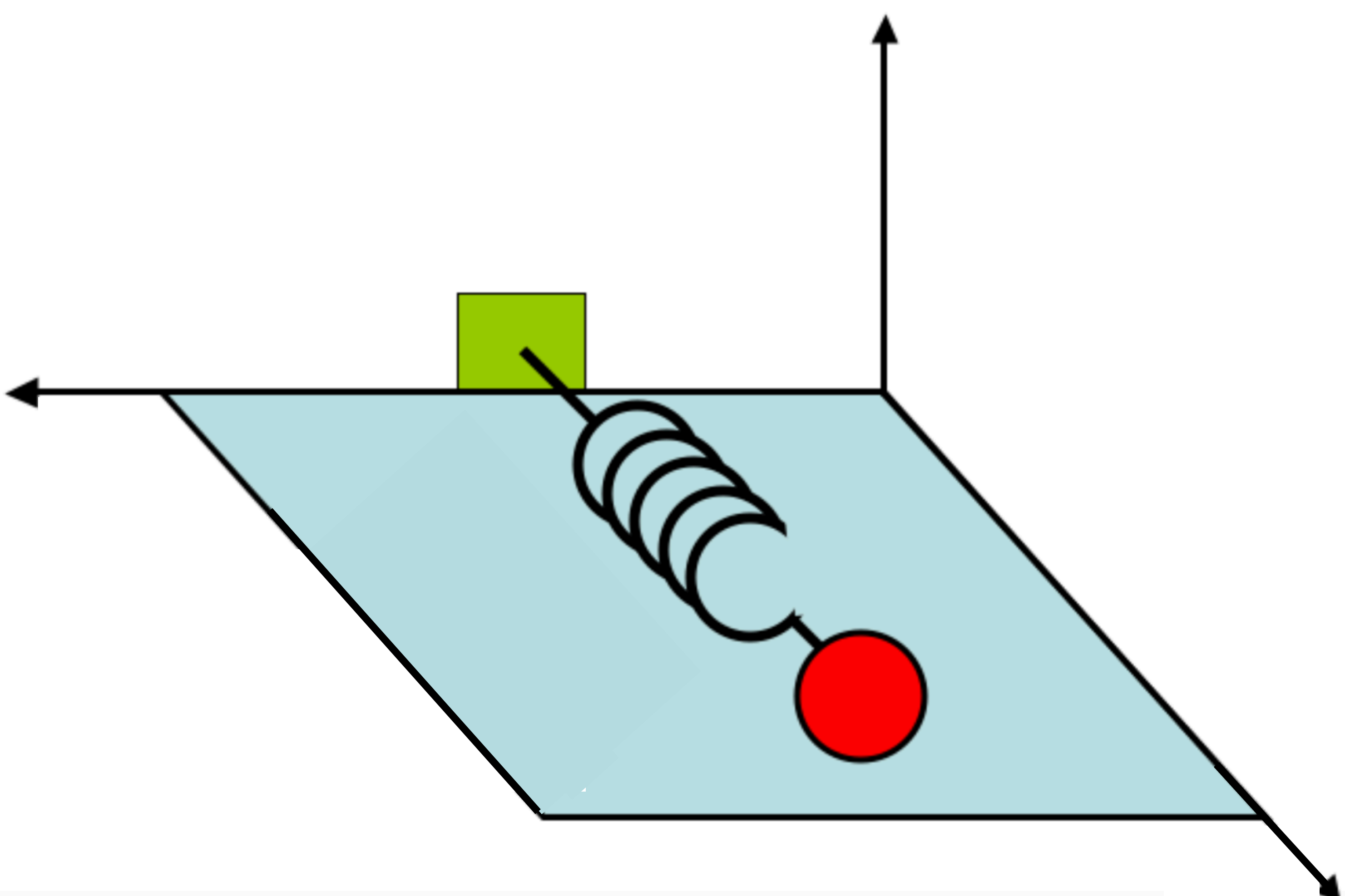
[Example thanks to Schlens 2014]

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

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.

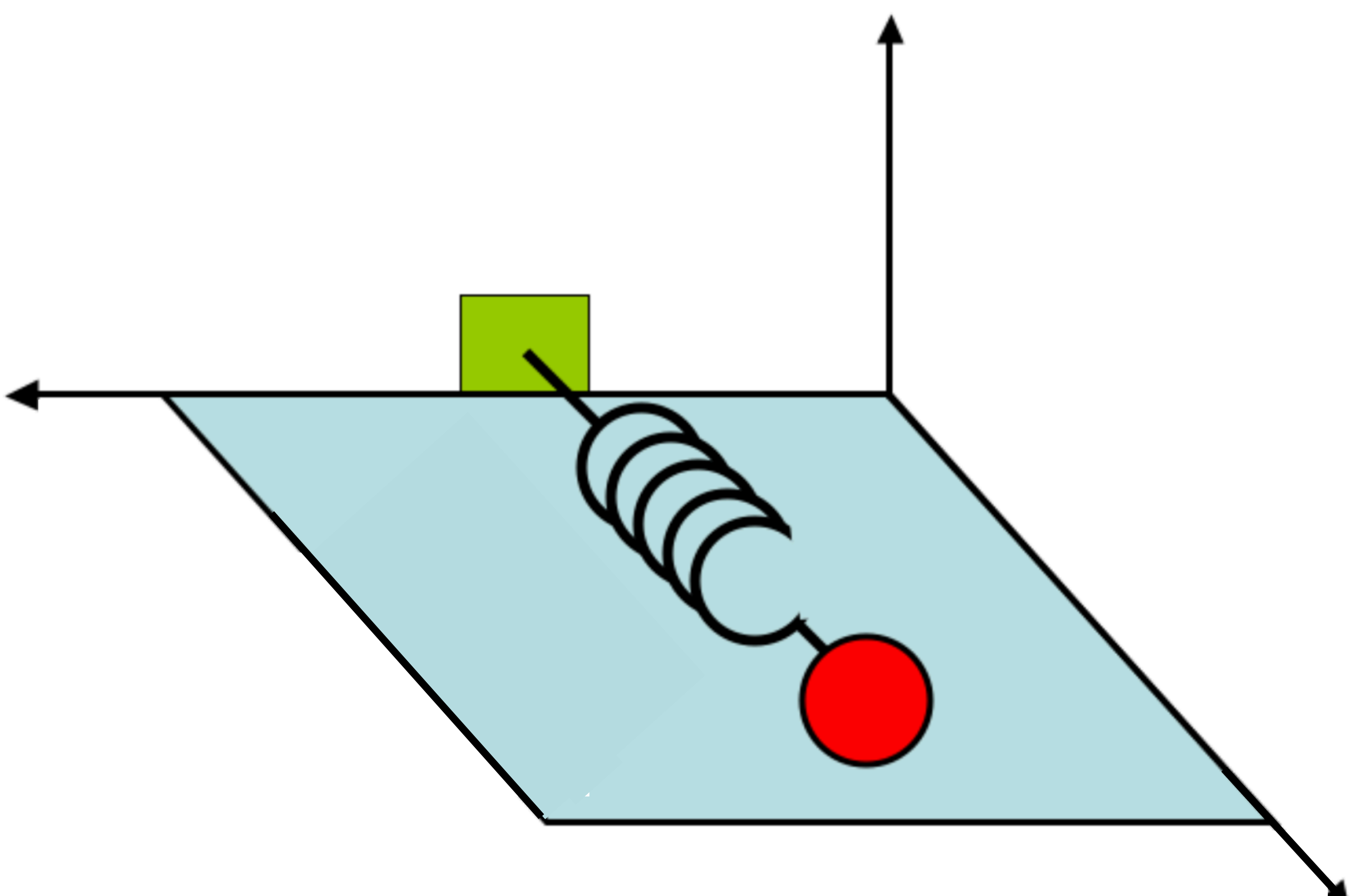


[Adapted from Schlens 2014]

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.

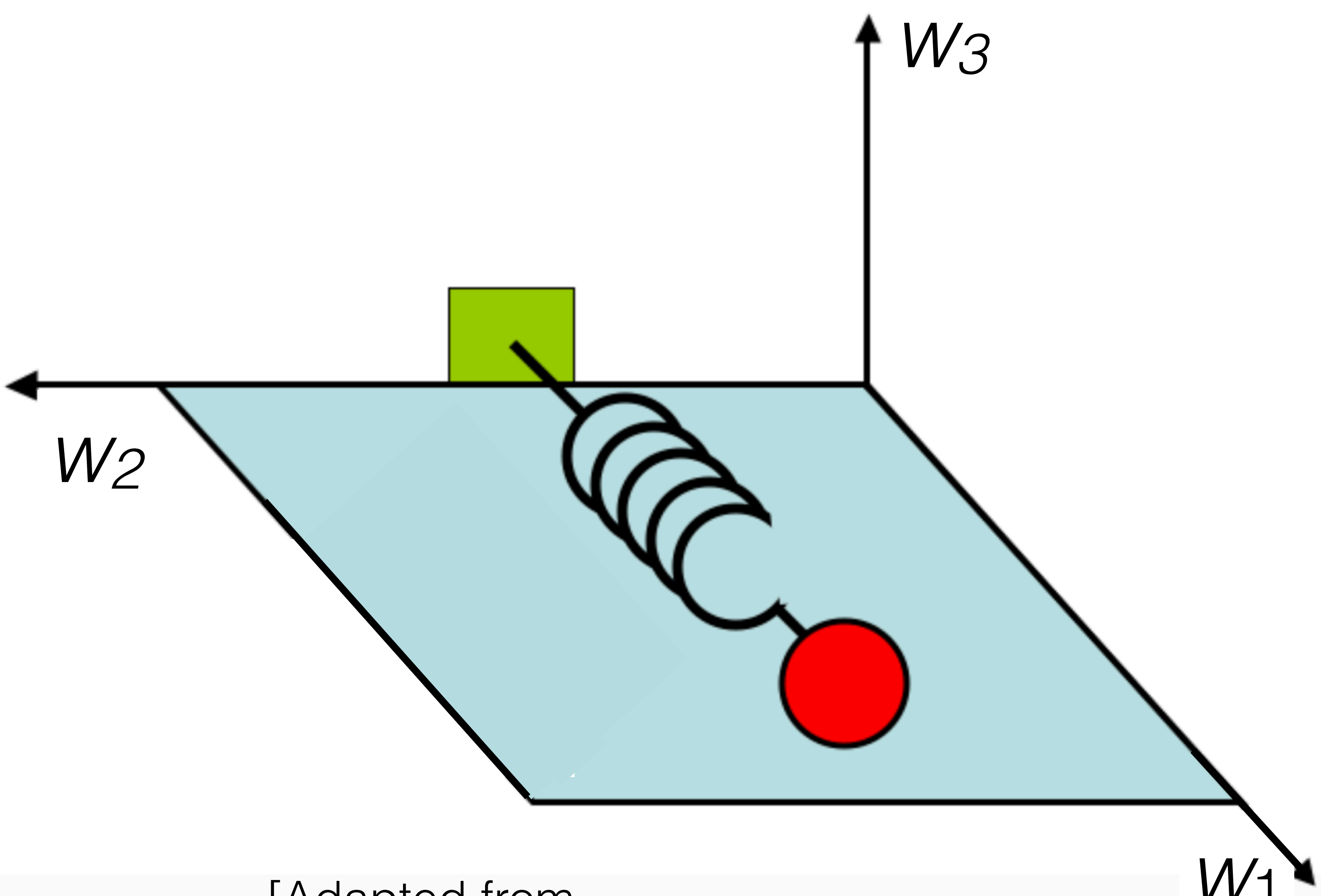


[Adapted from Schlens 2014]

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.

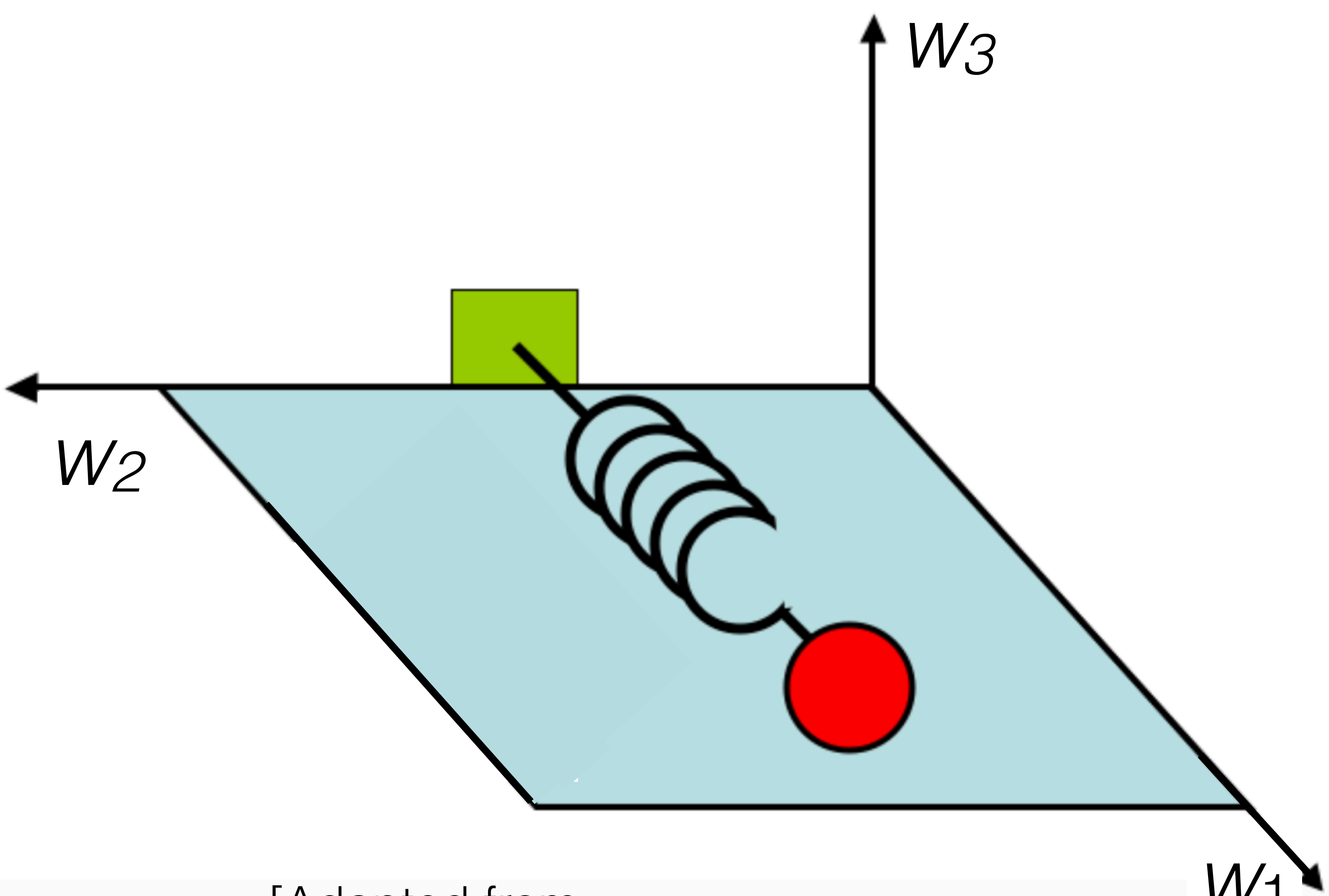


[Adapted from Schlens 2014]

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.

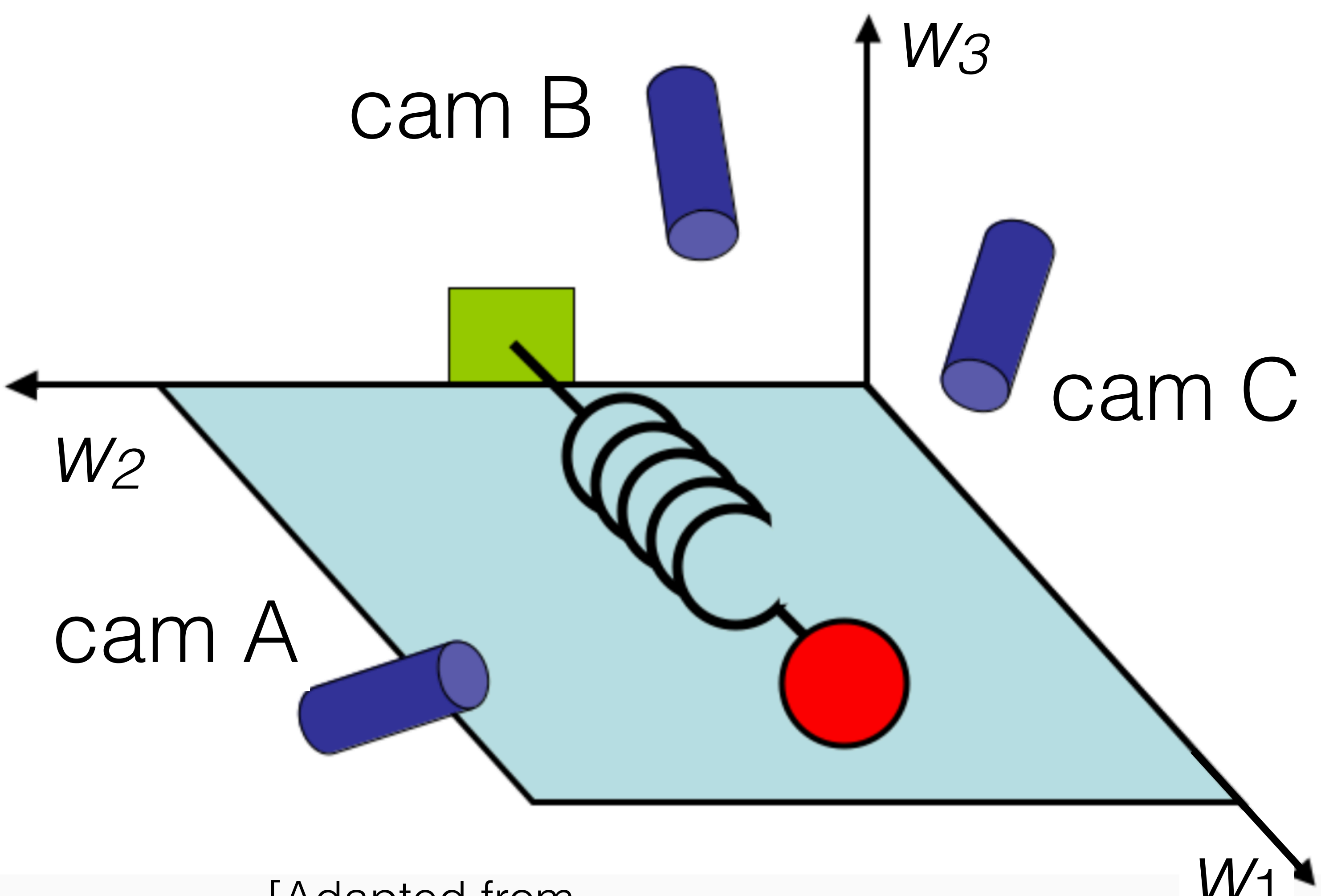


[Adapted from Schlens 2014]

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.

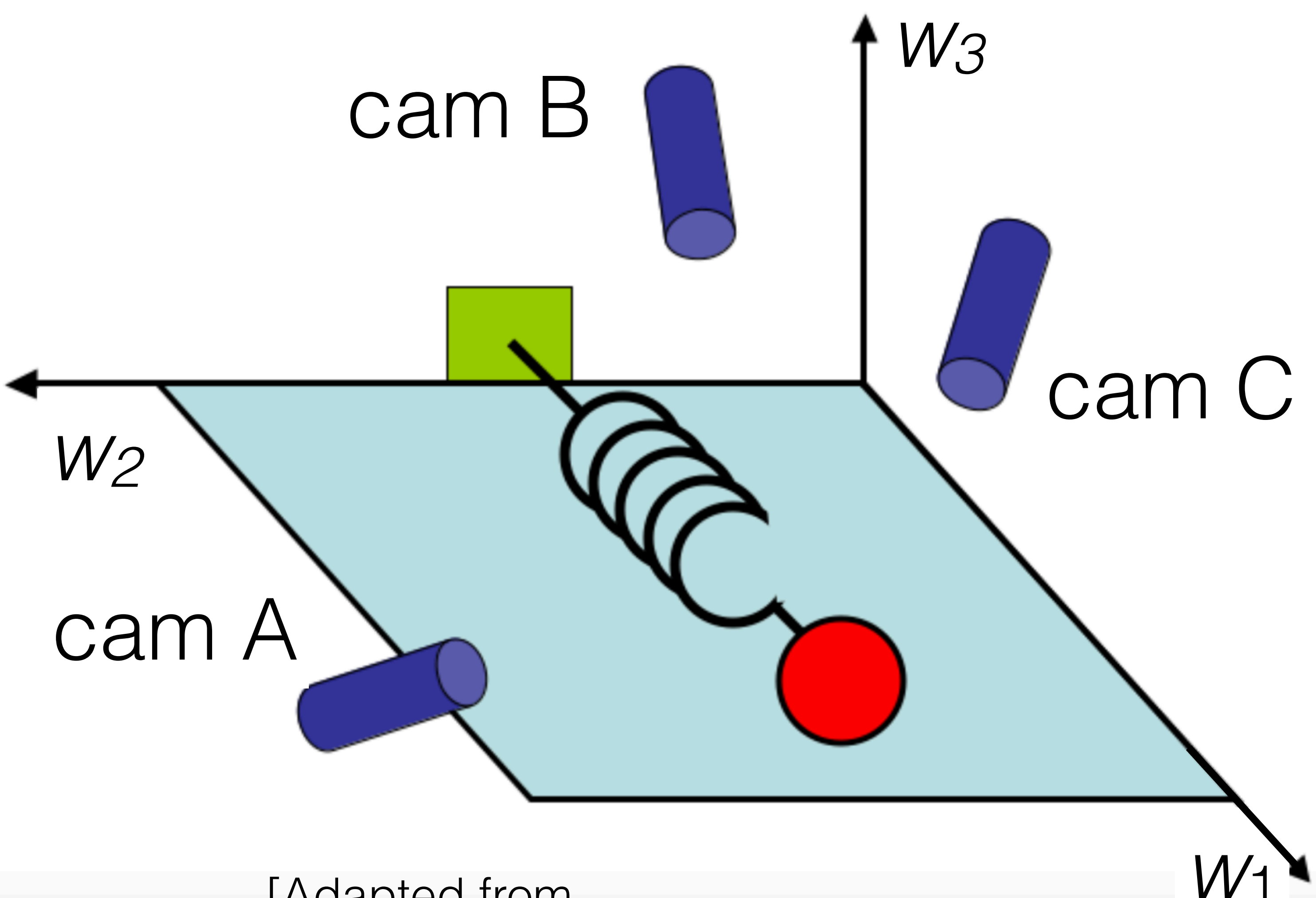


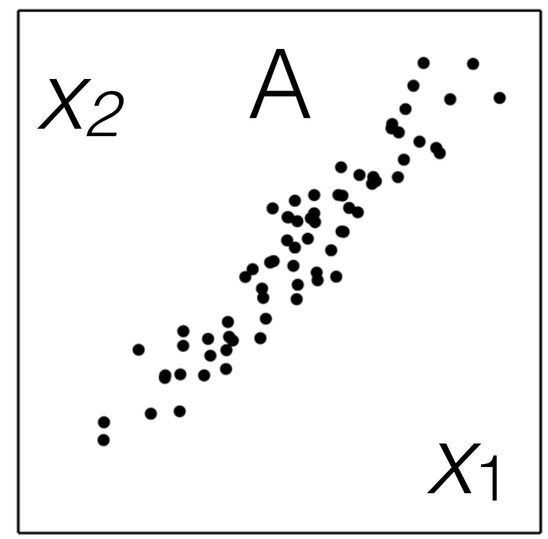
[Adapted from Schlens 2014]

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.



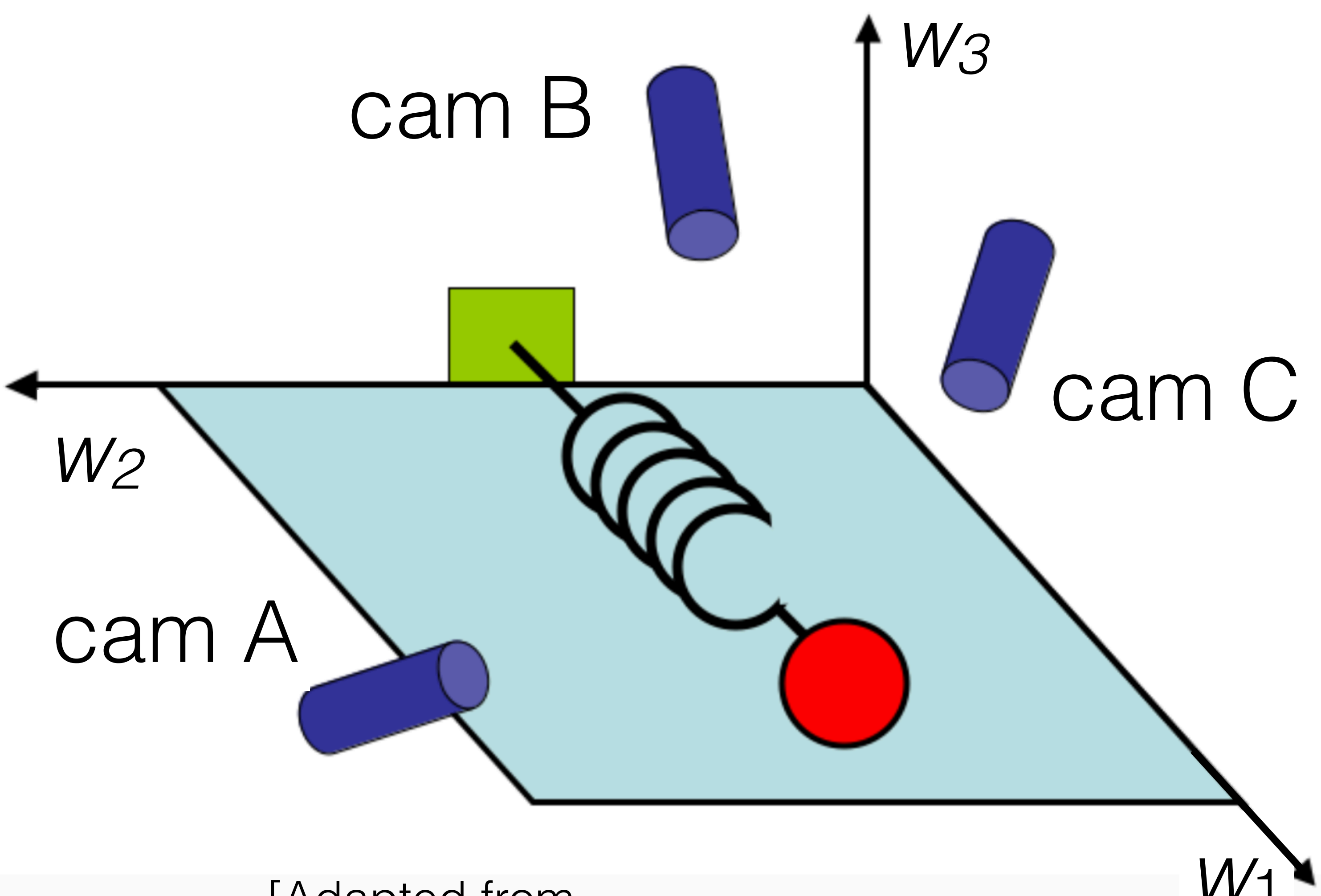


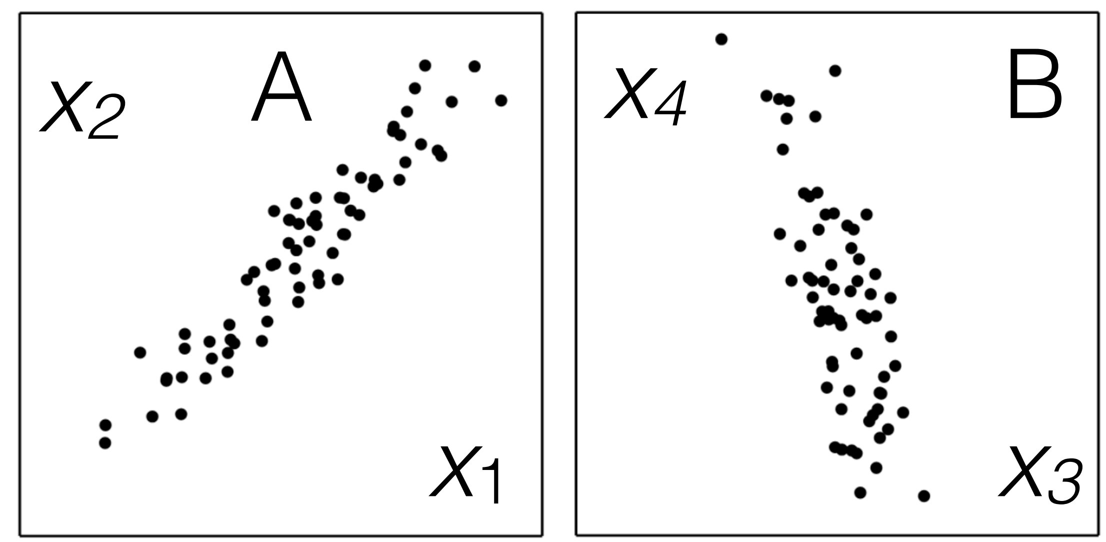
[Adapted from Schlens 2014]

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.



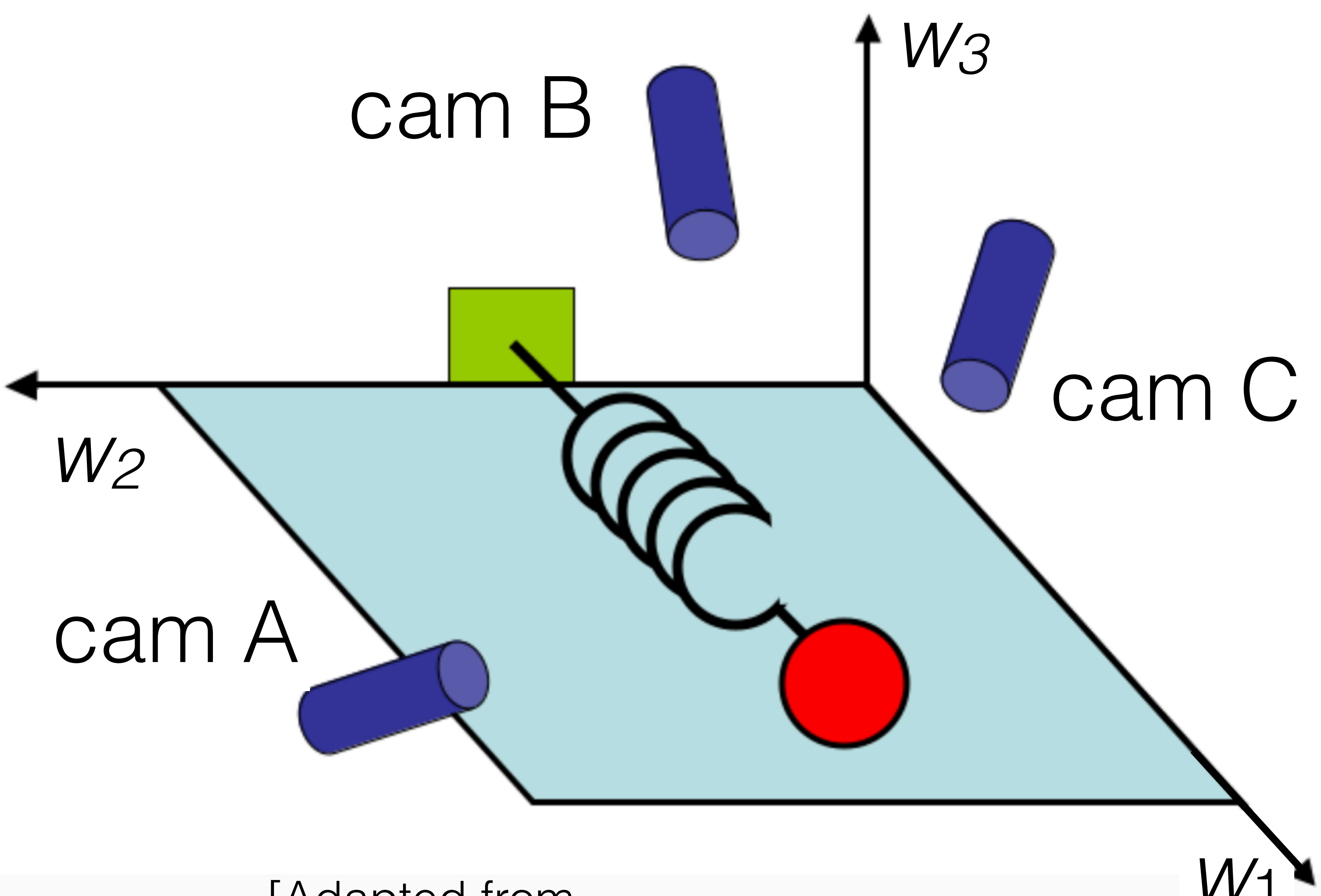


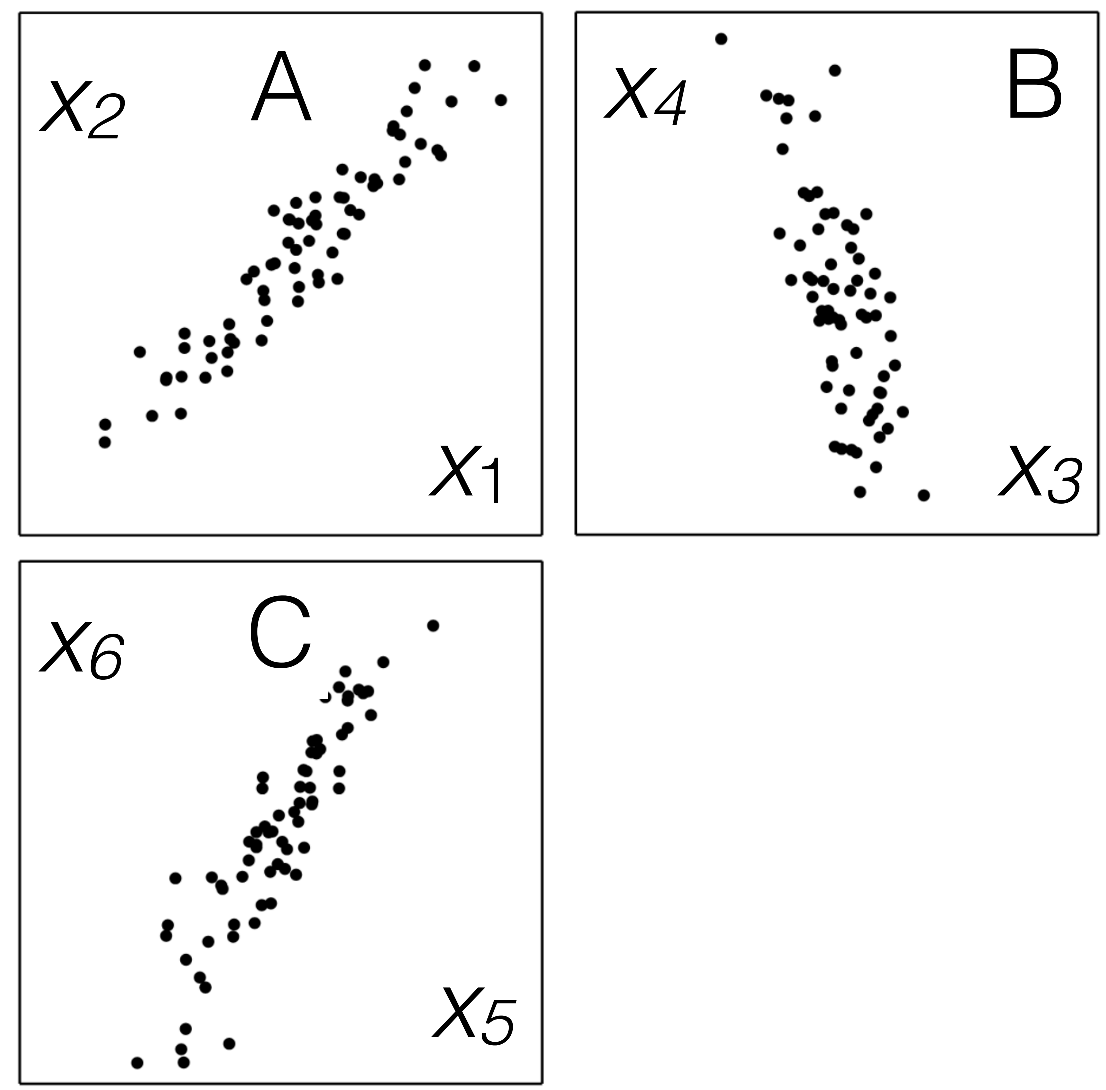
[Adapted from Schlens 2014]

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.



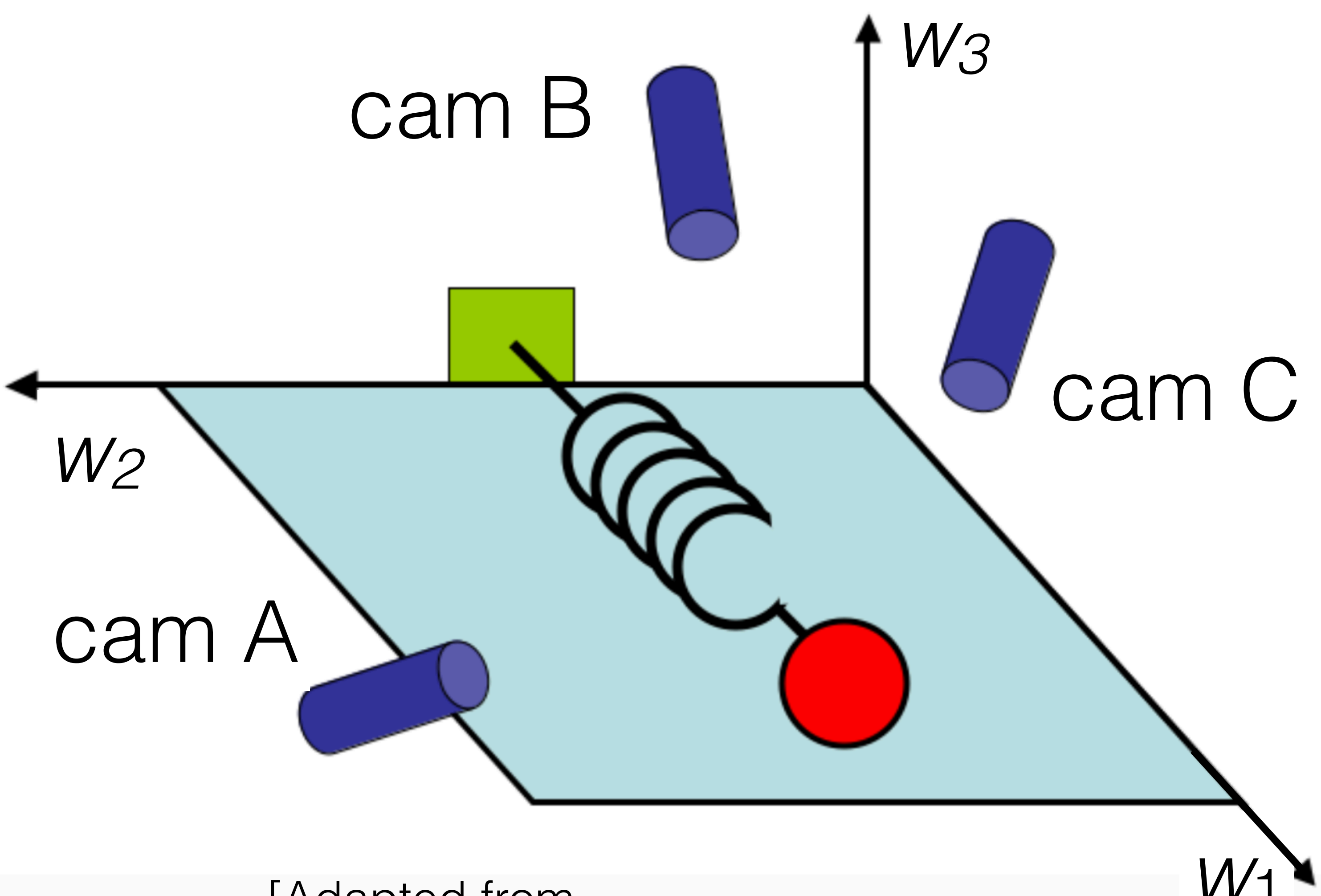


[Adapted from Schlens 2014]

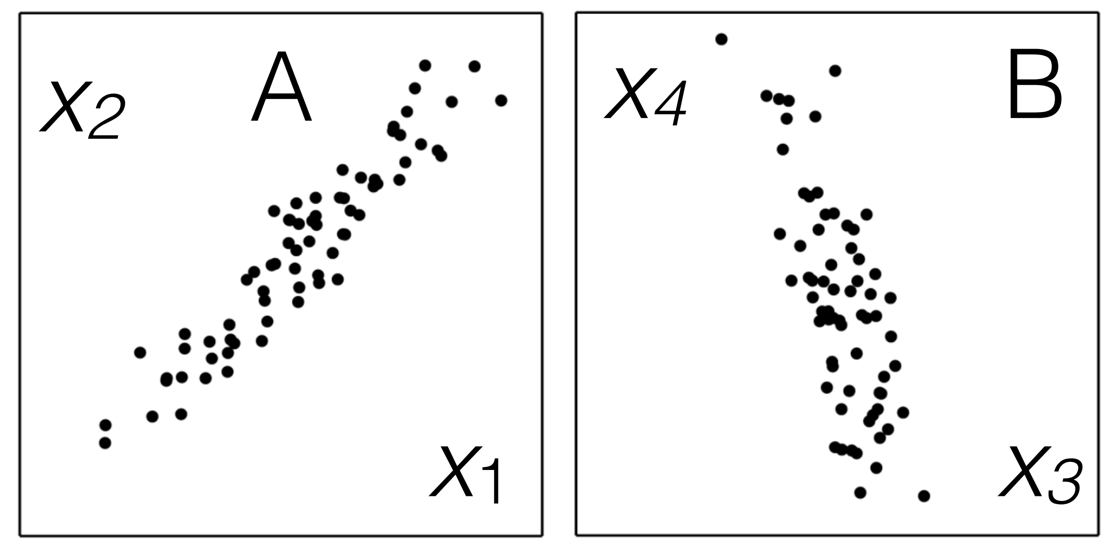
Motivating example for Principal Components Analysis (PCA)

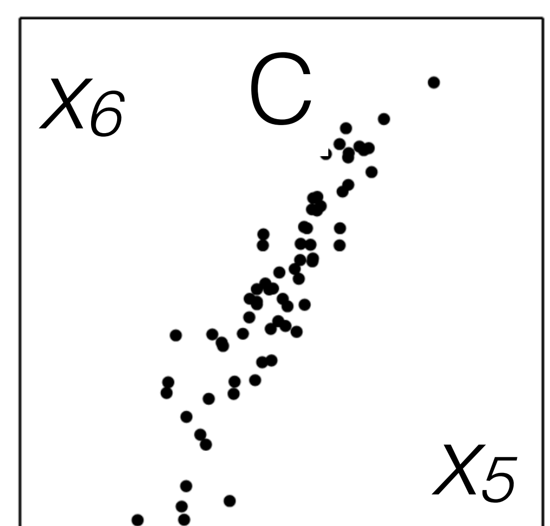
[Example thanks to Schlens 2014]

- 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.



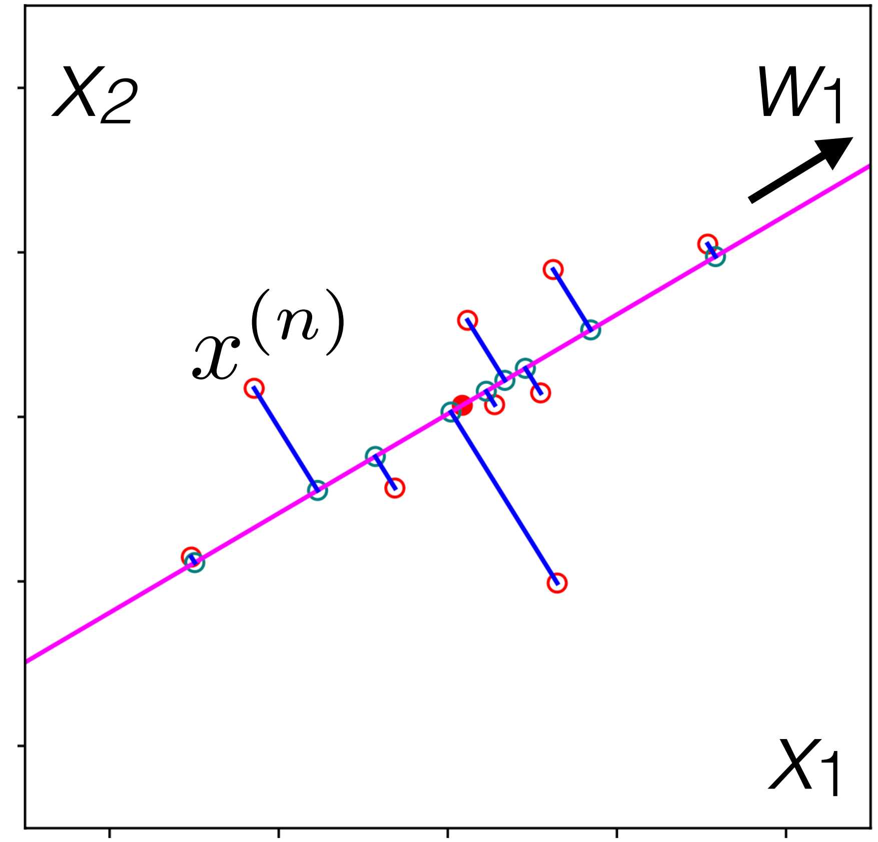
[Adapted from Schlens 2014]





- We have 6 features, but need only 1

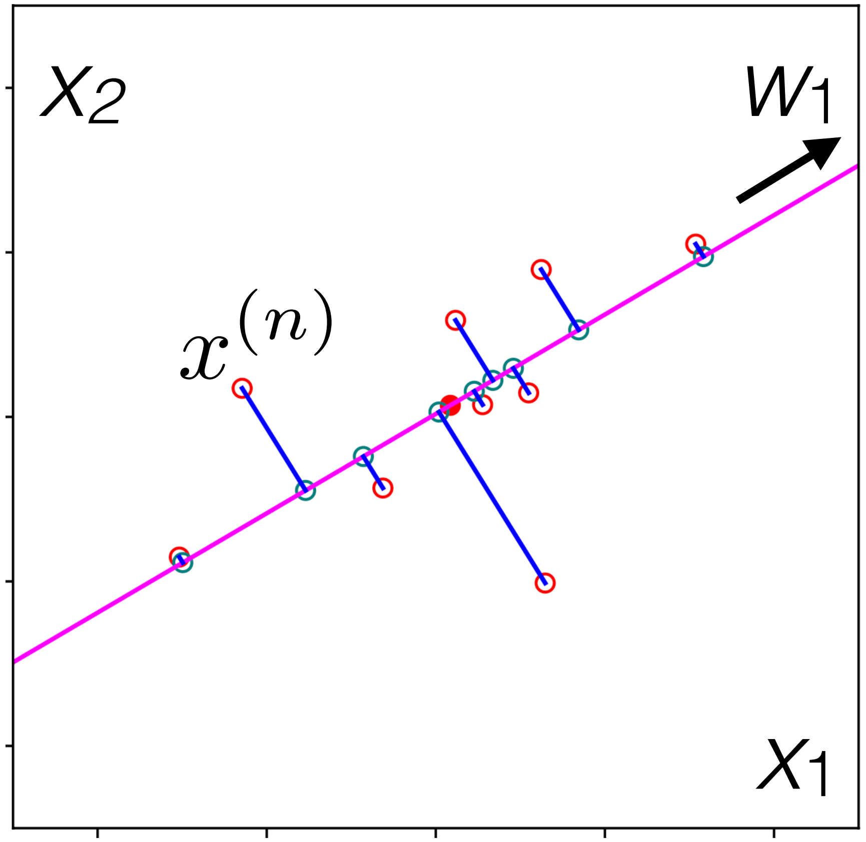
Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

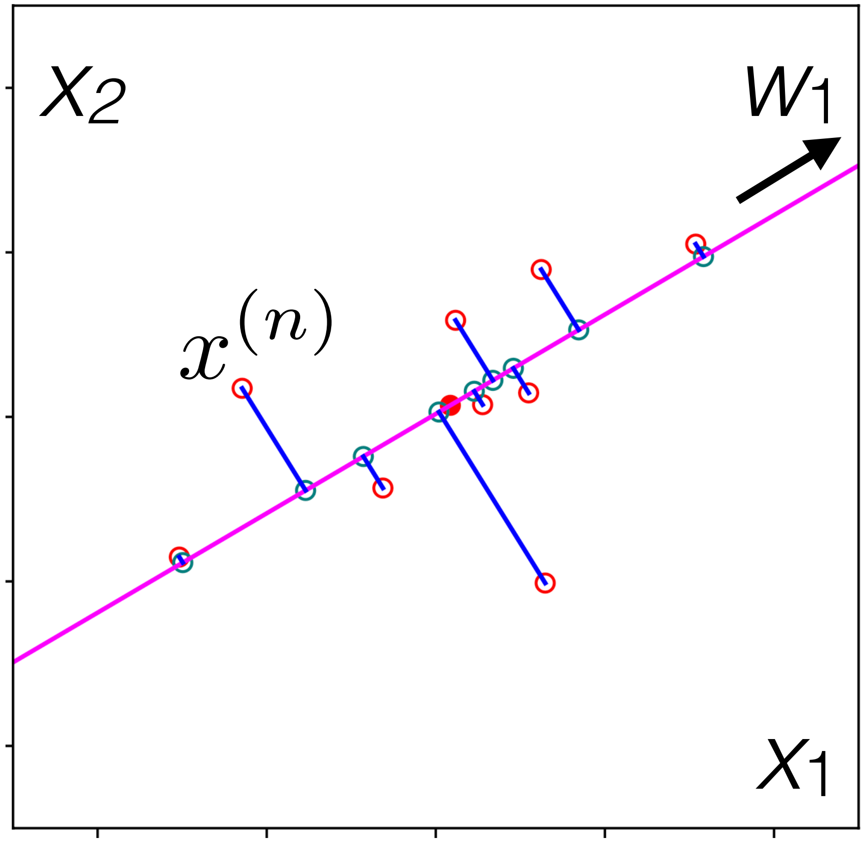
- As usual $x^{(n)}$ is a Dx1 vector



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

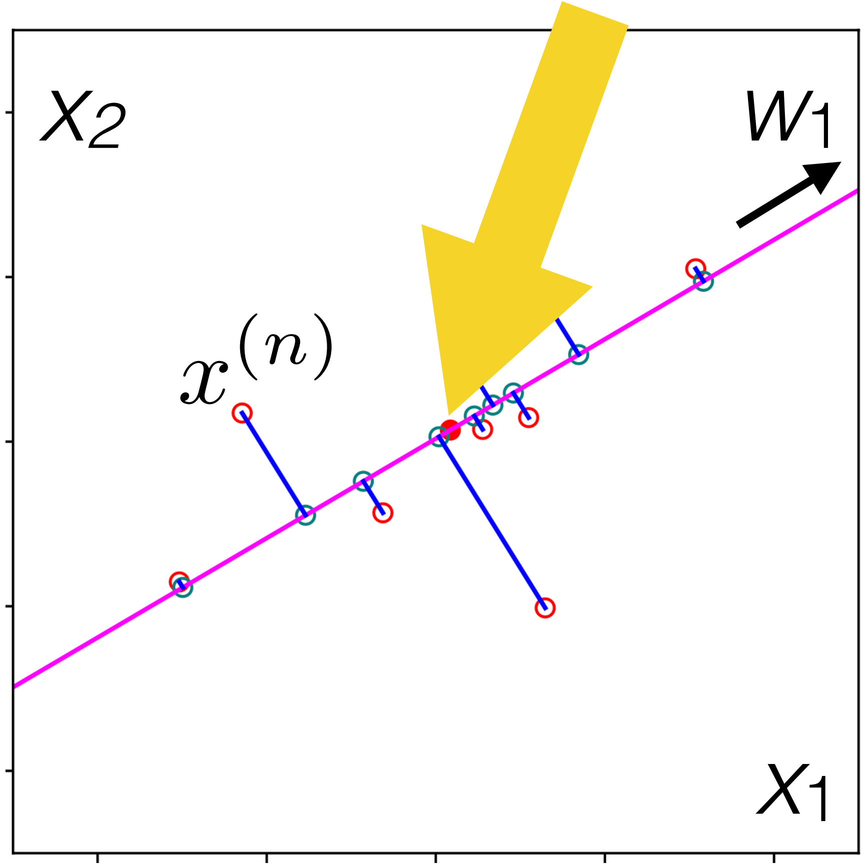
- As usual $x^{(n)}$ is a Dx1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

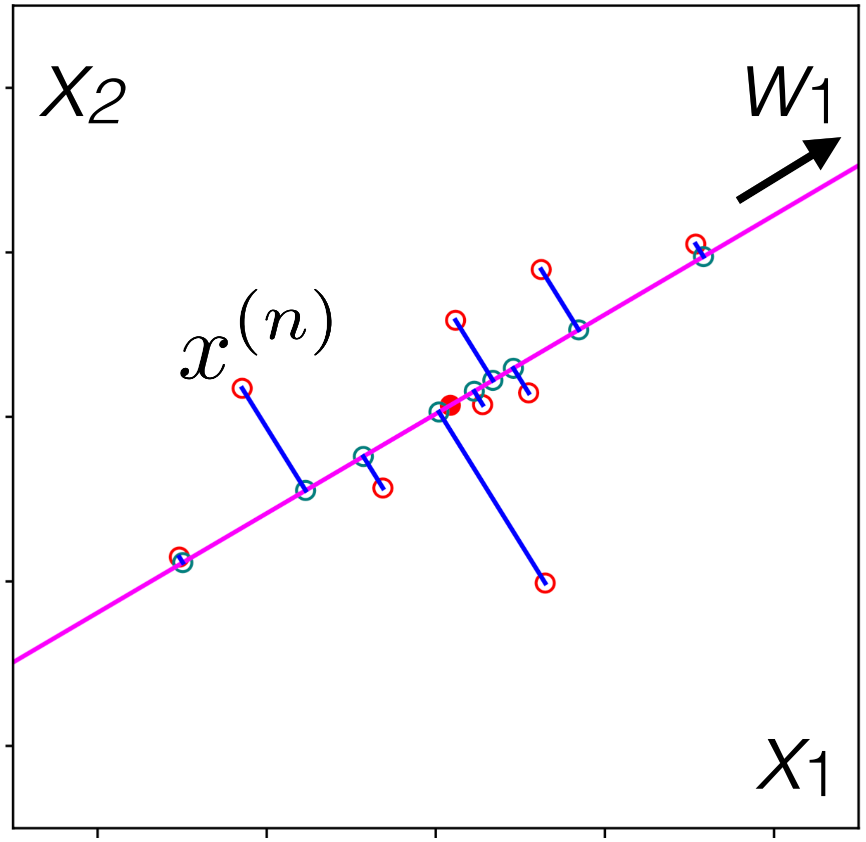
- As usual $x^{(n)}$ is a Dx1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

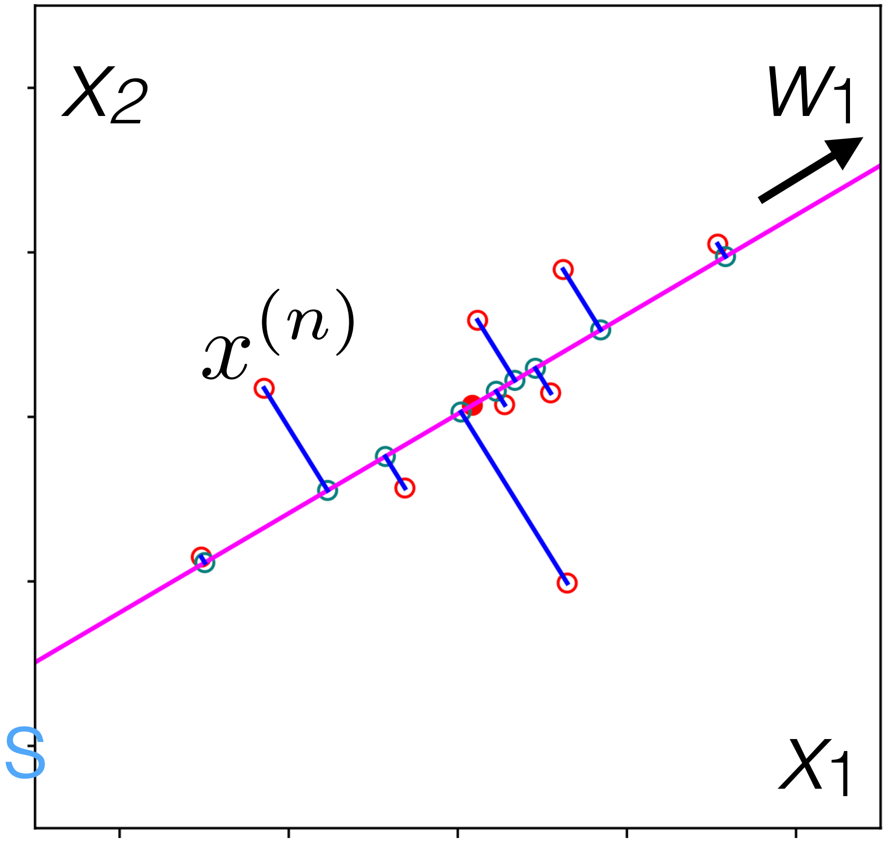


[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

- As usual $x^{(n)}$ is a Dx1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

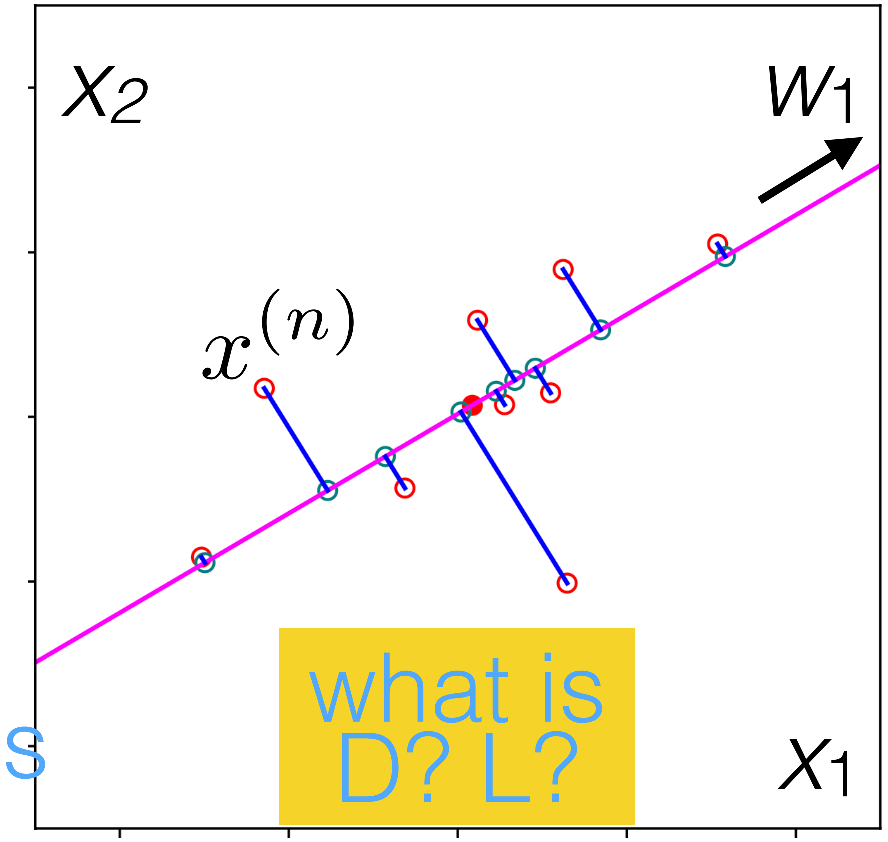


[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

- As usual $x^{(n)}$ is a Dx1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components



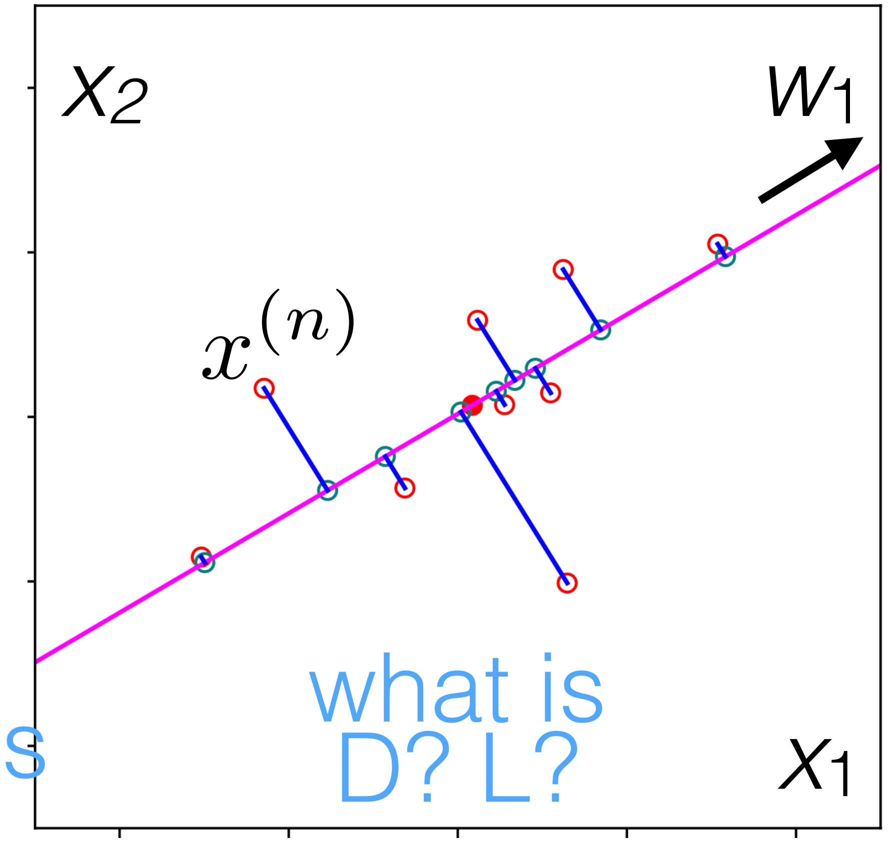
[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

$x^{(n)}$
Dx1



[adapted from Murphy 2022, Fig 20.1]

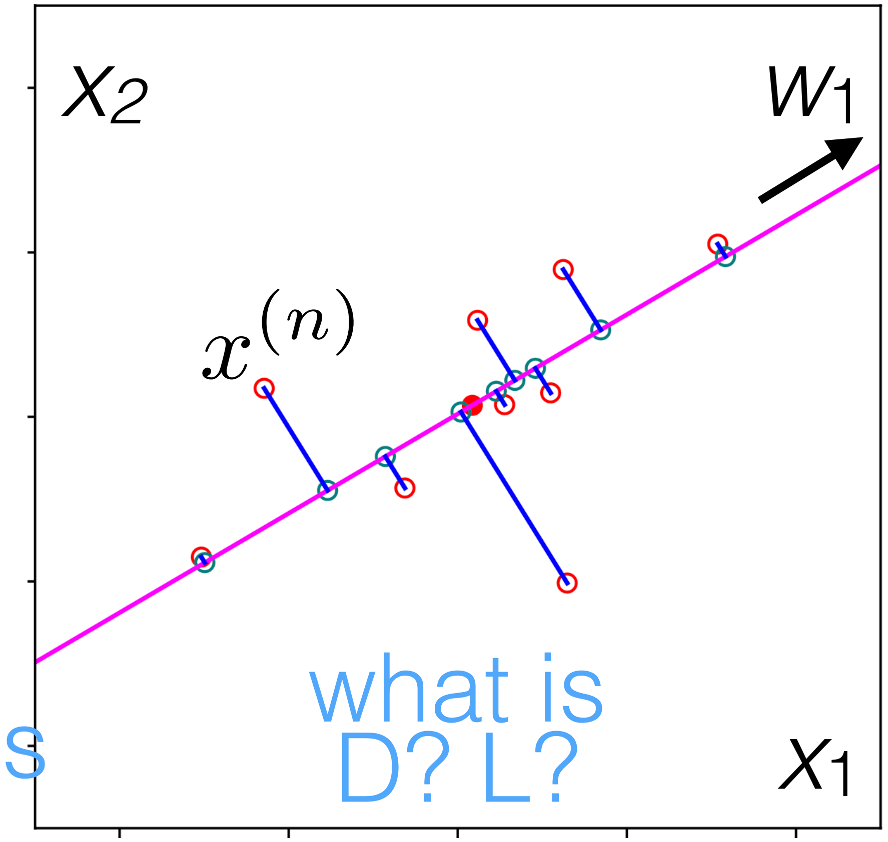
Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1



[adapted from Murphy 2022, Fig 20.1]

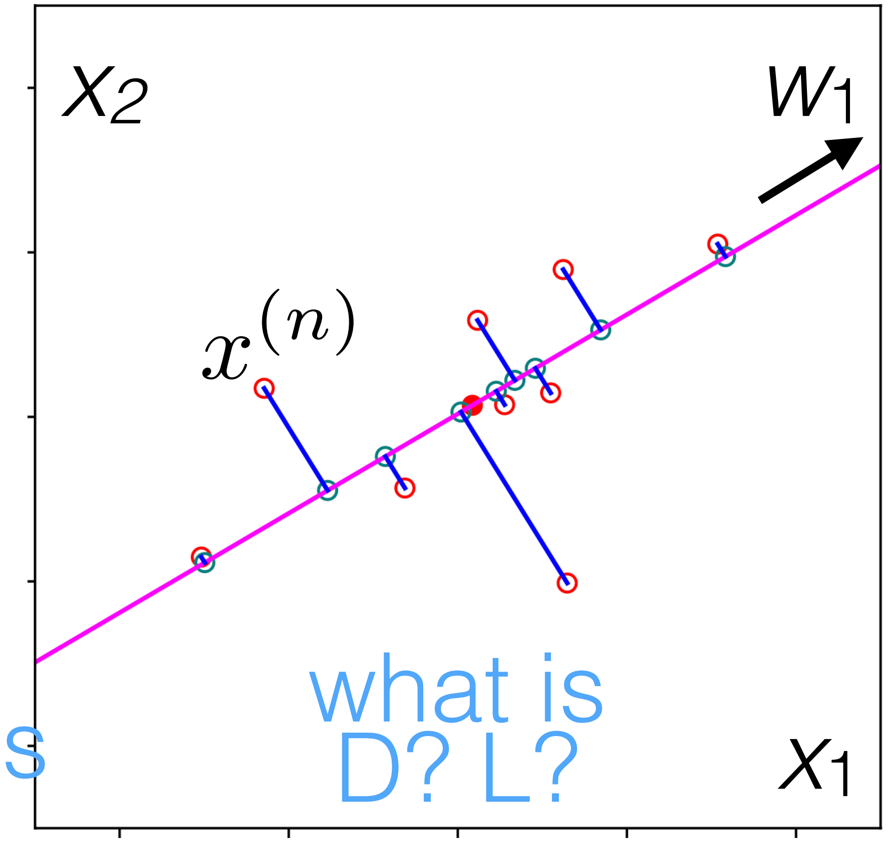
Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1 weights:1x1 Dx1



[adapted from Murphy 2022, Fig 20.1]

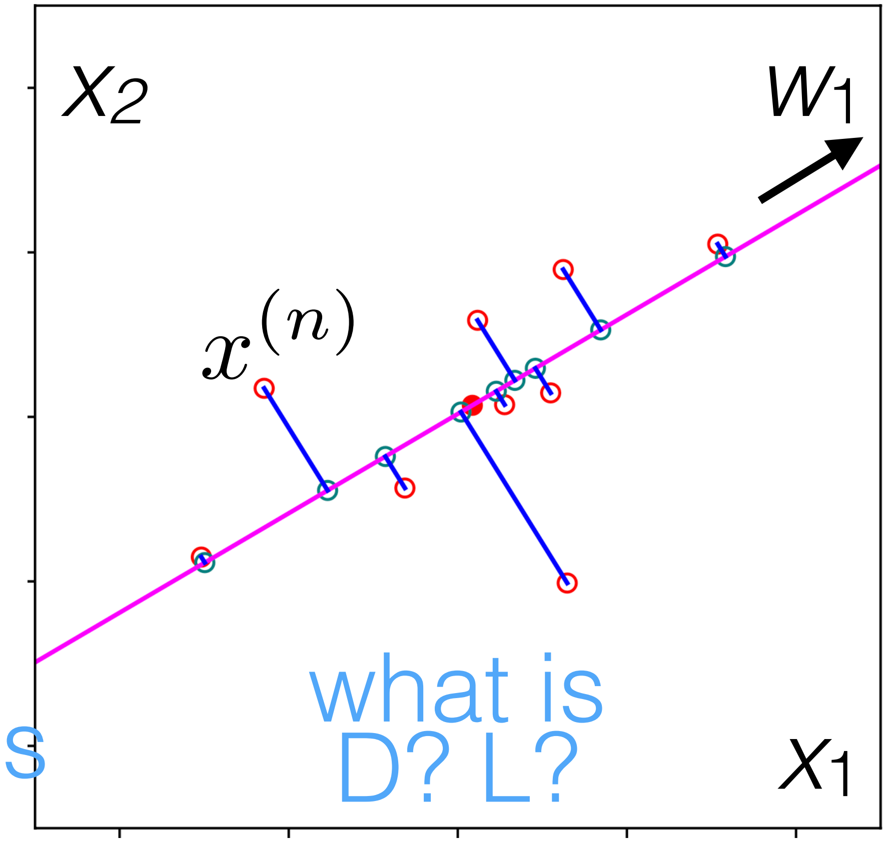
Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1 weights:1x1 Dx1



[adapted from Murphy 2022, Fig 20.1]

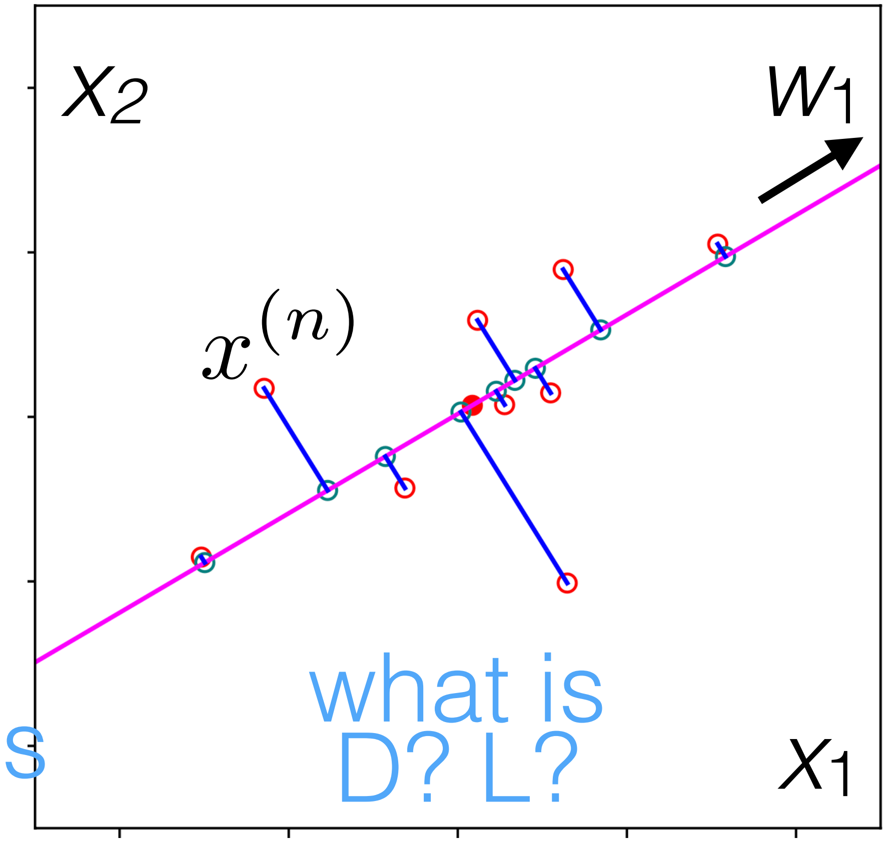
Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1



[adapted from Murphy 2022, Fig 20.1]

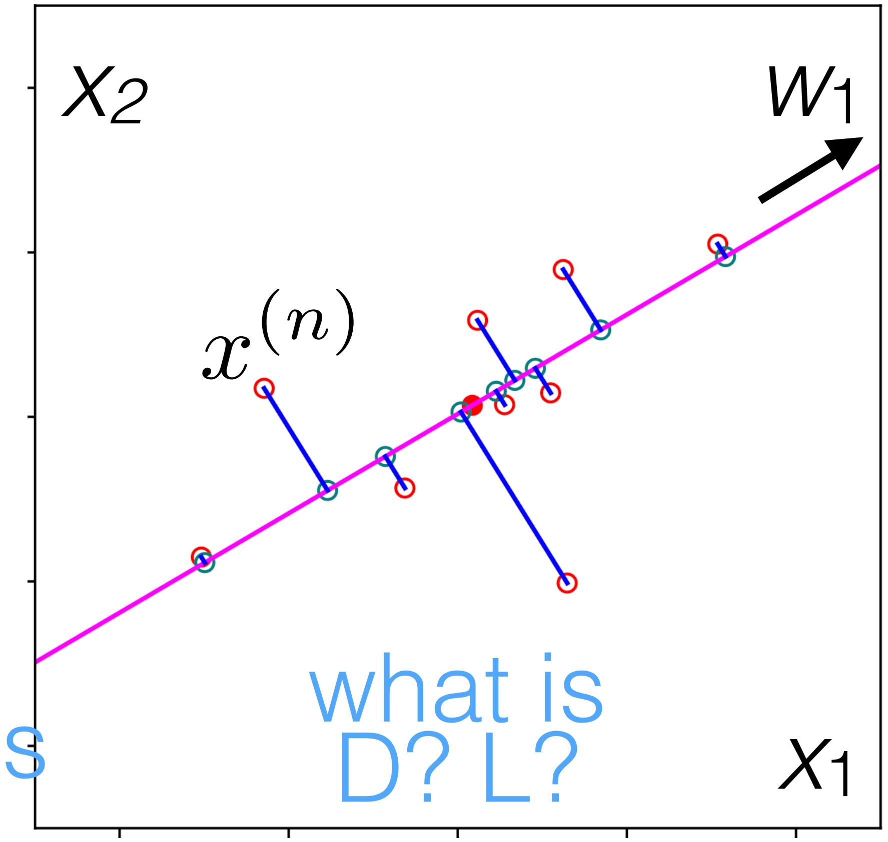
Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1



[adapted from Murphy 2022, Fig 20.1]

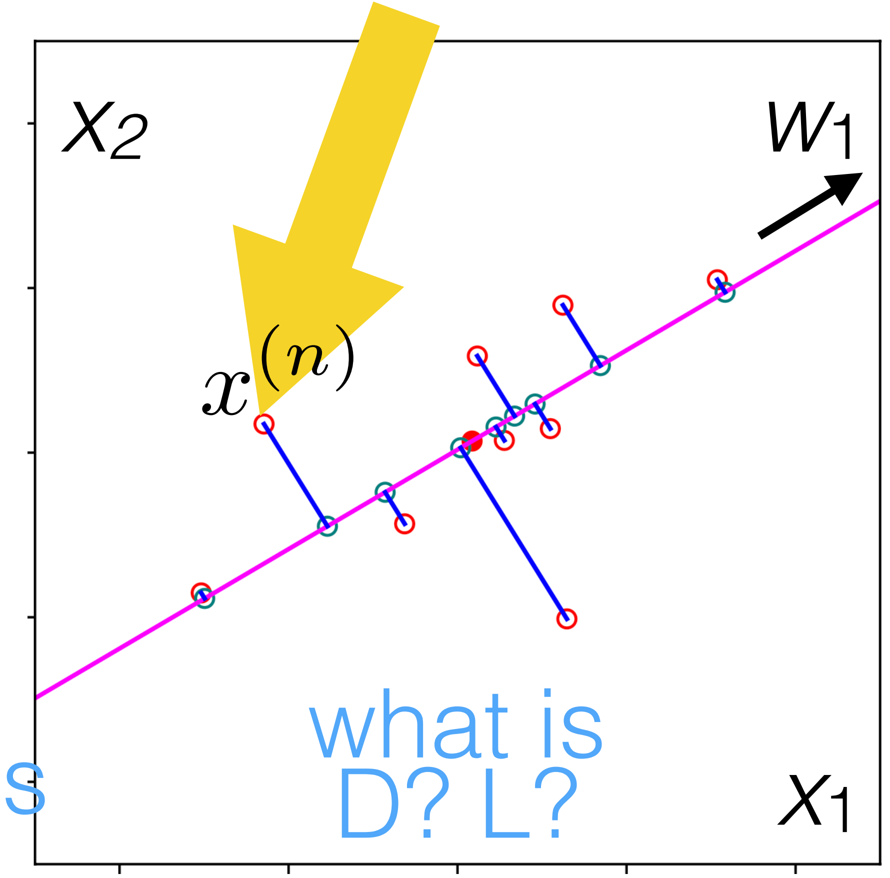
Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1



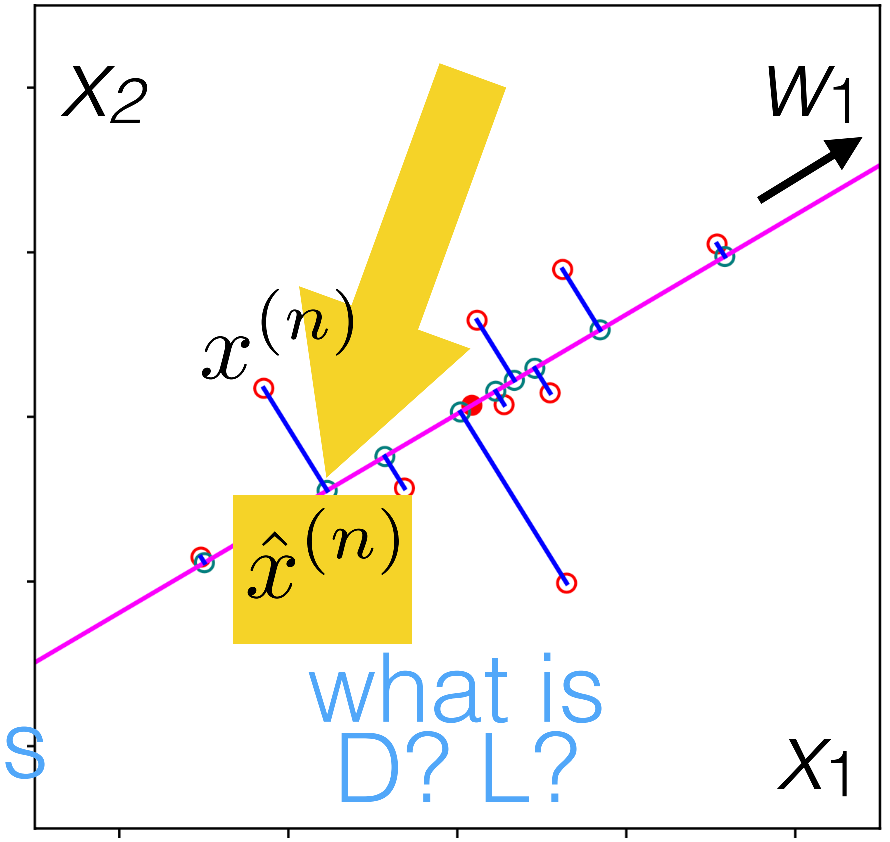
[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

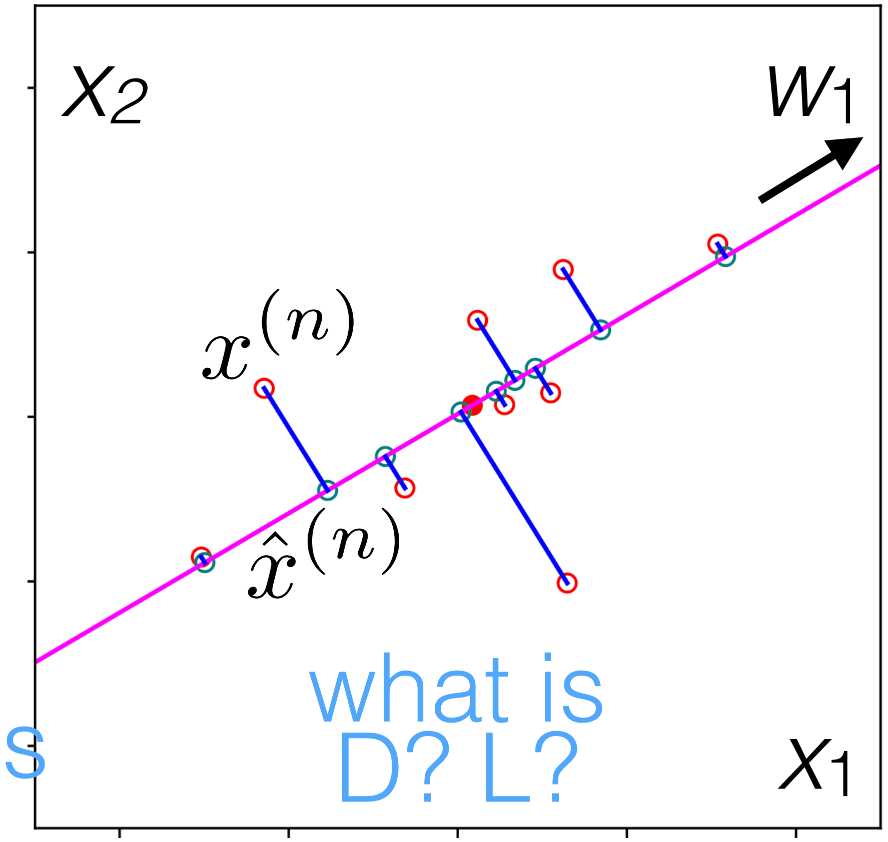
- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1

no offset (just rotating basis)



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

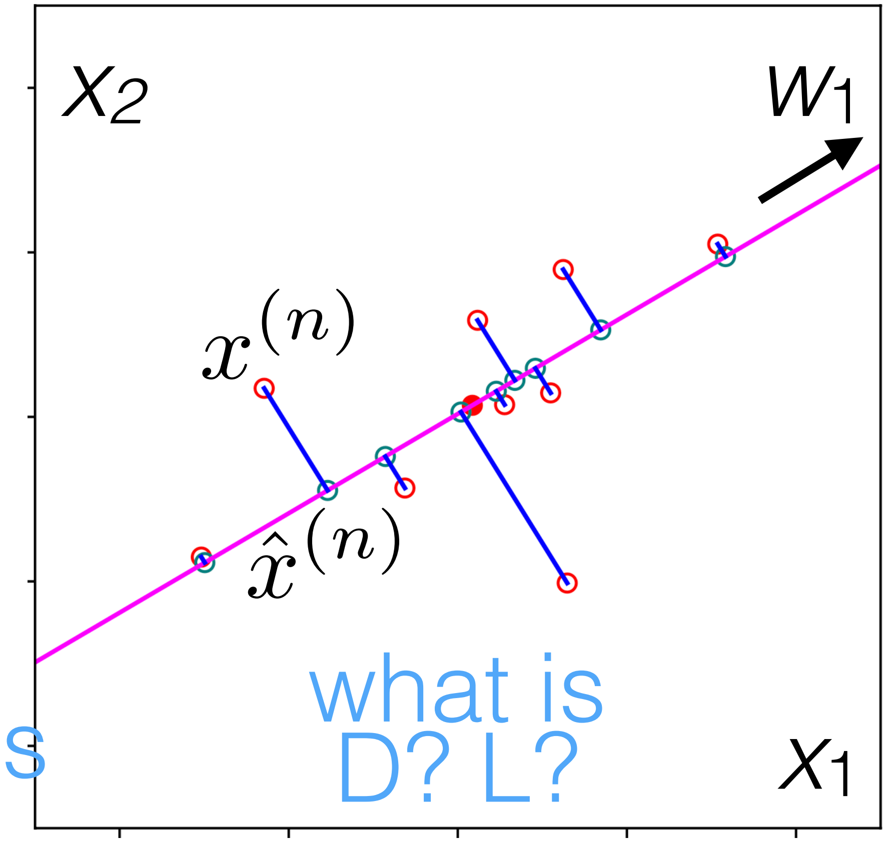
- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

- Goal: projection is "close" to original data (square loss)



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA

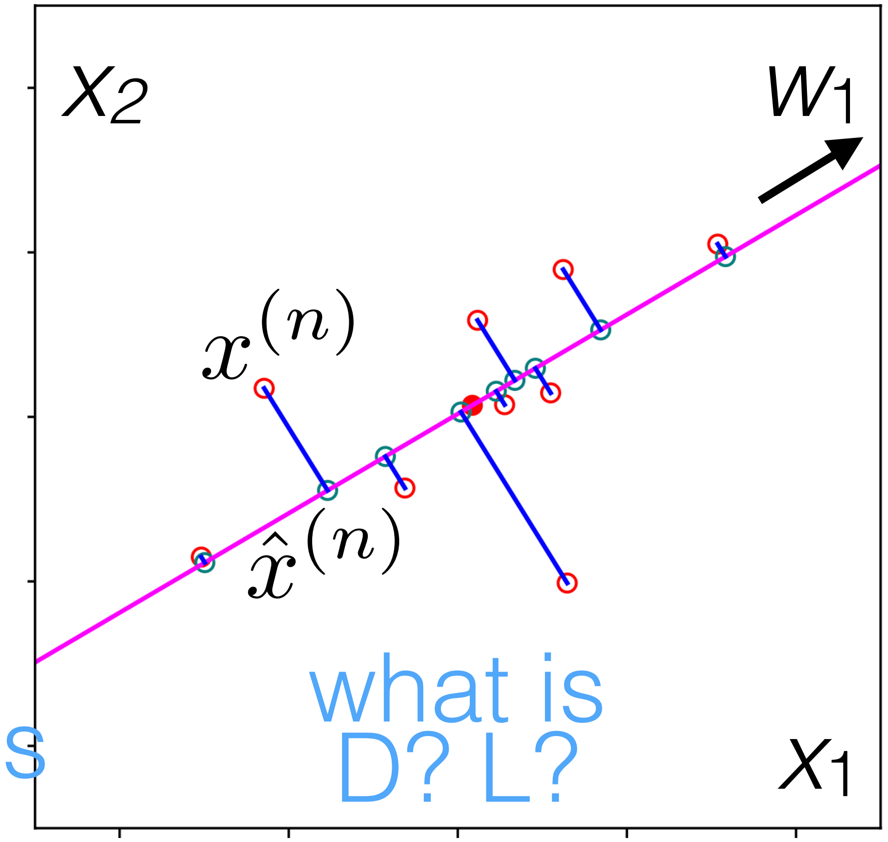
- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

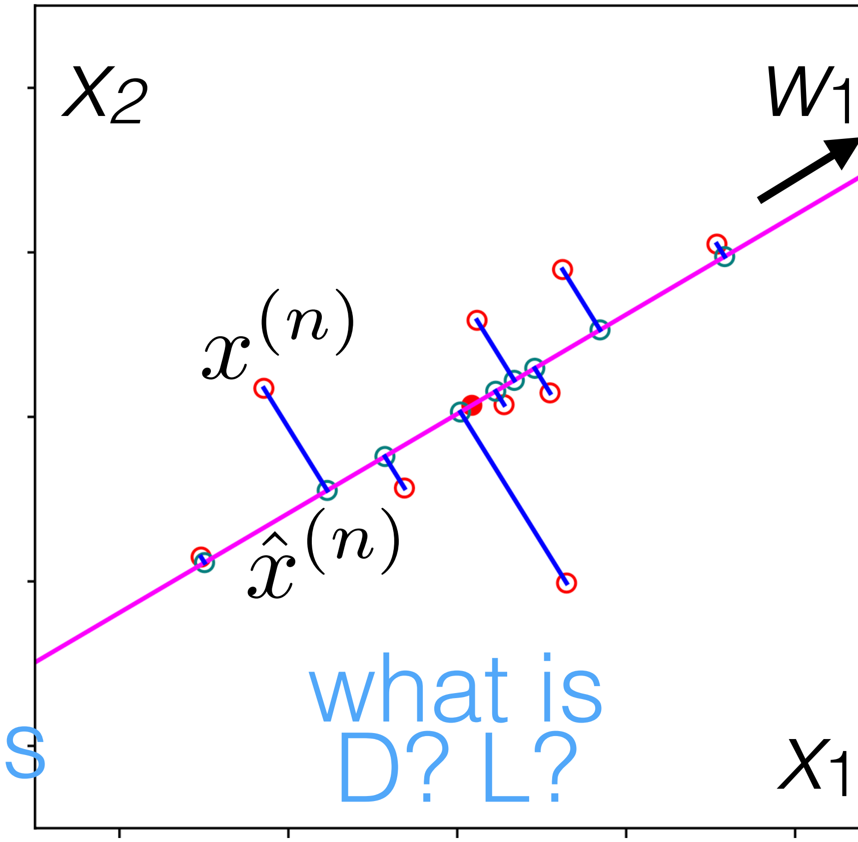
- Goal: projection is "close" to original data (square loss)

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



[adapted from Murphy 2022, Fig 20.1]

Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

- As usual $x^{(n)}$ is a Dx1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

- Goal: projection is "close" to original data (square loss)

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

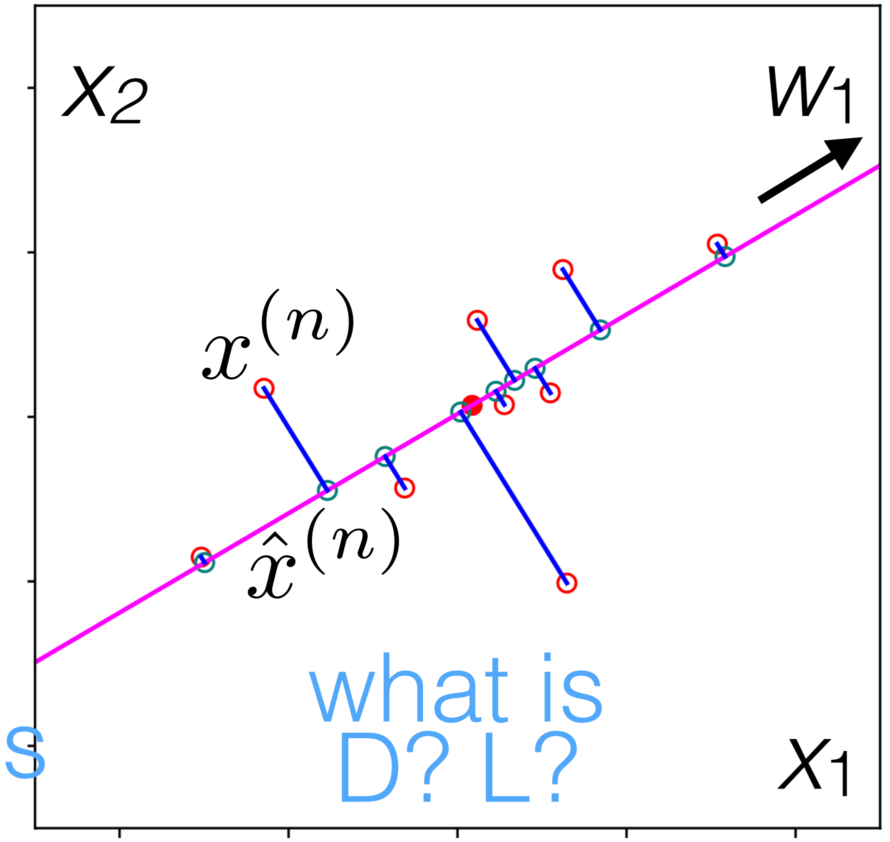
  - optimizing over W (DxL) and Z (NxL)

Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)



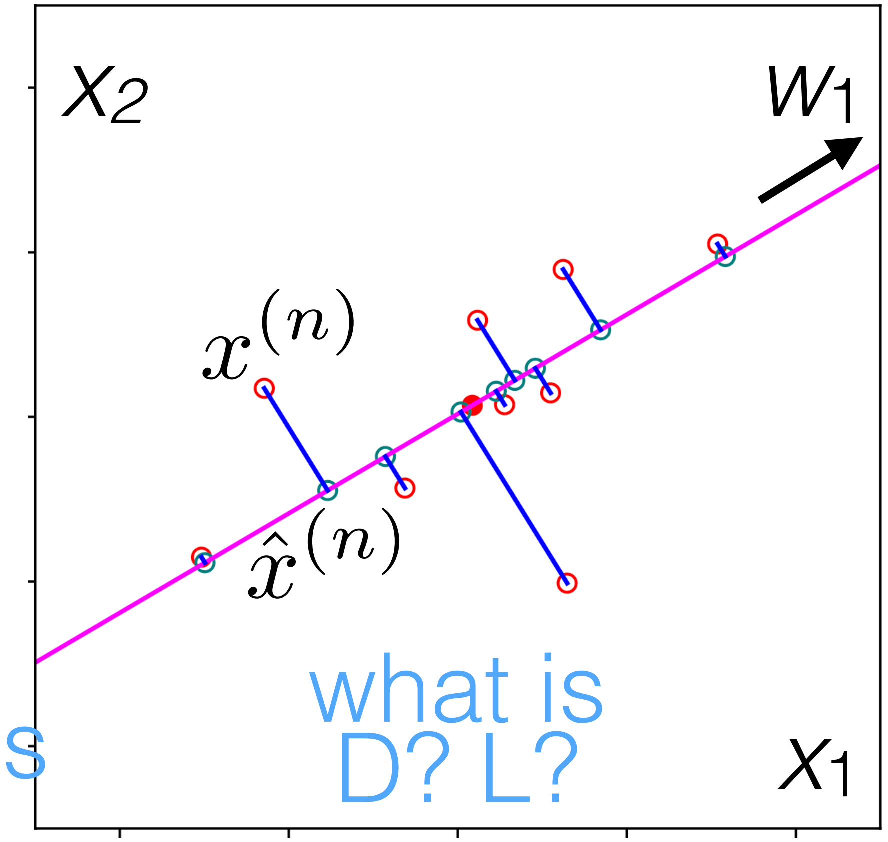
[adapted from Murphy 2022, Fig 20.1]

- Goal: projection is "close" to original data (square loss)

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

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

Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$

principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

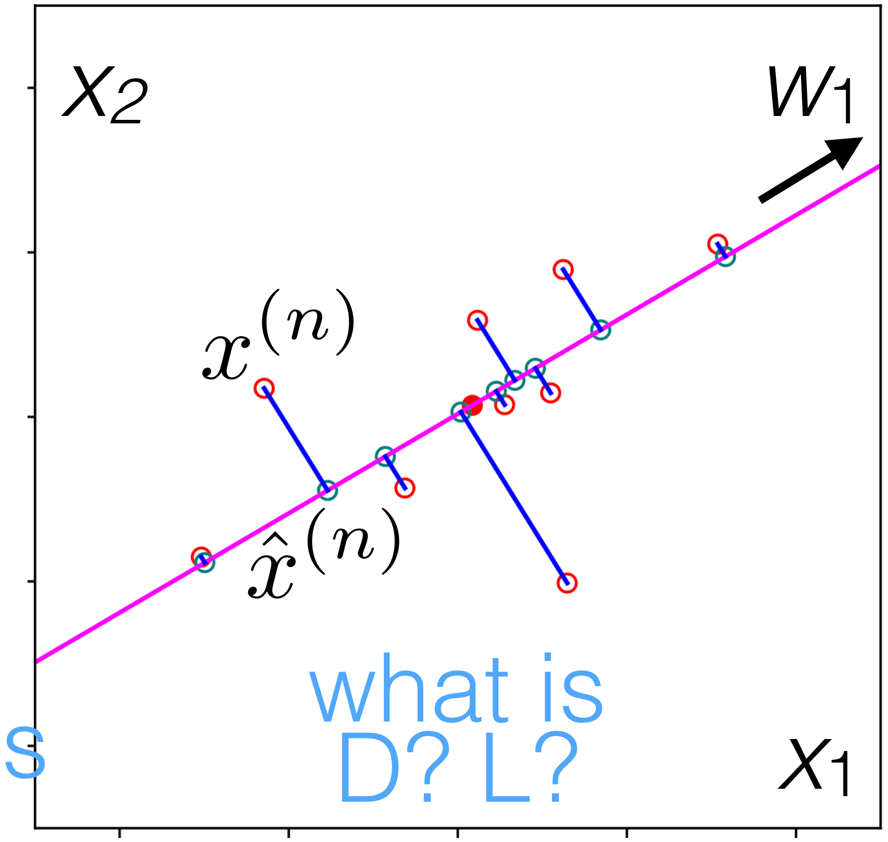
- Goal: projection is "close" to original data (square loss)

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

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

Problem setup for PCA

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ ← principal components



[adapted from Murphy 2022, Fig 20.1]

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

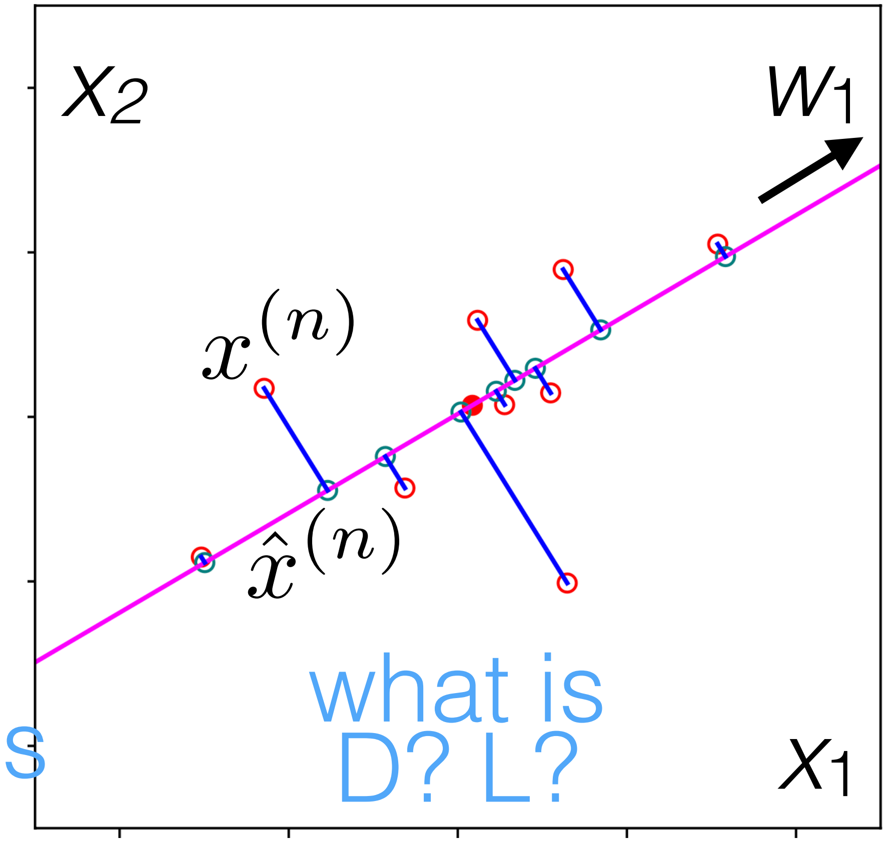
Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

- Goal: projection is "close" to original data (square loss)

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

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

Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

- As usual $x^{(n)}$ is a Dx1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ ← principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

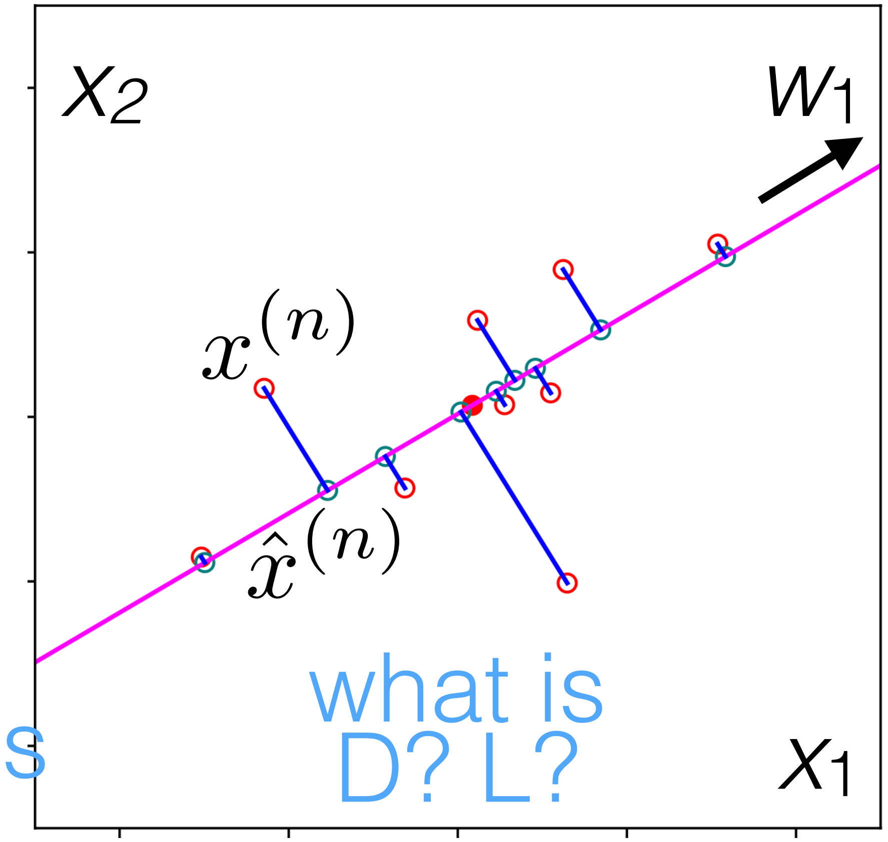
- Goal: projection is "close" to original data (square loss)

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

DxN DxL LxN

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

Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ ← principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

- Goal: projection is "close" to original data (square loss)

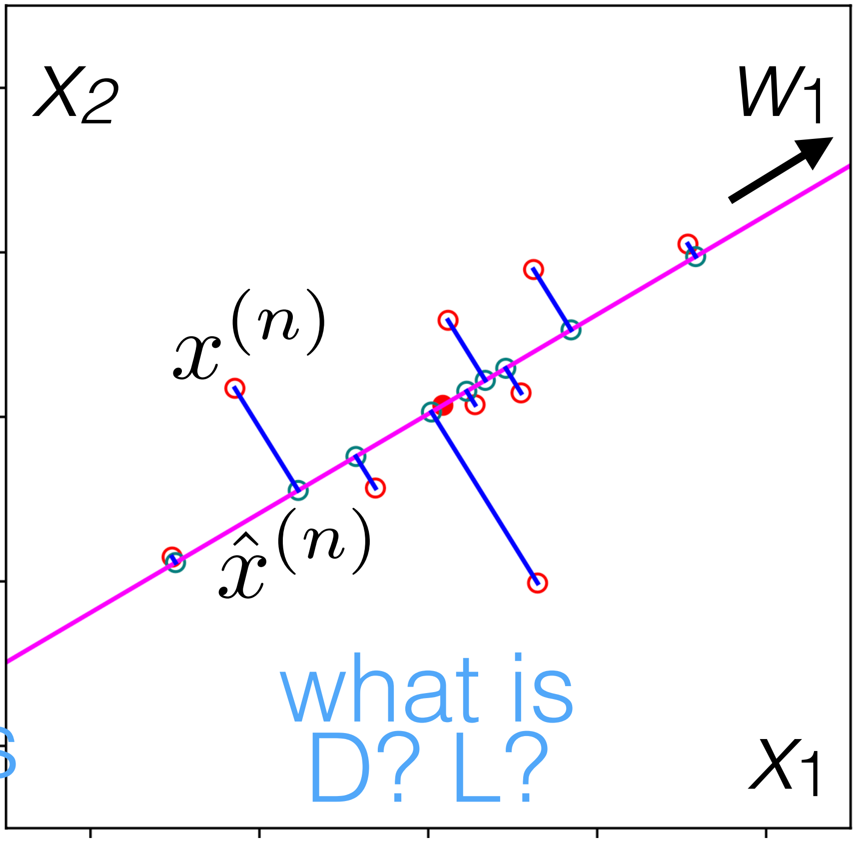
$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - W Z^\top\|_F^2$$

DxN DxL LxN

Frobenius norm (square is sum of square entries)

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

Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

- As usual $x^{(n)}$ is a Dx1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

- Goal: projection is "close" to original data (square loss)

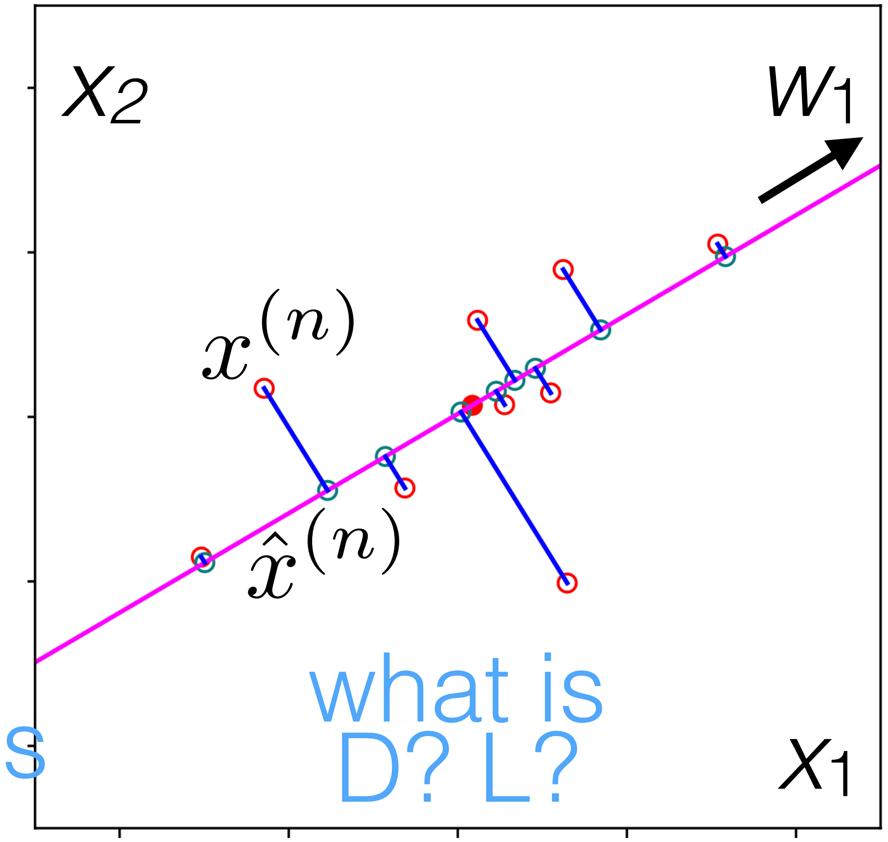
$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - W Z^\top\|_F^2 \boxed{= \|X - Z W^\top\|_F^2}$$

DxN DxL LxN

Frobenius norm (square is sum of square entries)

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

Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

- Goal: projection is "close" to original data (square loss)

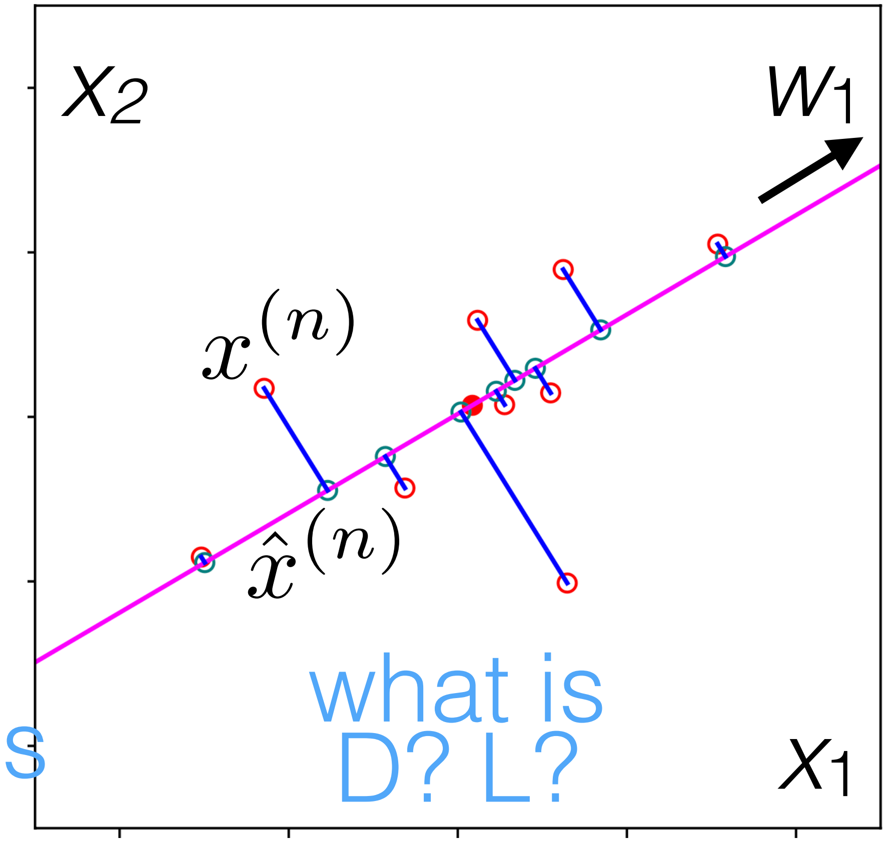
$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - W Z^\top\|_F^2 = \|X - Z W^\top\|_F^2$$

DxN DxL LxN NxD NxL LxD

Frobenius norm (square is sum of square entries)

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

Problem setup for PCA



[adapted from Murphy 2022, Fig 20.1]

- As usual $x^{(n)}$ is a *D*x1 vector
- Pre-process so $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
- Assume: we'd like to approximate the data with its projection onto a low-dimensional subspace, with orthonormal basis $w_1, \ldots, w_L$ principal components

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

Dx1 weights:1x1 Dx1 DxL Lx1 no offset (just rotating basis)

- Goal: projection is "close" to original data (square loss)

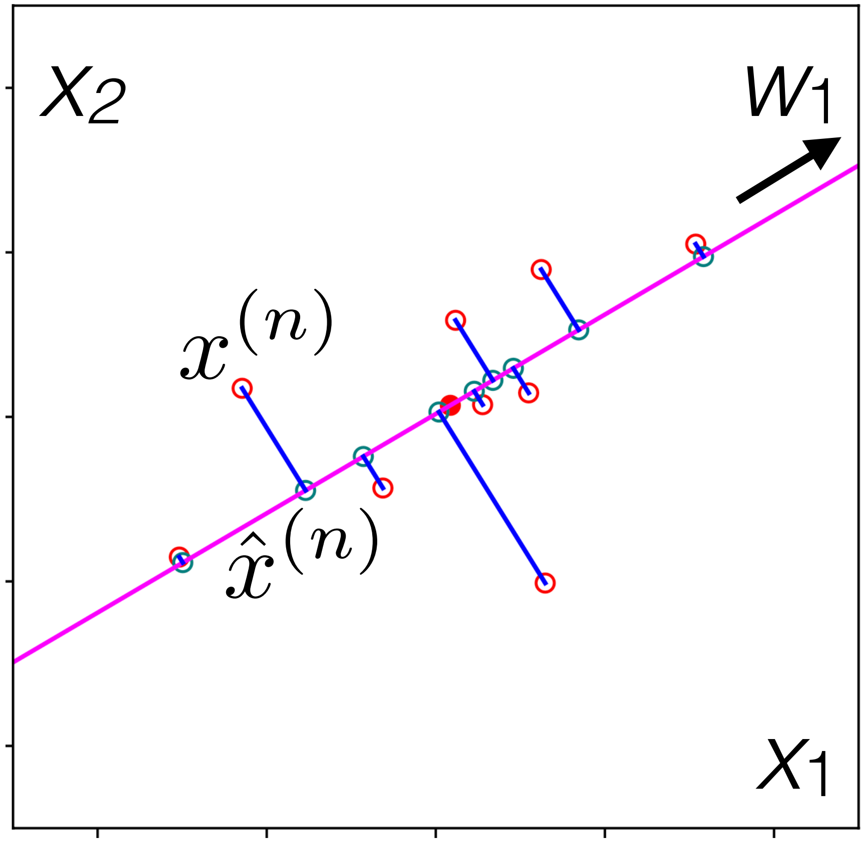
$$\min \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2 = \|X^\top - W Z^\top\|_F^2 = \|X - Z W^\top\|_F^2$$

DxN DxL LxN NxD NxL LxD

Frobenius norm (square is sum of square entries)

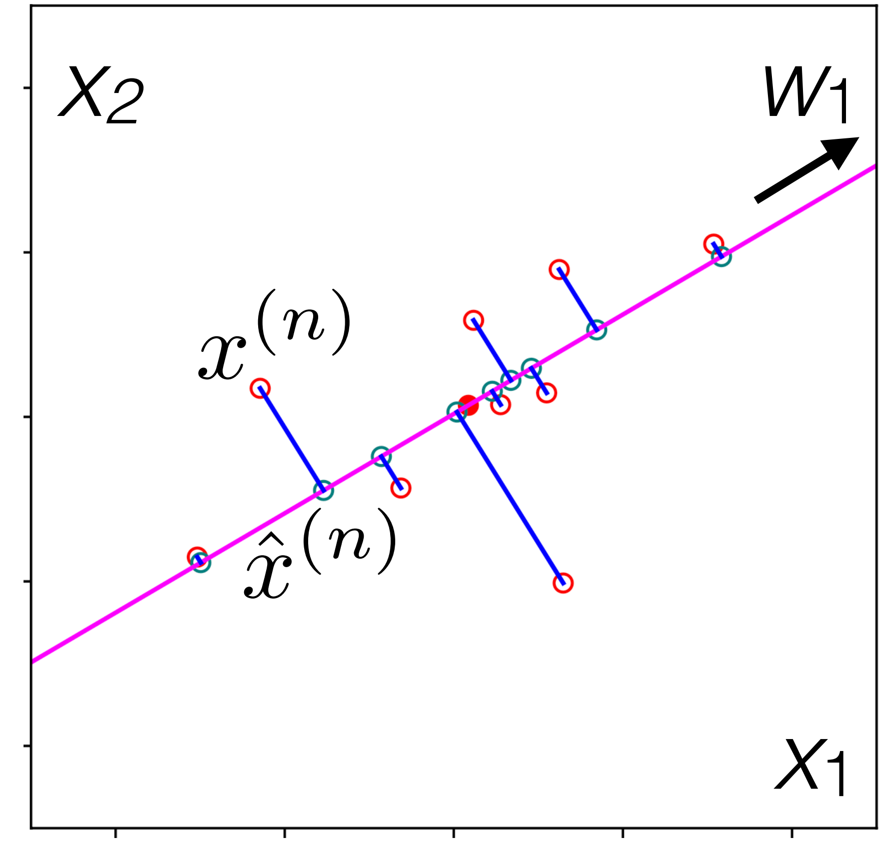
  - optimizing over *W* (DxL) and *Z* (NxL)
  - constraint: *W* represents an orthonormal basis
  - Observe: if we find a best basis, could instead use -1 times any basis vector, -1 times the corresponding *z* values, and get the same result

Solving when $L=1$



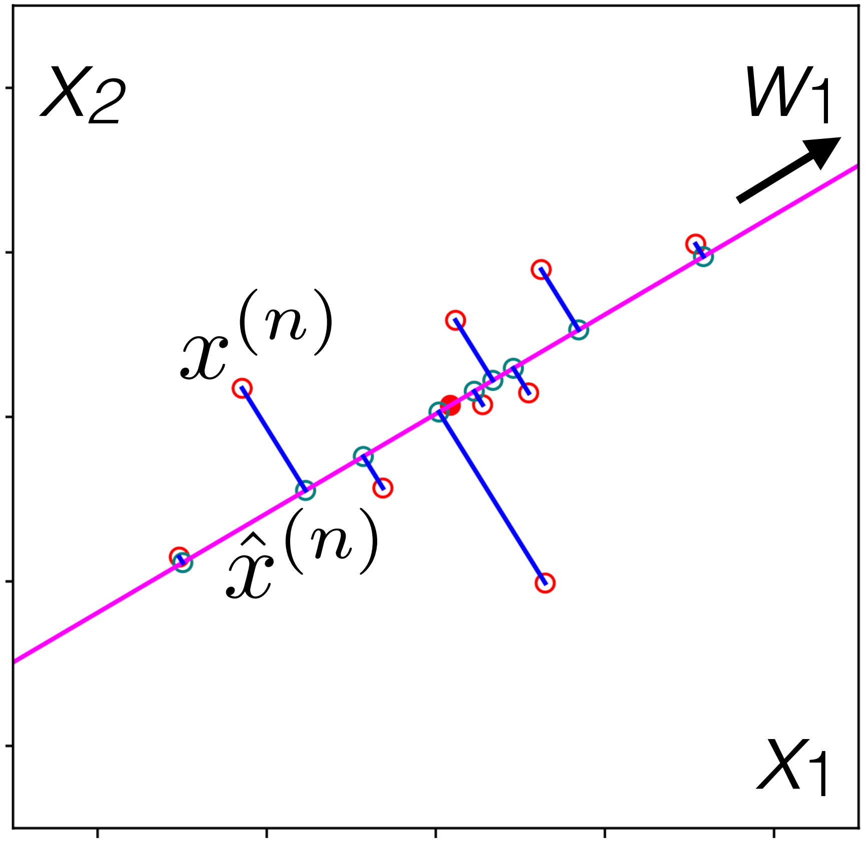
Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$



Solving when L=1

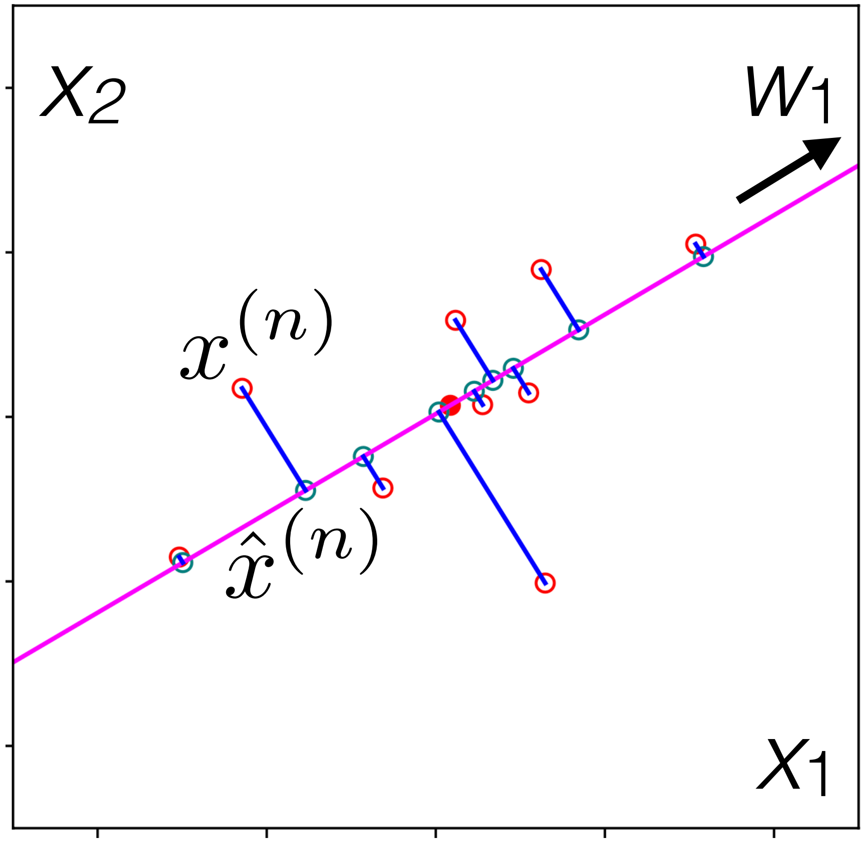
- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)



Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

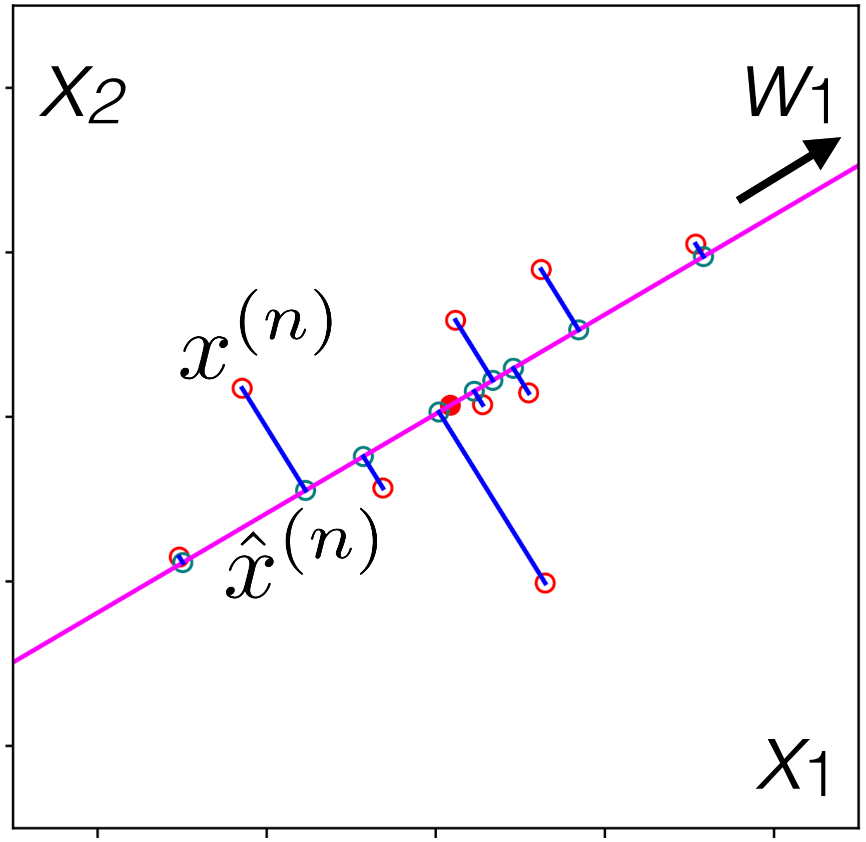
$$\min \sum_{n=1}^{N} \|x^{(n)} - z_1^{(n)} w_1\|^2$$



Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

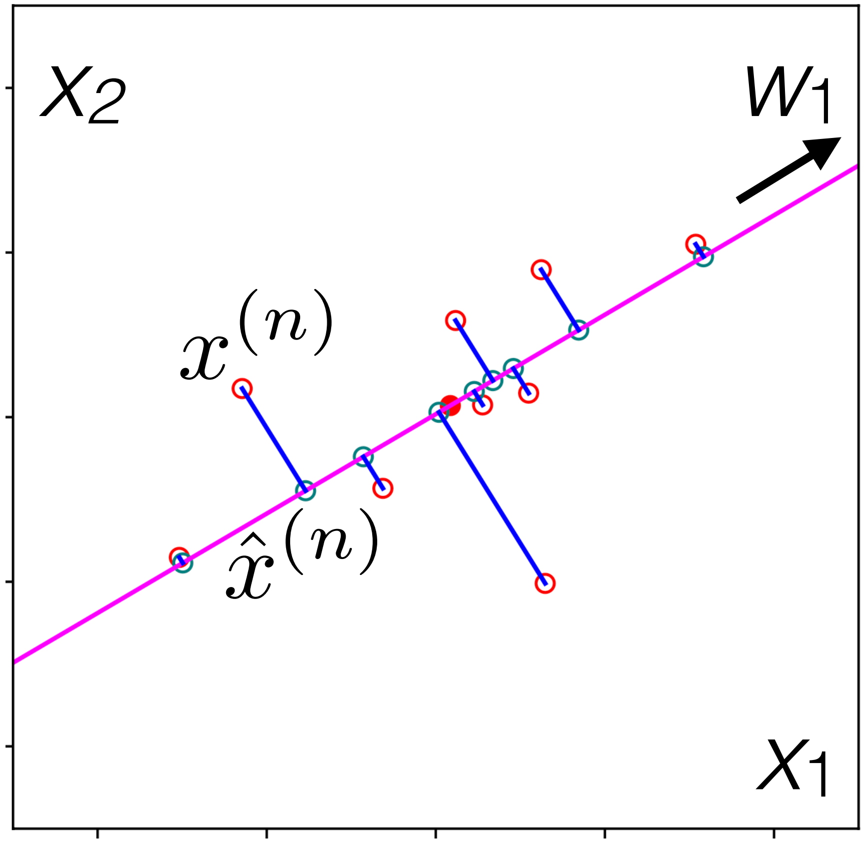
$$\min \sum_{n=1}^{N} \|x^{(n)} - z_1^{(n)} w_1\|^2$$



Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

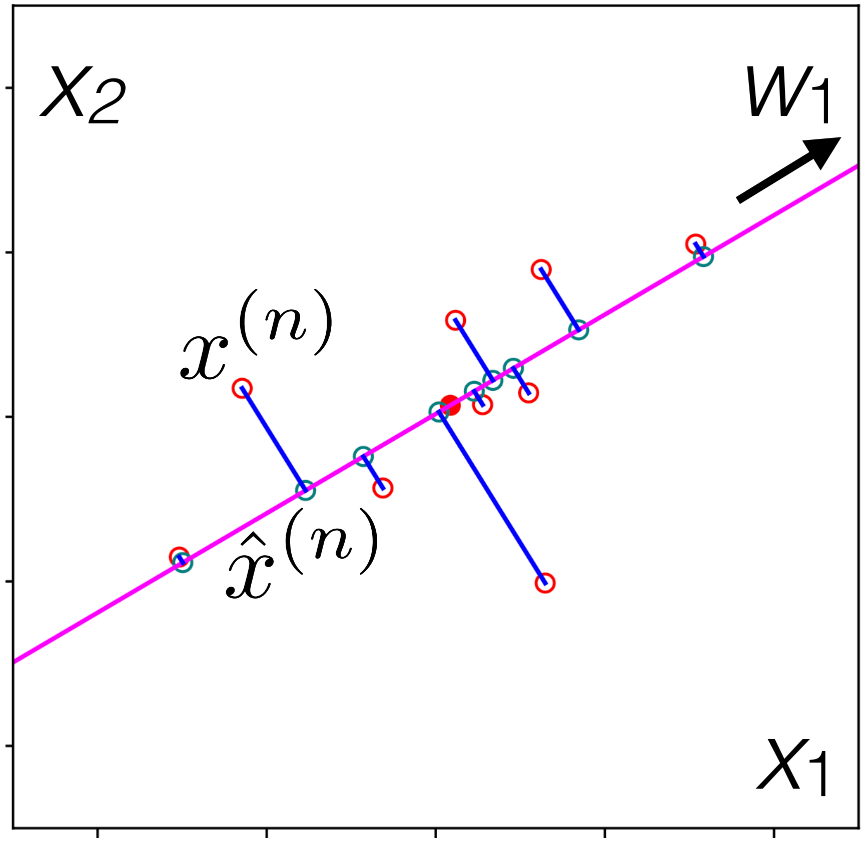
$$\min \sum_{n=1}^{N} \|x^{(n)} - z_1^{(n)} w_1\|^2$$



Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

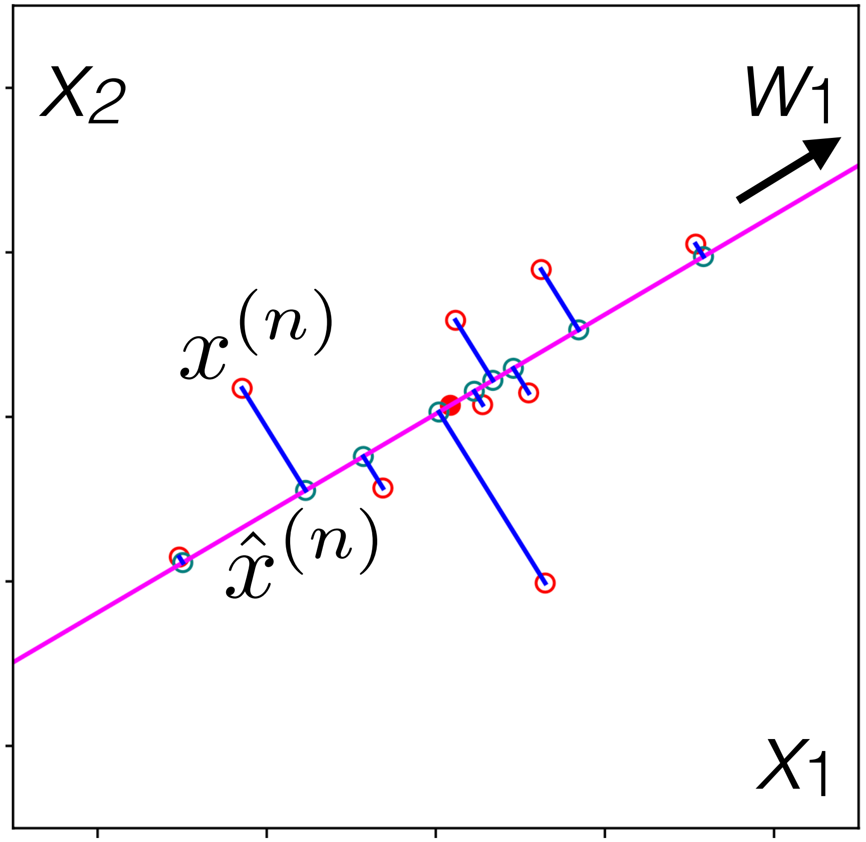


Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^N x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^N \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$(x^{(n)})^\top x^{(n)} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 w_1^\top w_1$$

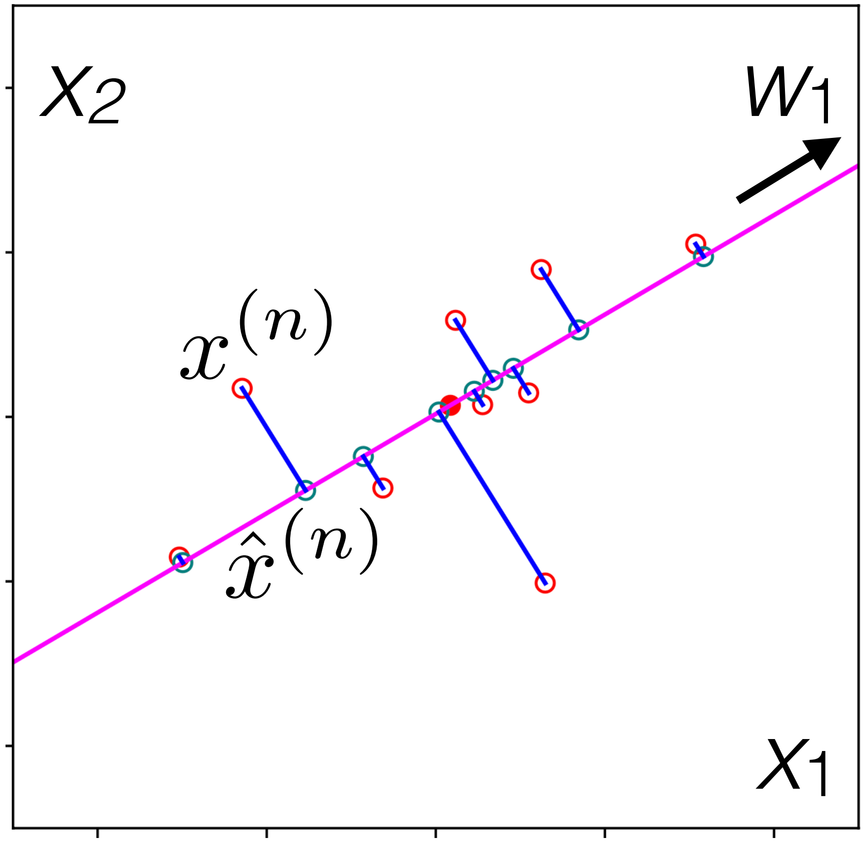


Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

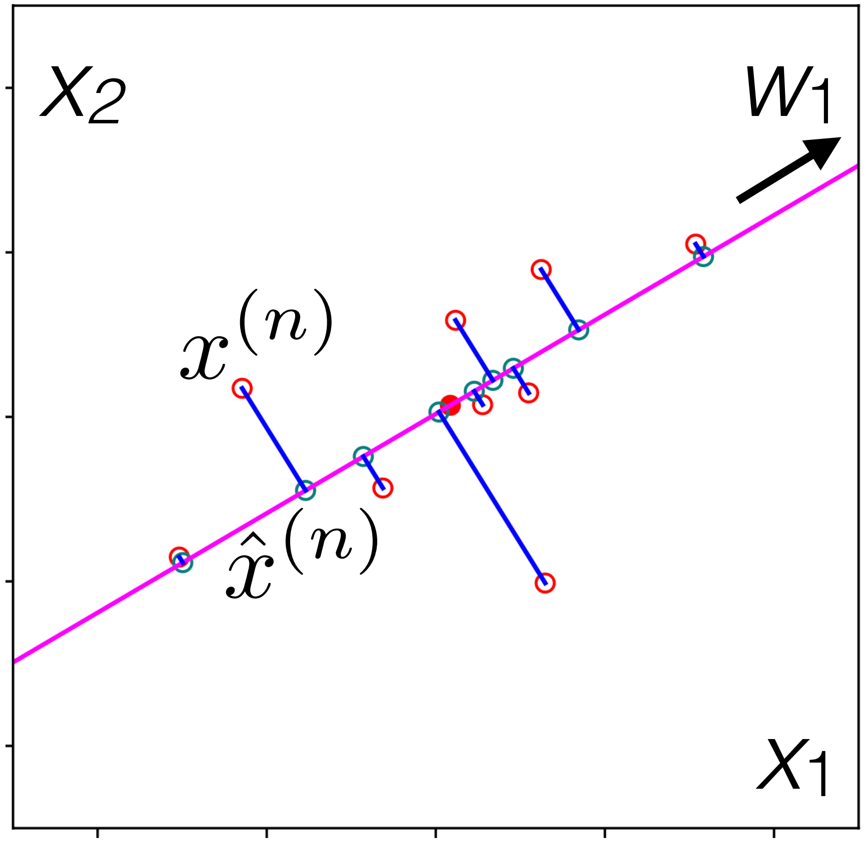
$$\underbrace{(x^{(n)})^\top x^{(n)}} - 2z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 w_1^\top w_1$$



Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^N x^{(n)} = 0_D$
    - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^N \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 w_1^\top w_1}$$

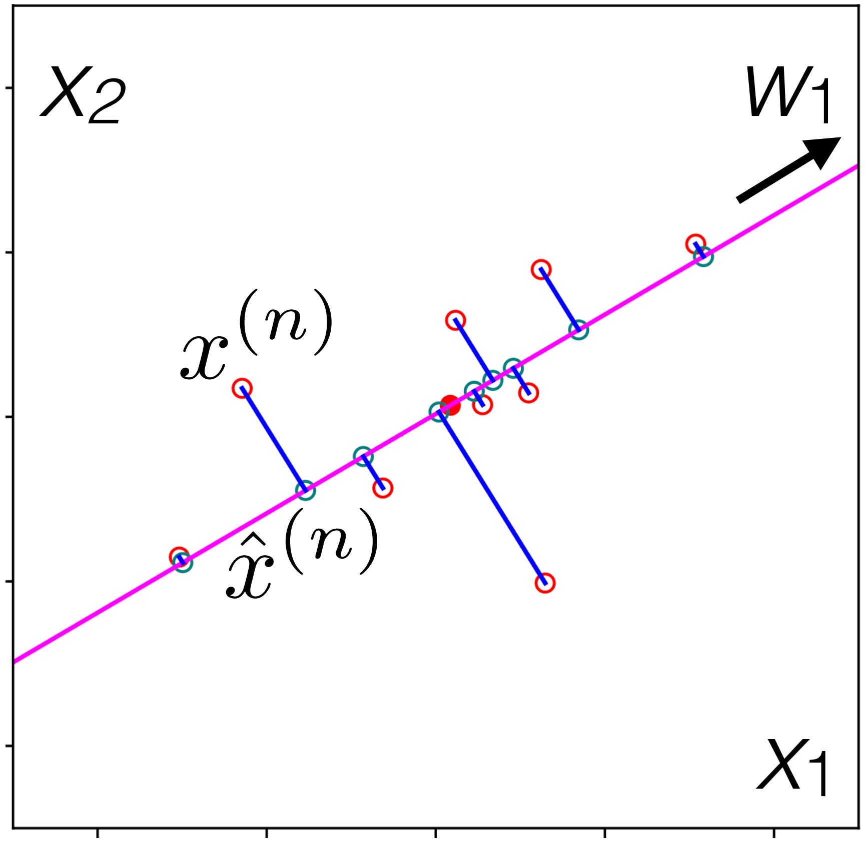


Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{}$$

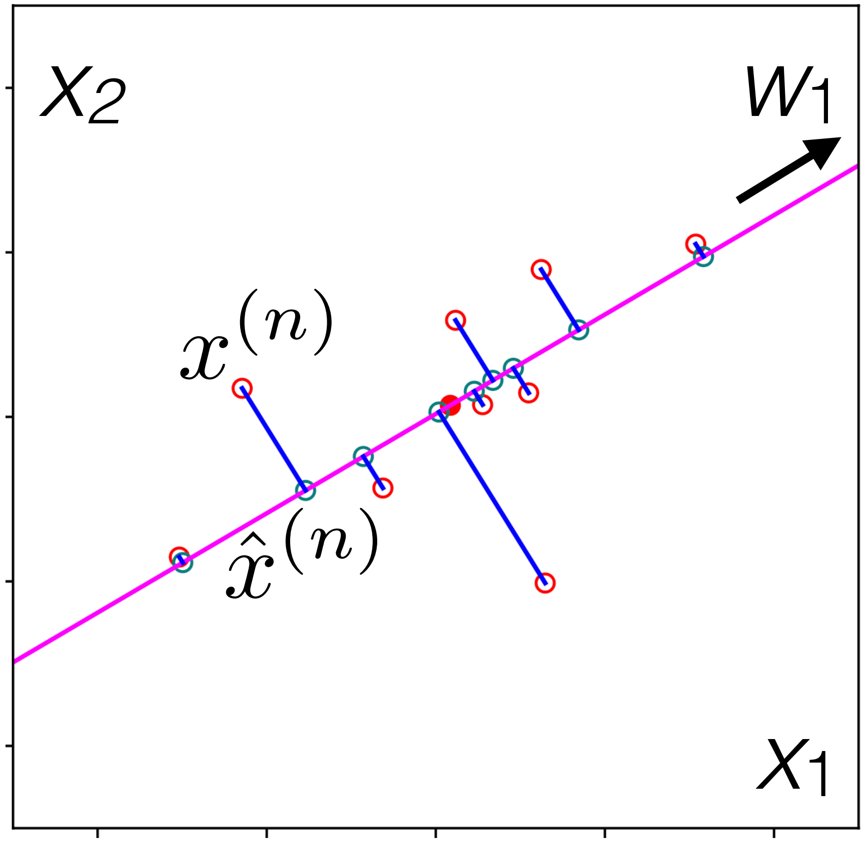


Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

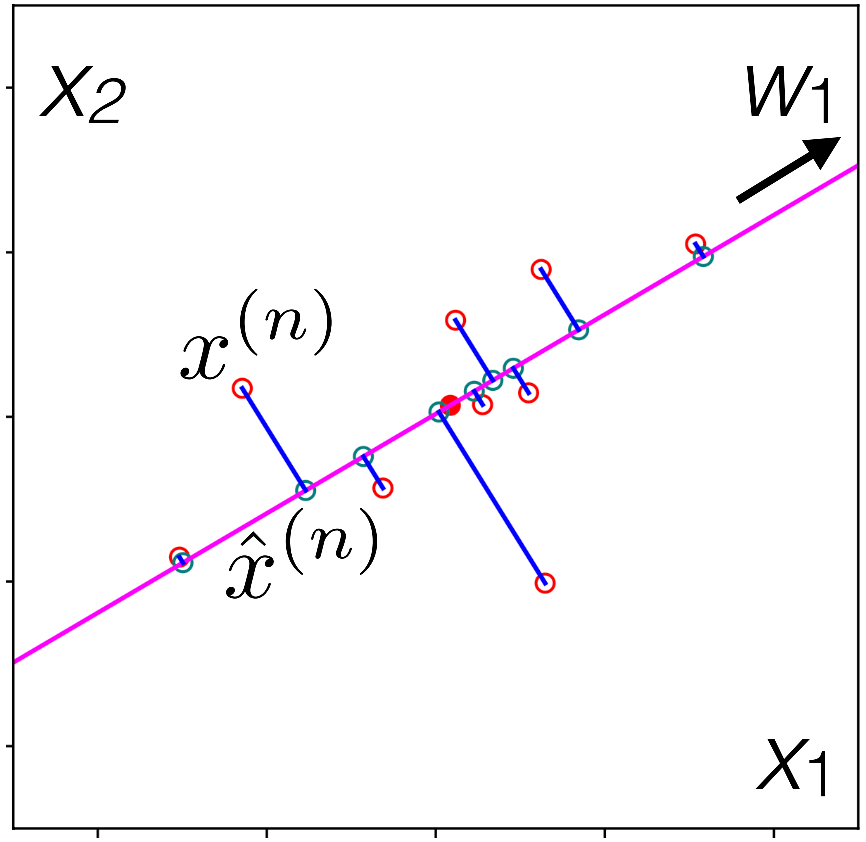
$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)



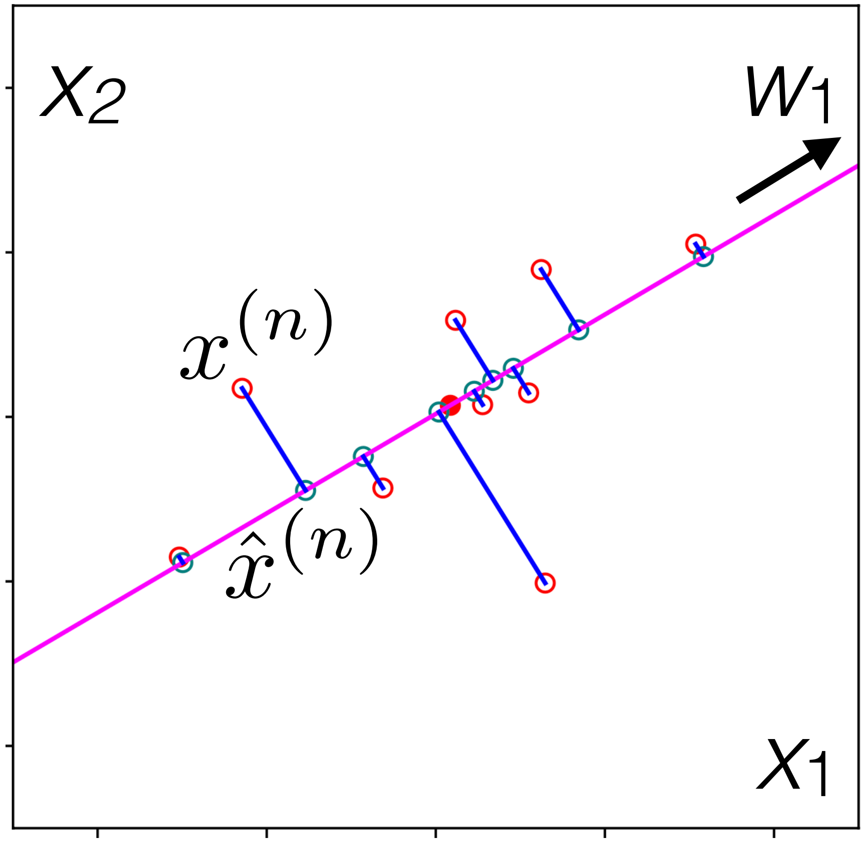
$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

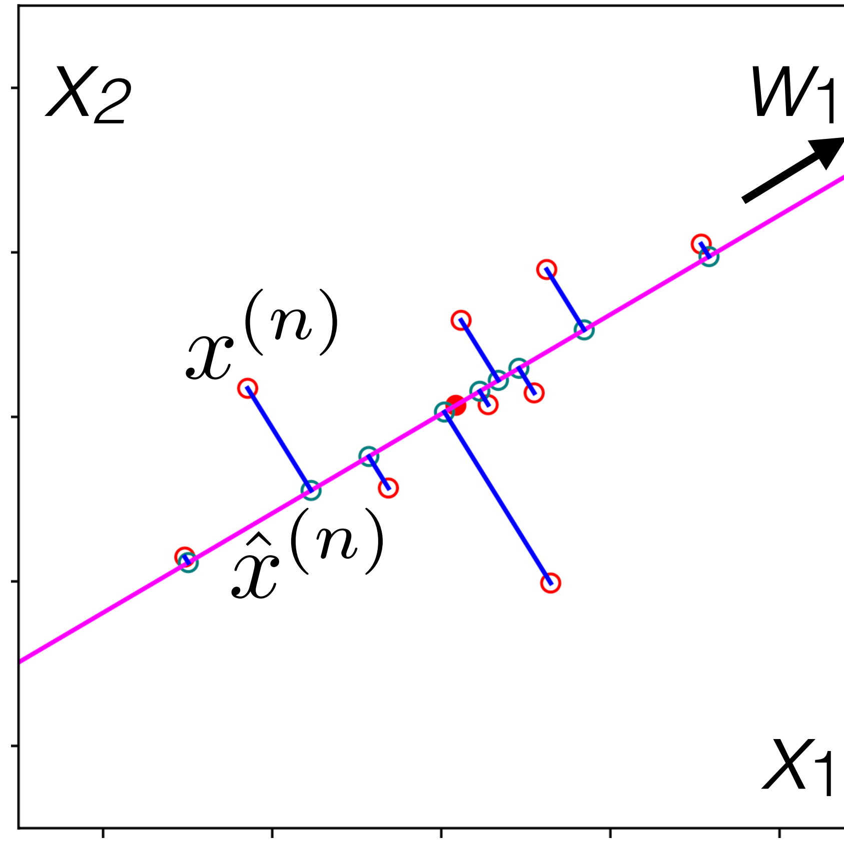
$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0:

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0:

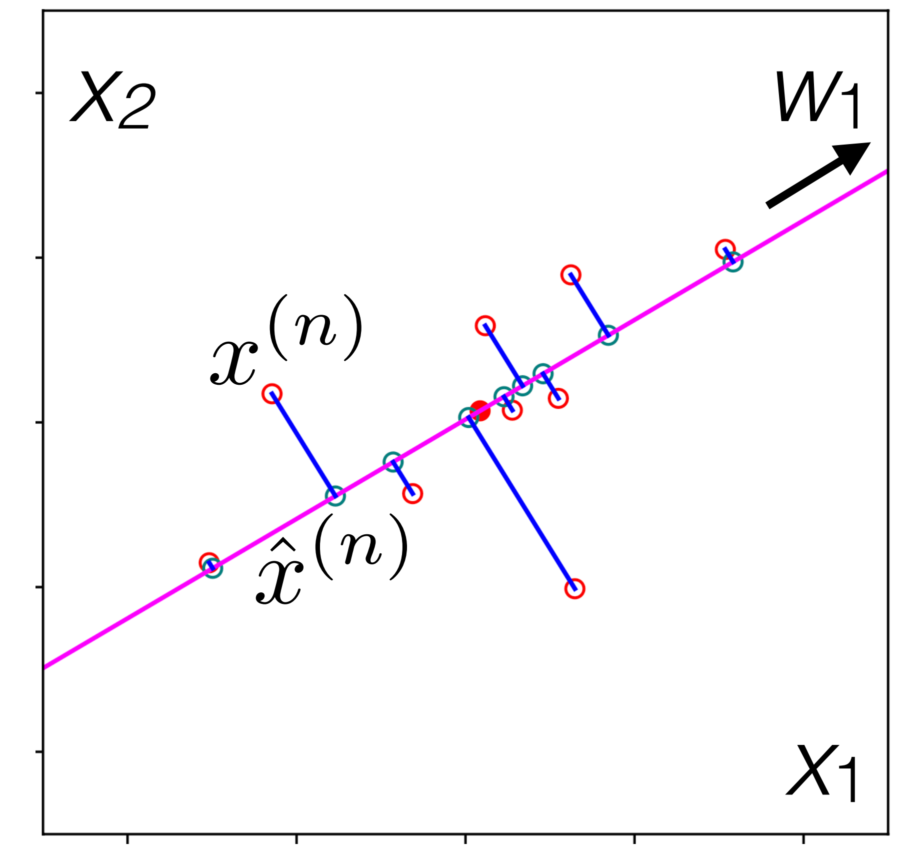
second order condition: check

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0:

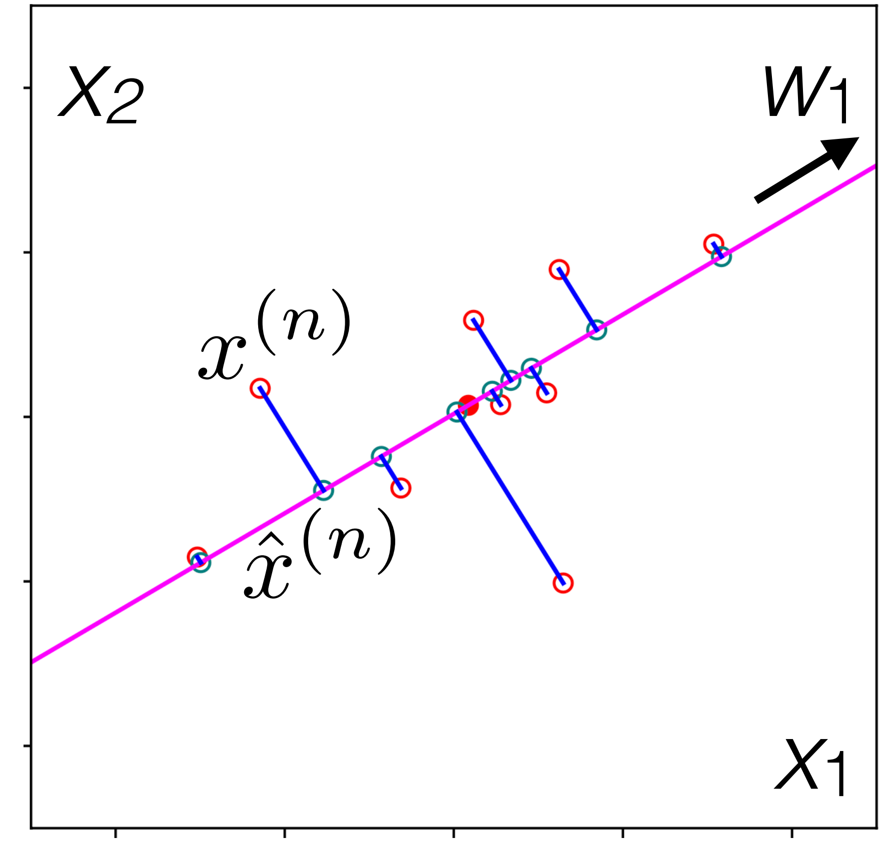
second order condition: check

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

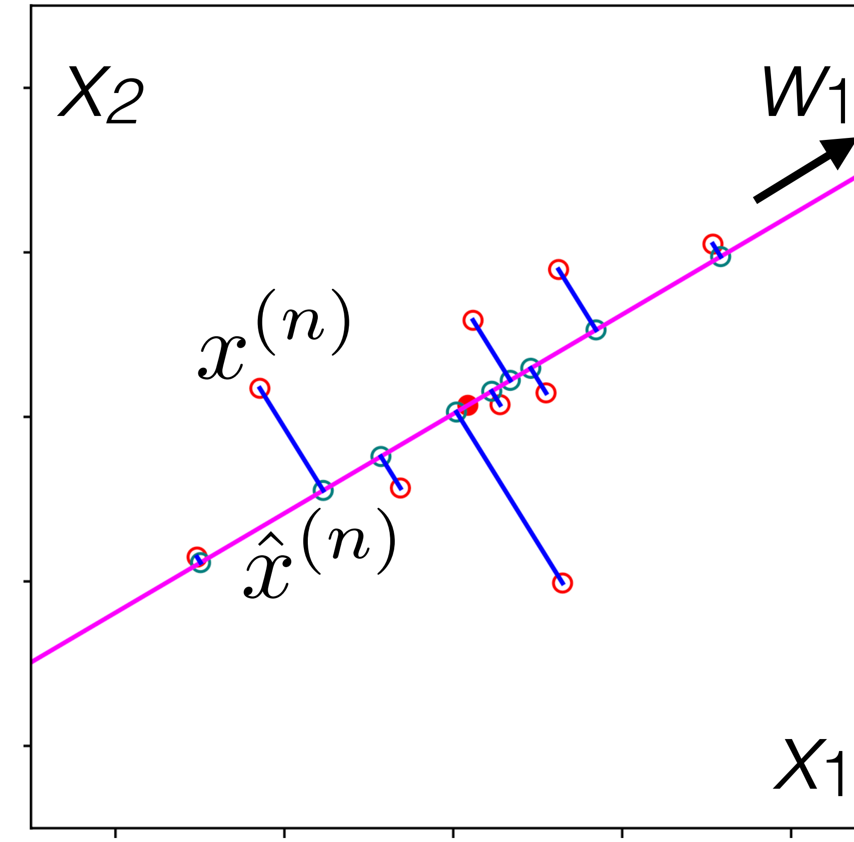
second order condition: check

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

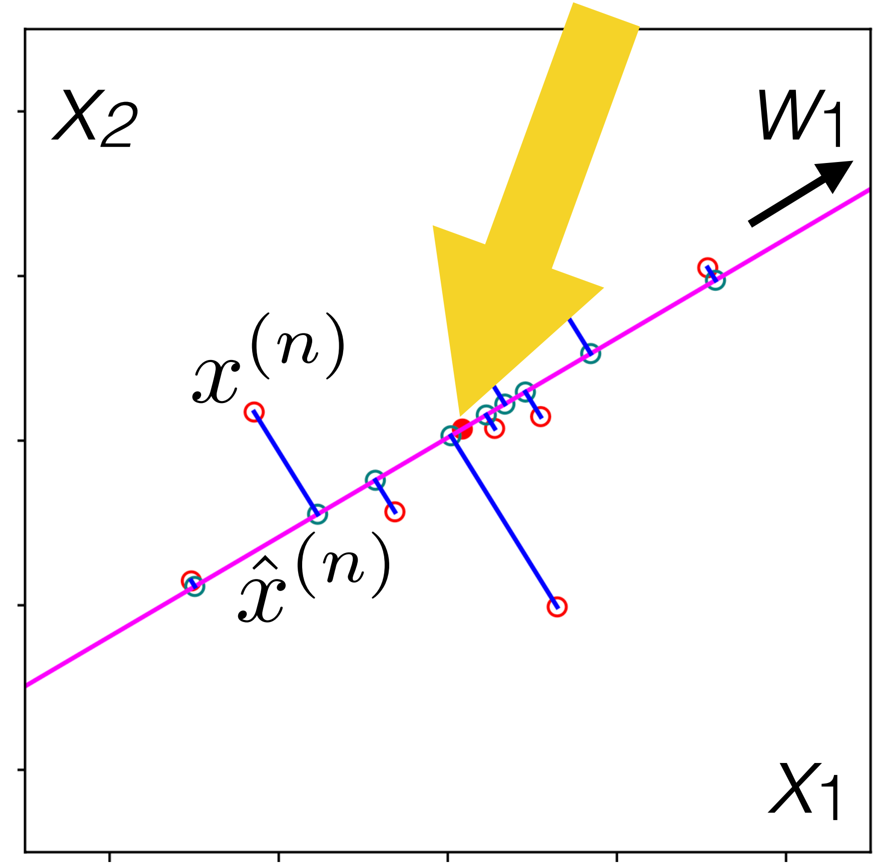
scalar projection of the data in the w_1 direction

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

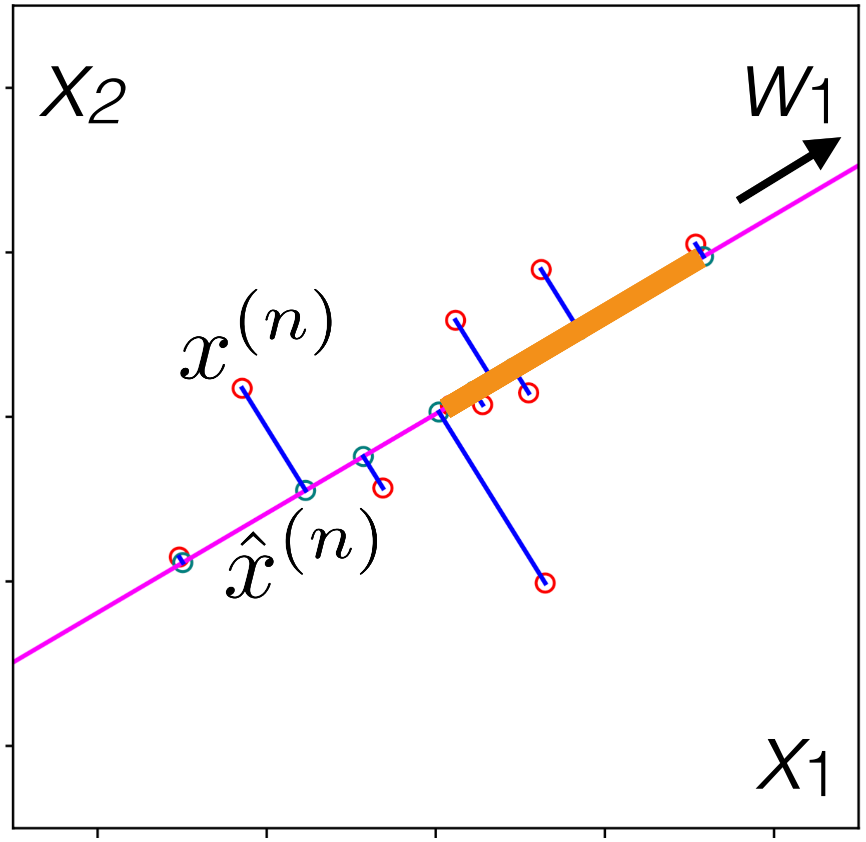
second order condition: check

scalar projection of the data in the w_1 direction

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

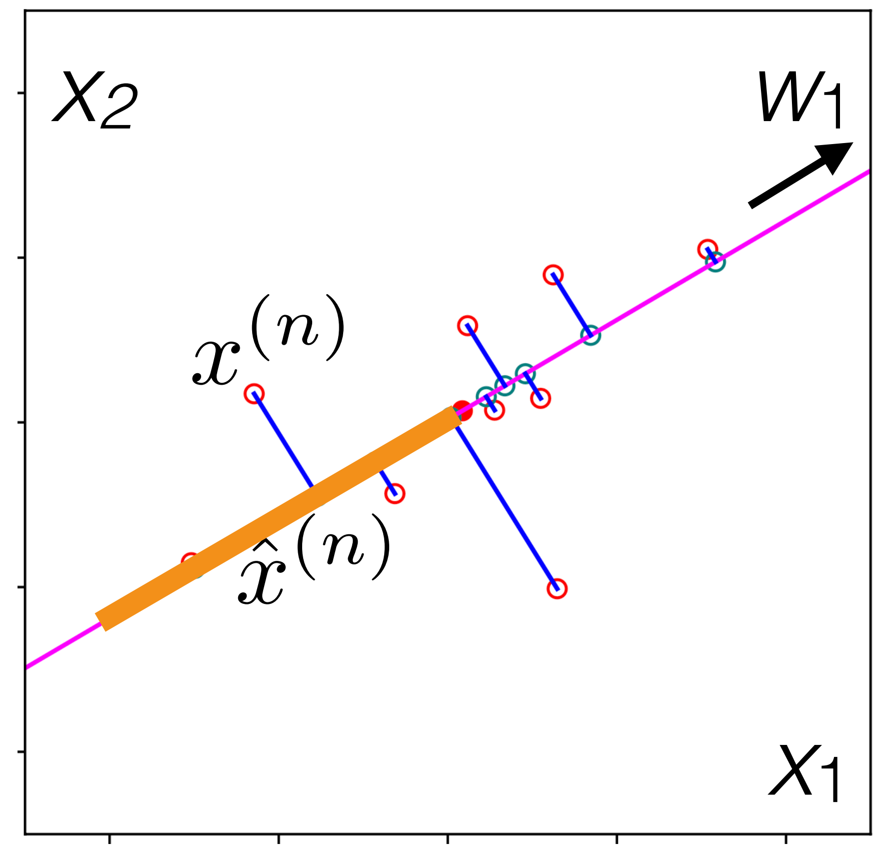
scalar projection of the data in the w_1 direction

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

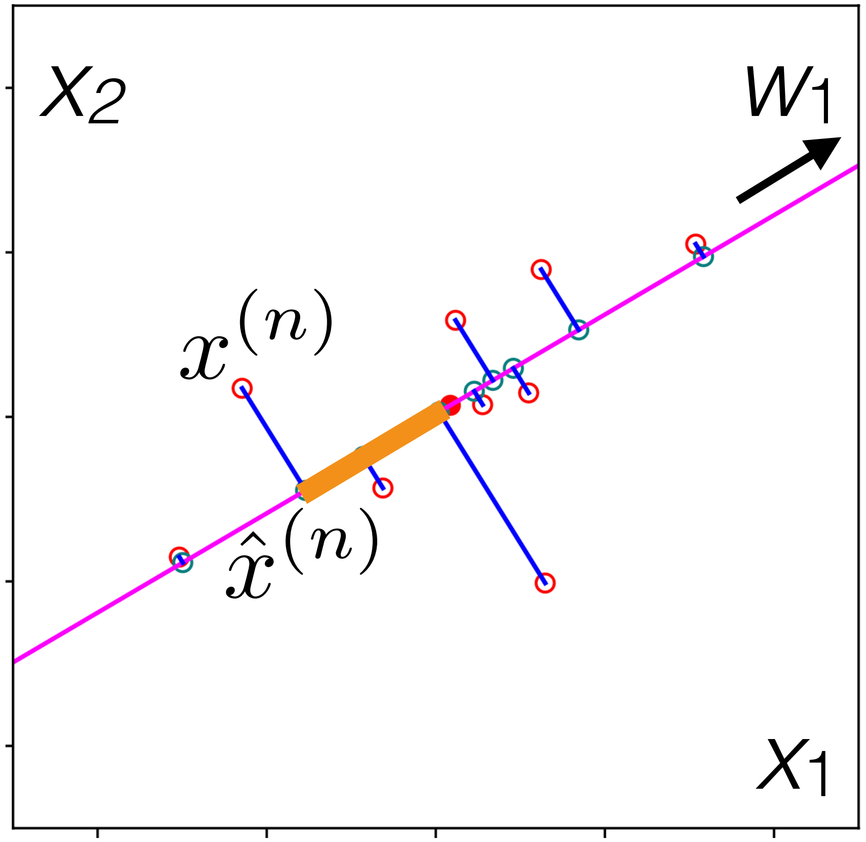
scalar projection of the data in the w_1 direction

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

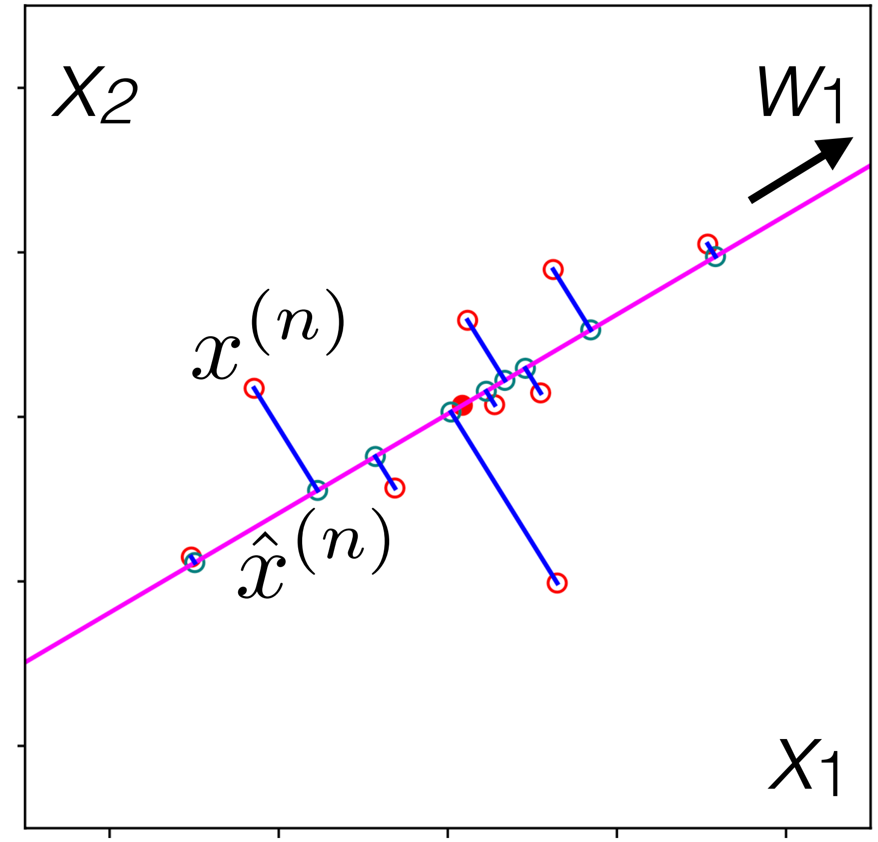
scalar projection of the data in the w_1 direction

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

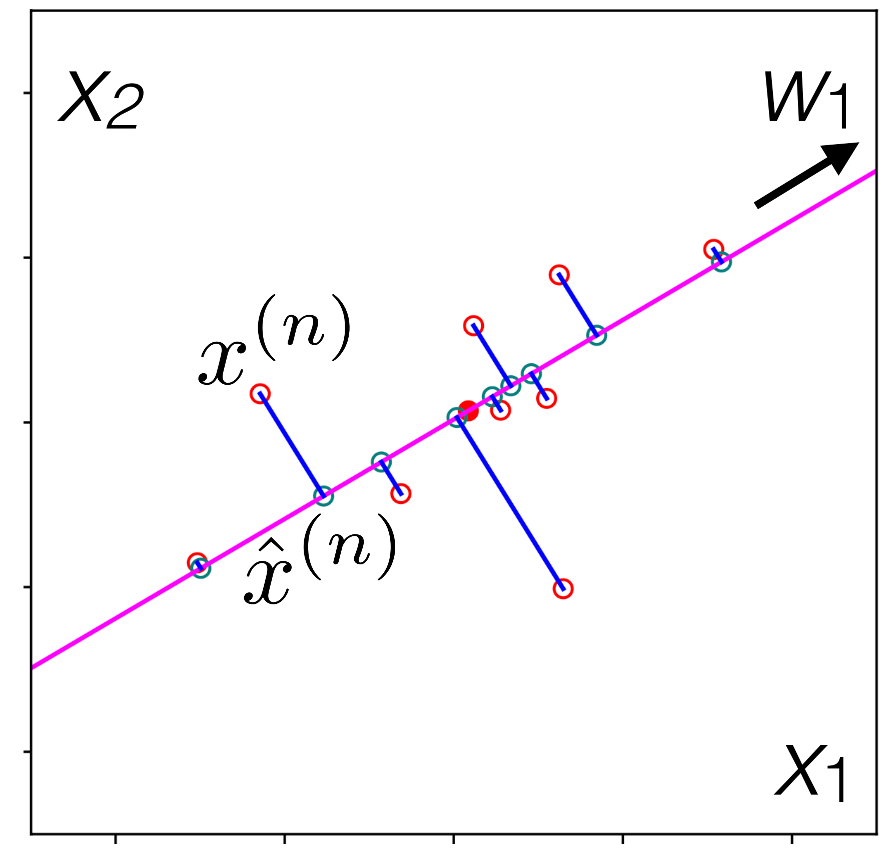
- Plug back in to the objective

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$



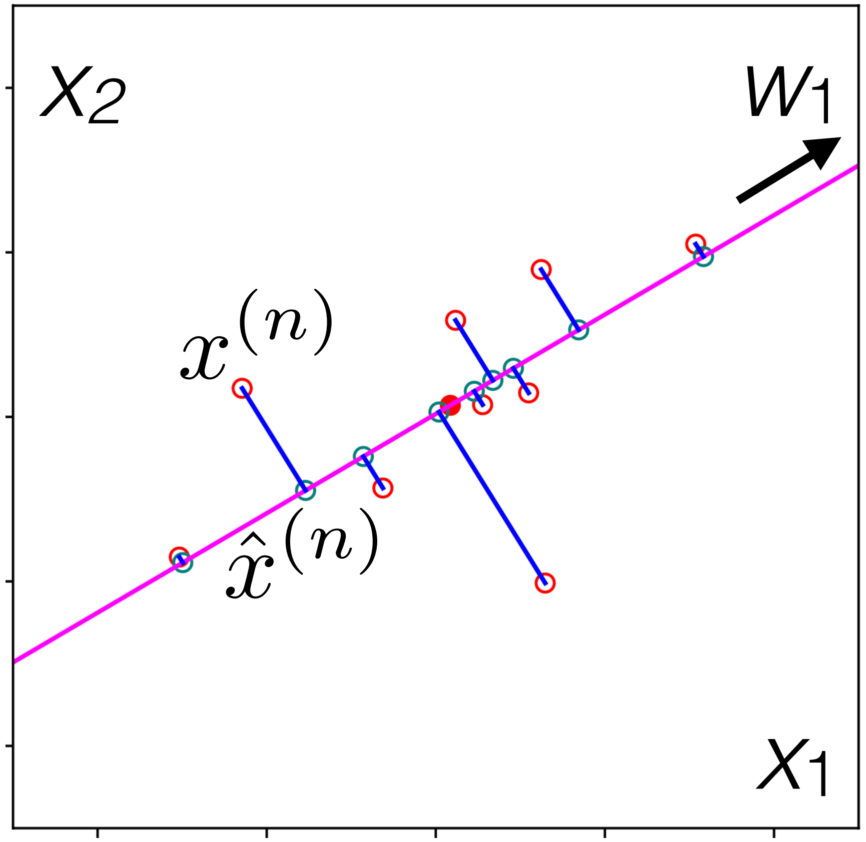
- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

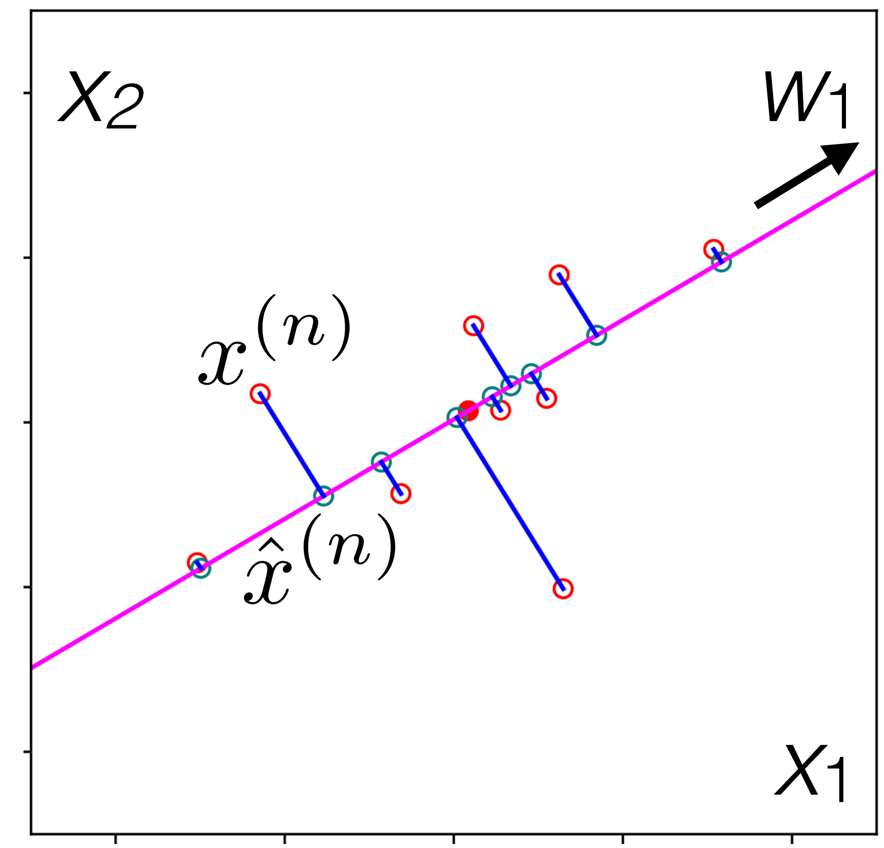
- Plug back in to the objective; we want to minimize

$$-\sum_{n=1}^{N}(w_1^\top x^{(n)})^2$$

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

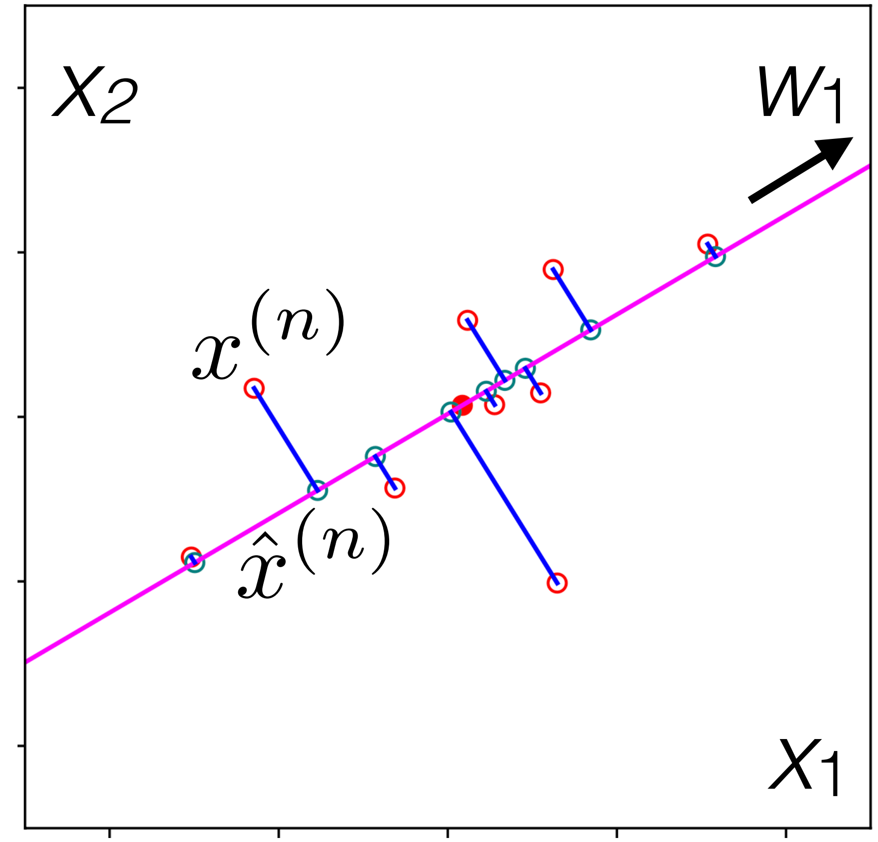
scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top\right] w_1$$

corrected from live lecture!

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

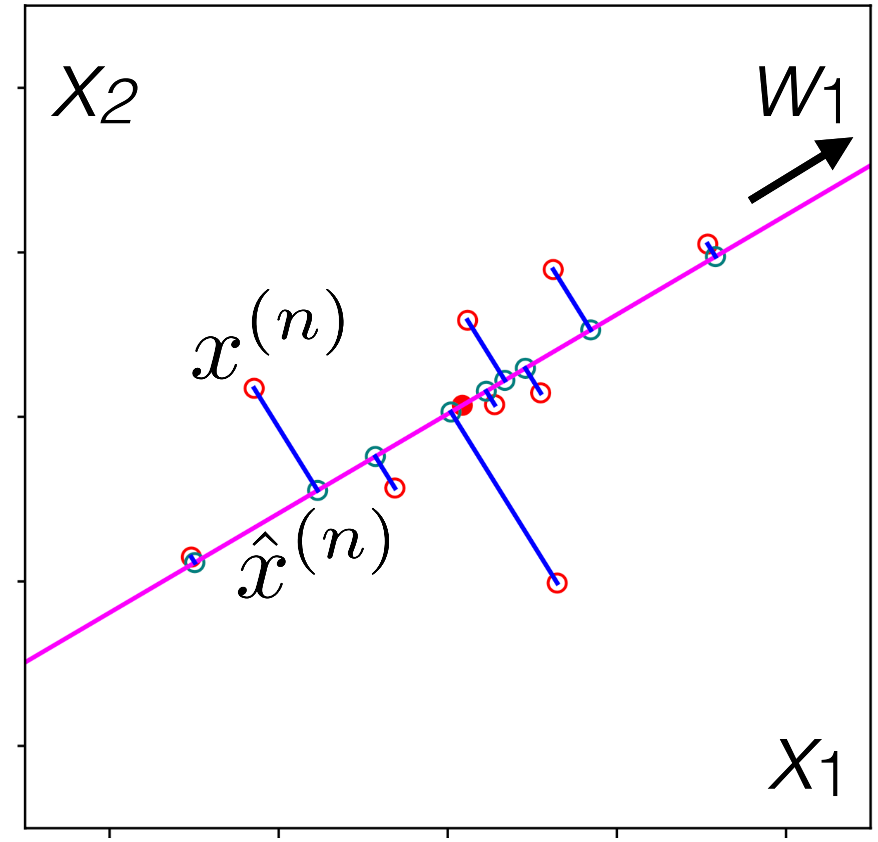
$$-\sum_{n=1}^{N}(w_1^\top x^{(n)})^2 = -w_1^\top \left[\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top\right] w_1$$

corrected from live lecture!

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

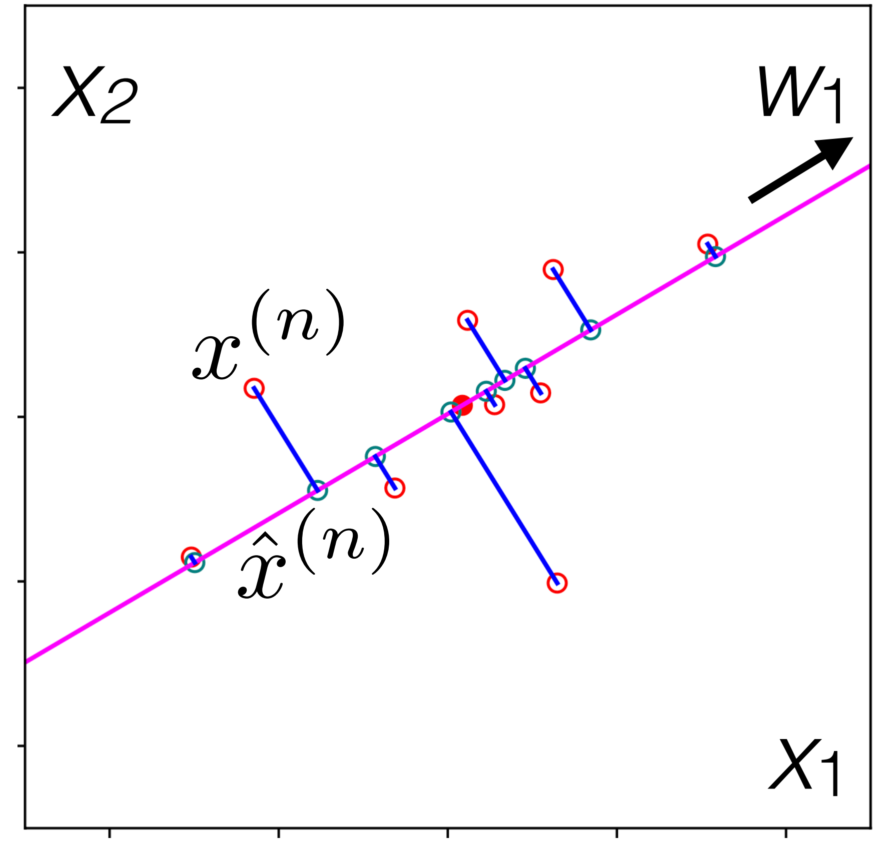
scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}}\right] w_1$$

corrected from live lecture!

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
    - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

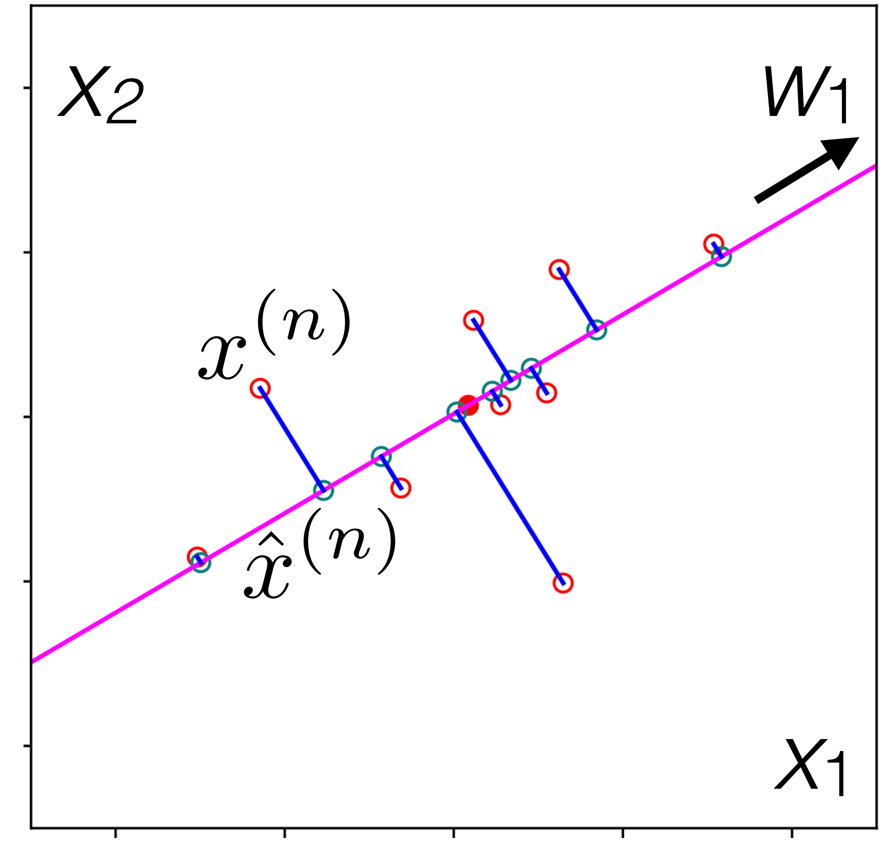
- Plug back in to the objective; we want to minimize

$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}}\right] w_1$$

corrected from live lecture!

to include constraint:

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

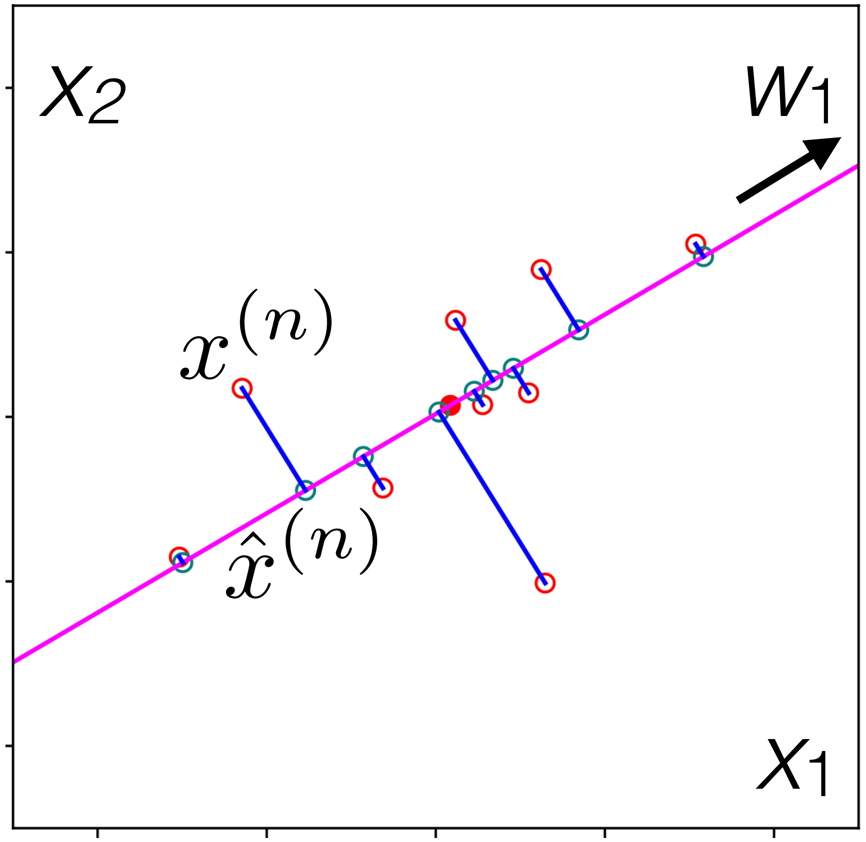
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}} \right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

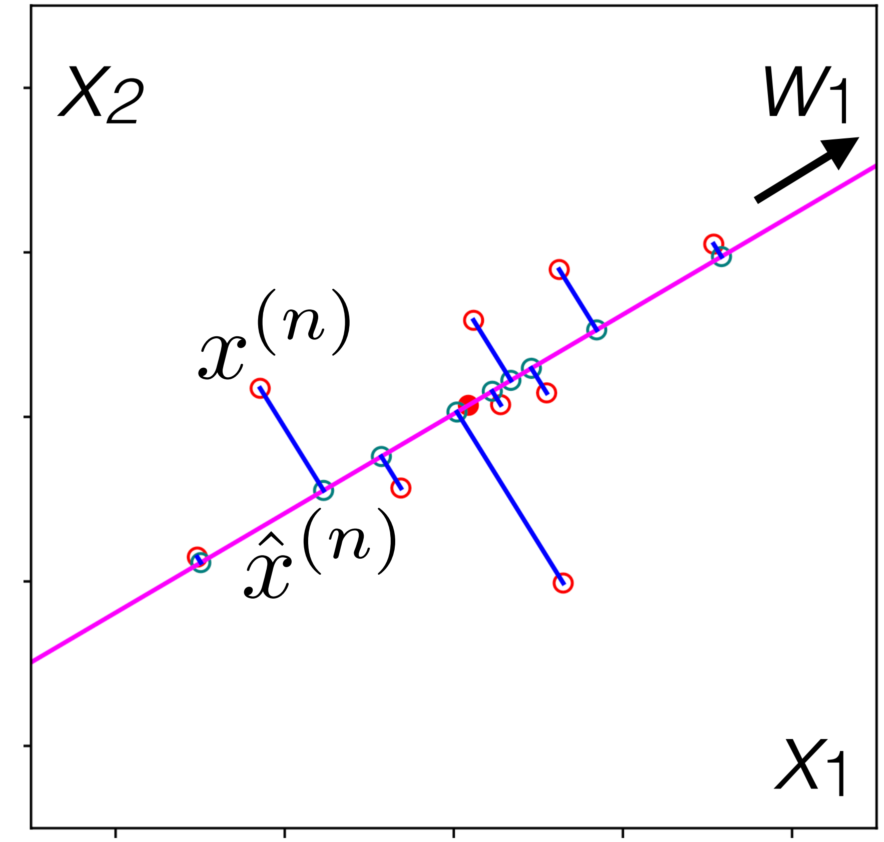
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}} \right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

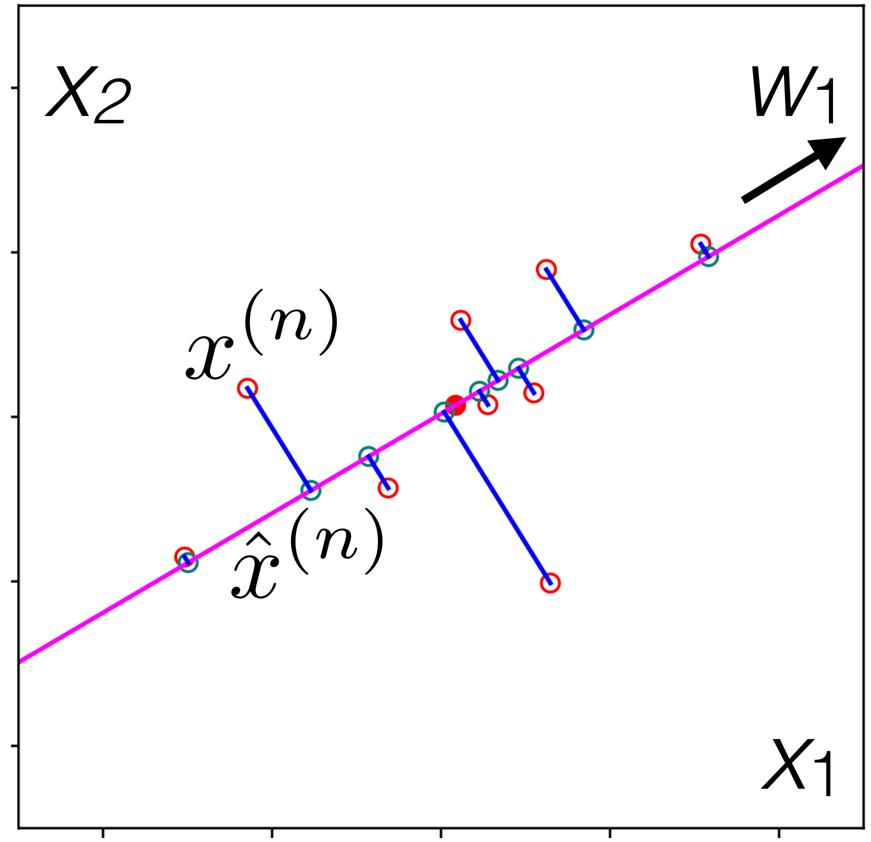
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}} \right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

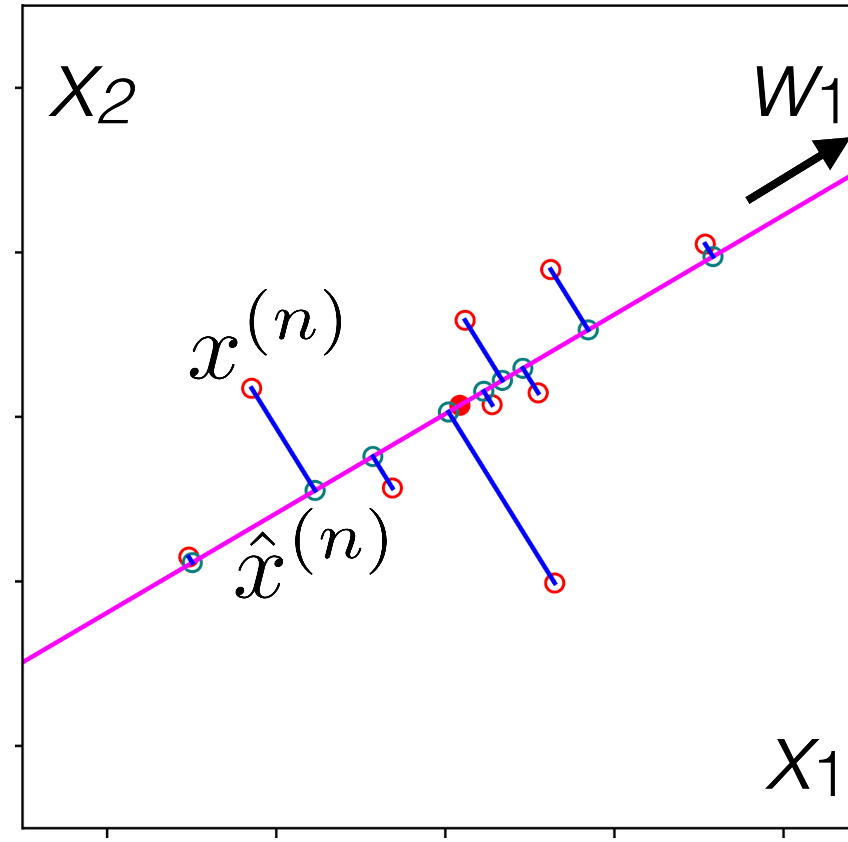
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}}\right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

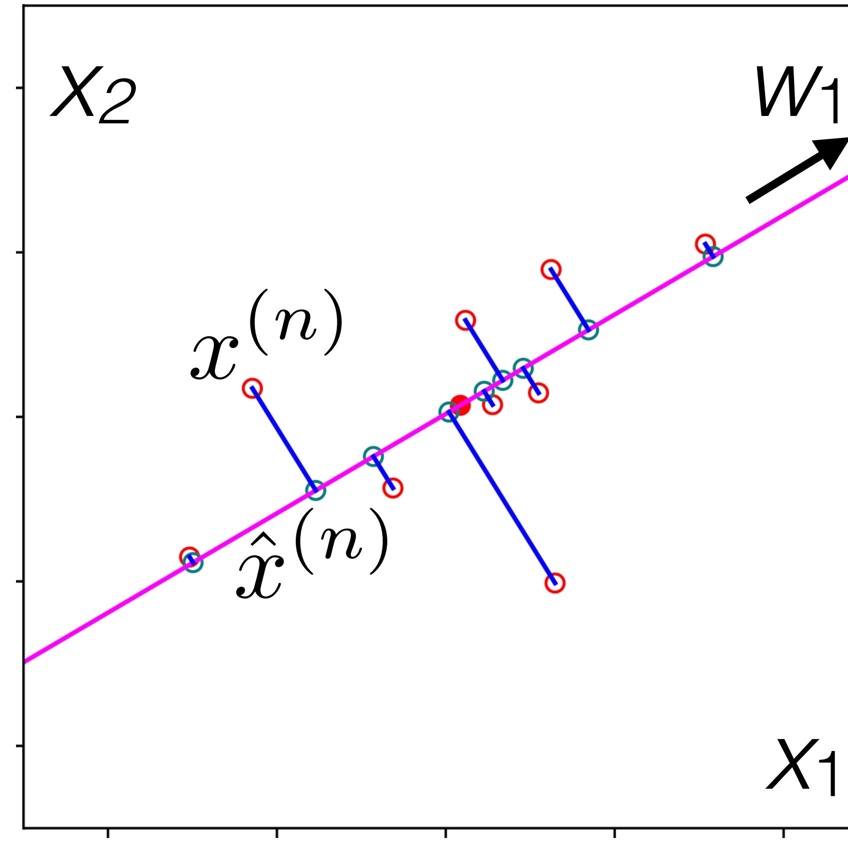
$$-\sum_{n=1}^{N}(w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)}(x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}}\right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1(w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \underbrace{\left[\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top\right]}_{\text{empirical covariance } \hat{\Sigma}} w_1$$

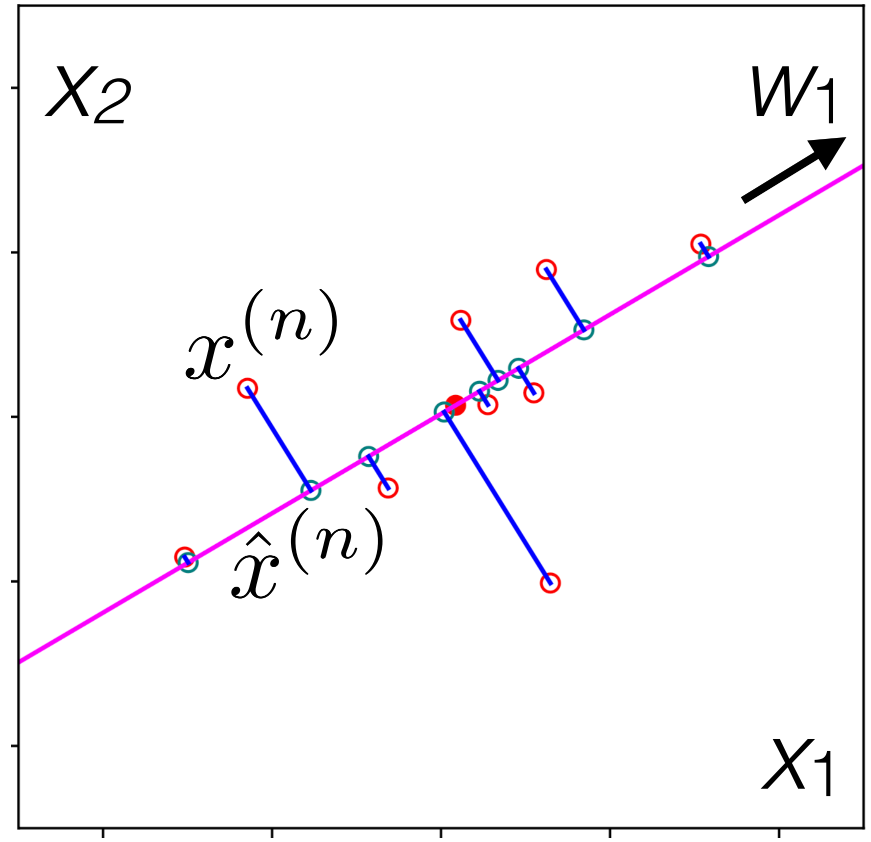
corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$

- Best w_1: eigenvec of covariance

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

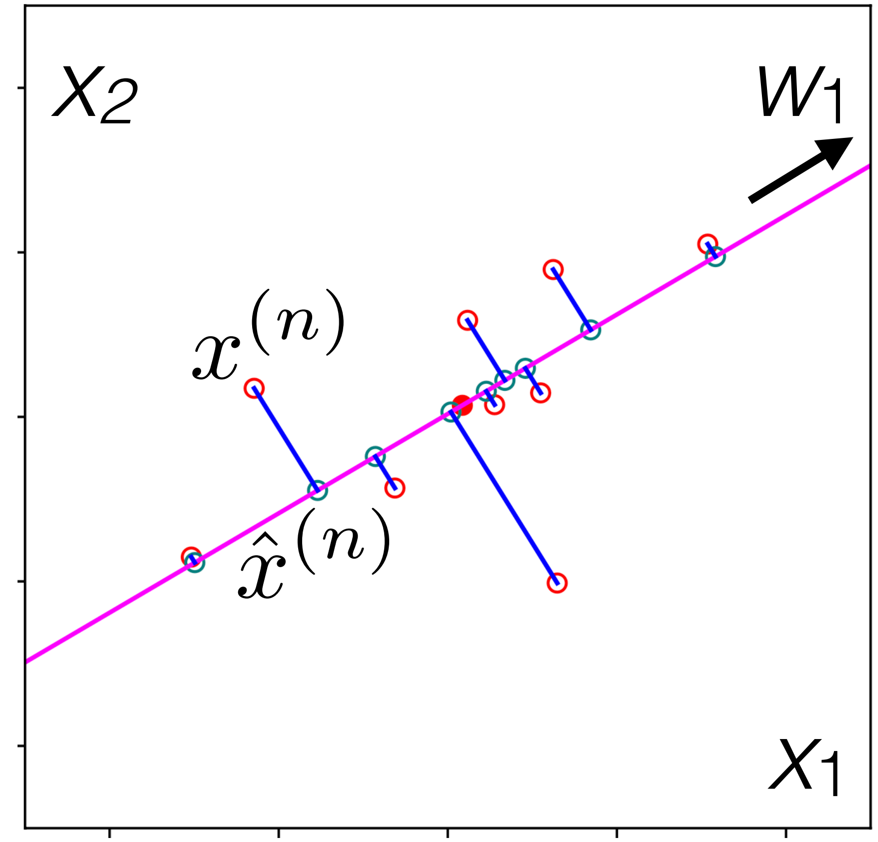
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}} \right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$
  - Best w_1: eigenvec of covariance: which one?

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

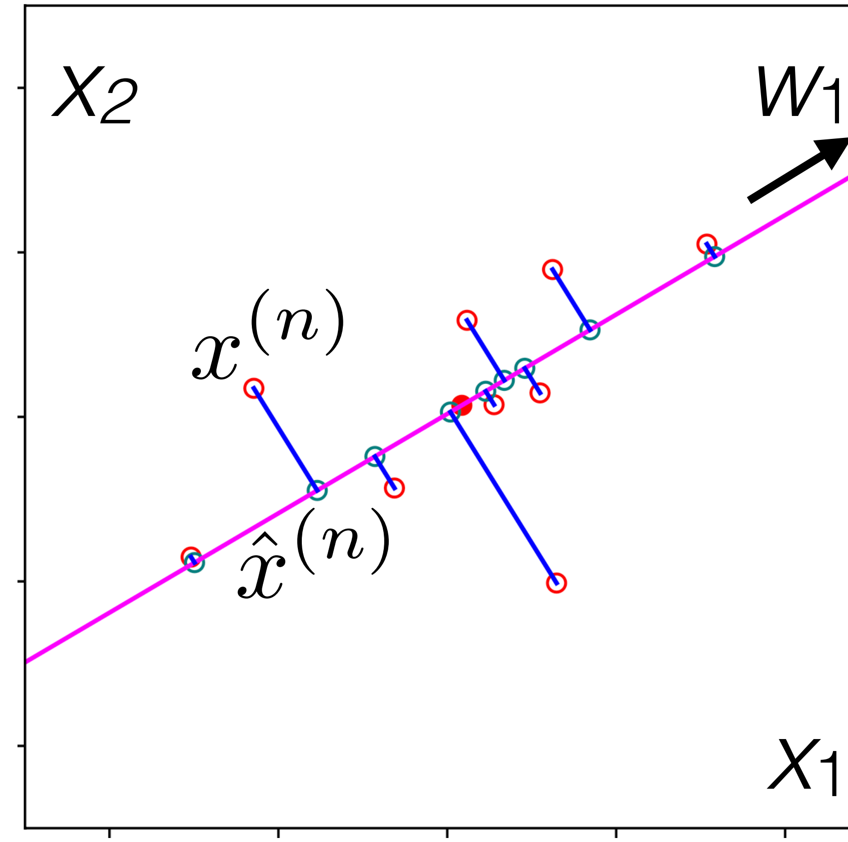
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}} \right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$
  - Plugging in to the objective:
  - Best w_1: eigenvec of covariance: which one?

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

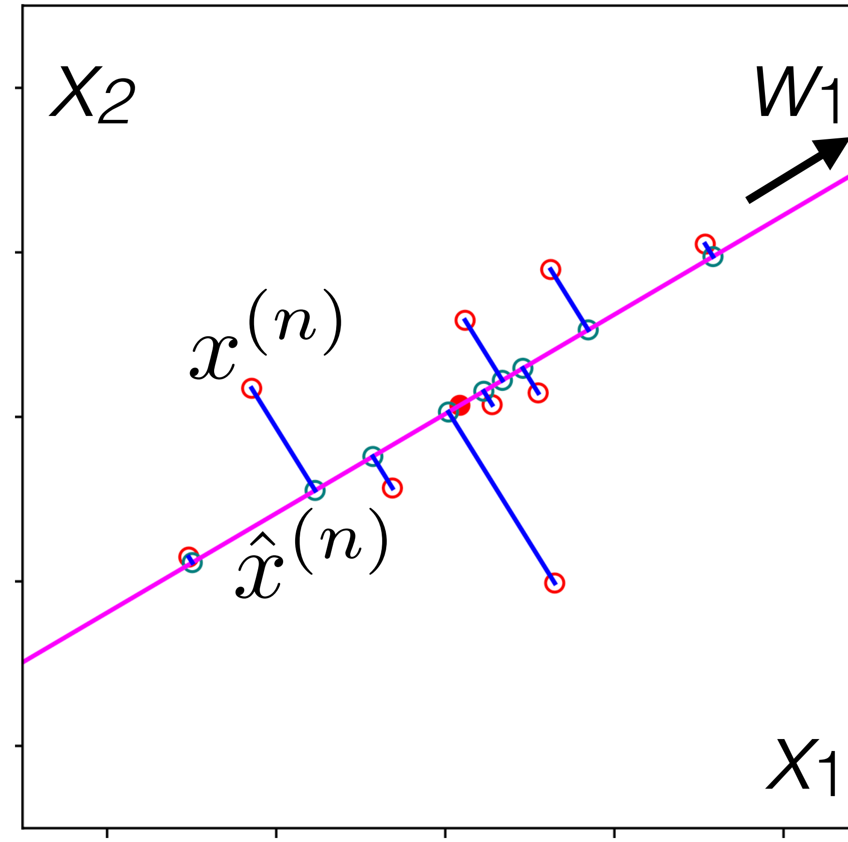
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \Big[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}}\Big] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$
  - Plugging in to the objective:
  - Best w_1: eigenvec of covariance: which one?

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}} \right] w_1$$

corrected from live lecture!

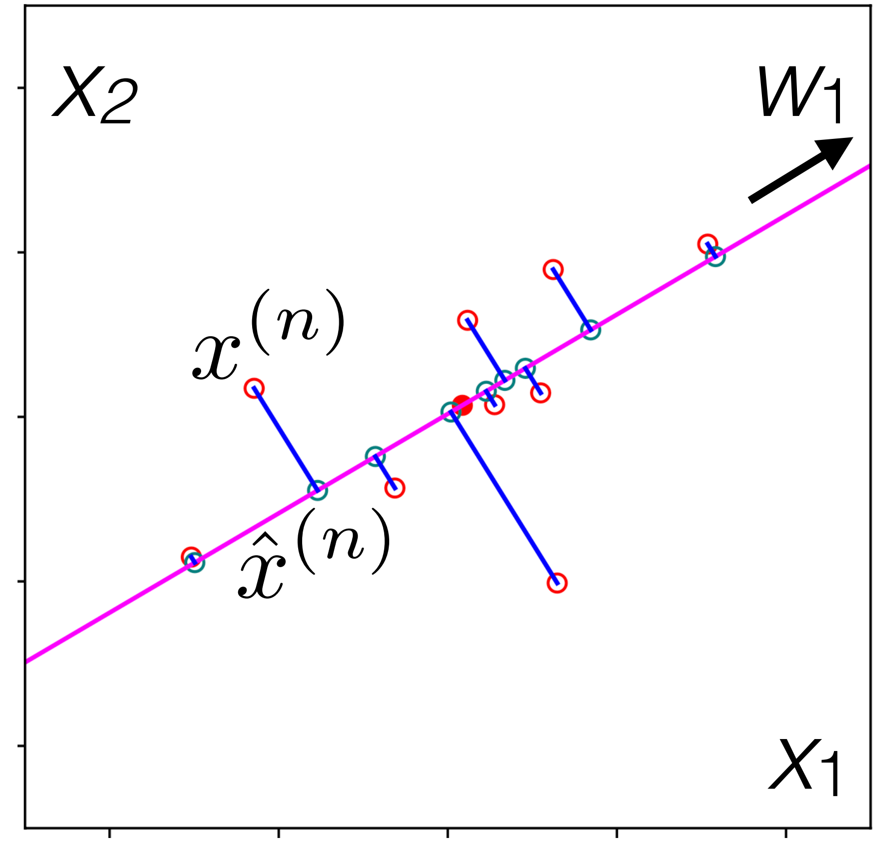
to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$
  - Plugging in to the objective: $-w_1^\top \hat{\Sigma} w_1$
  - Best w_1: eigenvec of covariance: which one?

Solving when L=1

- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}_{\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}}$$



- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

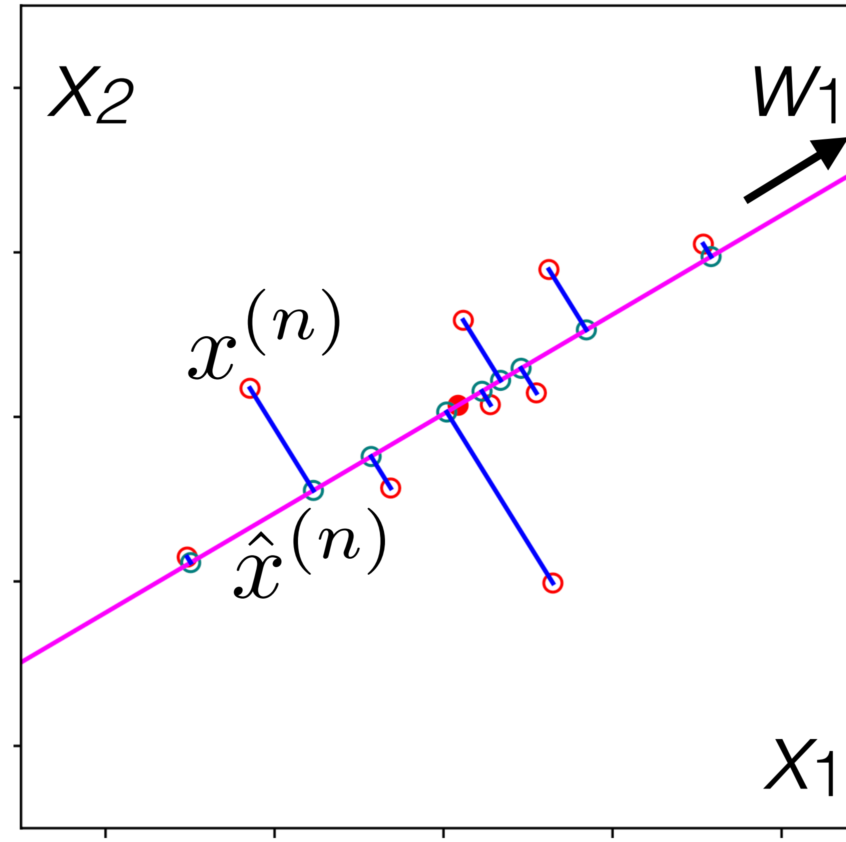
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \Big[\underbrace{\textstyle\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}}\Big] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$
  - Plugging in to the objective: $-w_1^\top \hat{\Sigma} w_1 = -\lambda_1$
  - Best w_1: eigenvec of covariance: which one?

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

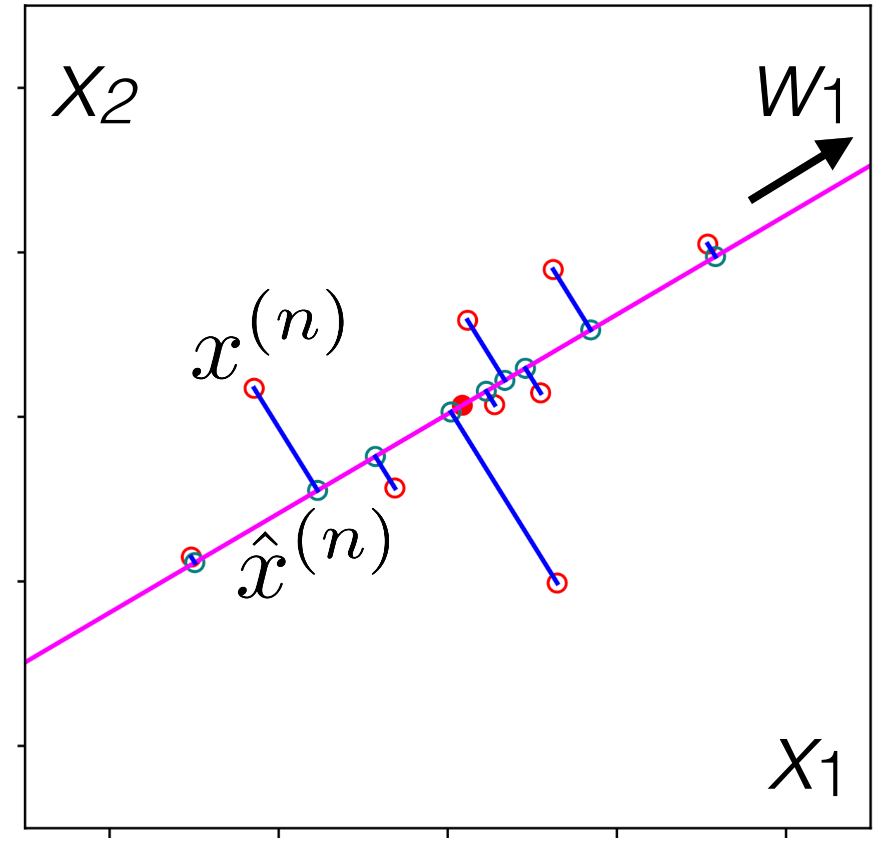
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}}\right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$
  - Plugging in to the objective: $-w_1^\top \hat{\Sigma} w_1 = -\lambda_1$
  - Best w_1: eigenvec of covariance: which one?

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

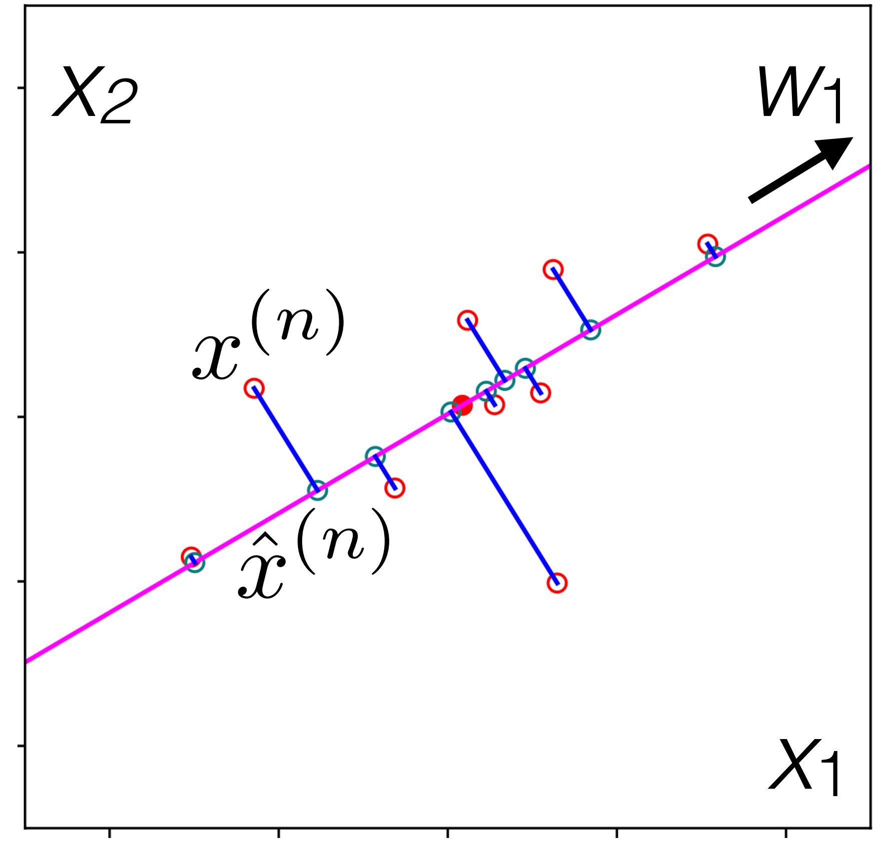
$$-\sum_{n=1}^{N}(w_1^\top x^{(n)})^2 = -w_1^\top \left[\underbrace{\sum_{n=1}^{N} x^{(n)}(x^{(n)})^\top}_{\text{empirical covariance } \hat{\Sigma}}\right] w_1$$

corrected from live lecture!

to include constraint:

- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1(w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$
  - Plugging in to the objective: $-w_1^\top \hat{\Sigma} w_1 = -\lambda_1$
  - Best w_1: eigenvec of covariance: which one?

Solving when L=1



- Assume: $\frac{1}{N}\sum_{n=1}^{N} x^{(n)} = 0_D$
  - & w_1 orthonormal basis (unit vector)

$$\min \sum_{n=1}^{N} \underbrace{\|x^{(n)} - z_1^{(n)} w_1\|^2}$$

$$\underbrace{(x^{(n)})^\top x^{(n)}}_{\text{constant in } Z \text{ and } W} - 2 z_1^{(n)} w_1^\top x^{(n)} + (z_1^{(n)})^2 \underbrace{w_1^\top w_1}_{1}$$

- Take derivative with respect to $z_1^{(n)}$ & set to 0: $z_1^{(n)} = w_1^\top x^{(n)}$

second order condition: check

scalar projection of the data in the w_1 direction

- Plug back in to the objective; we want to minimize

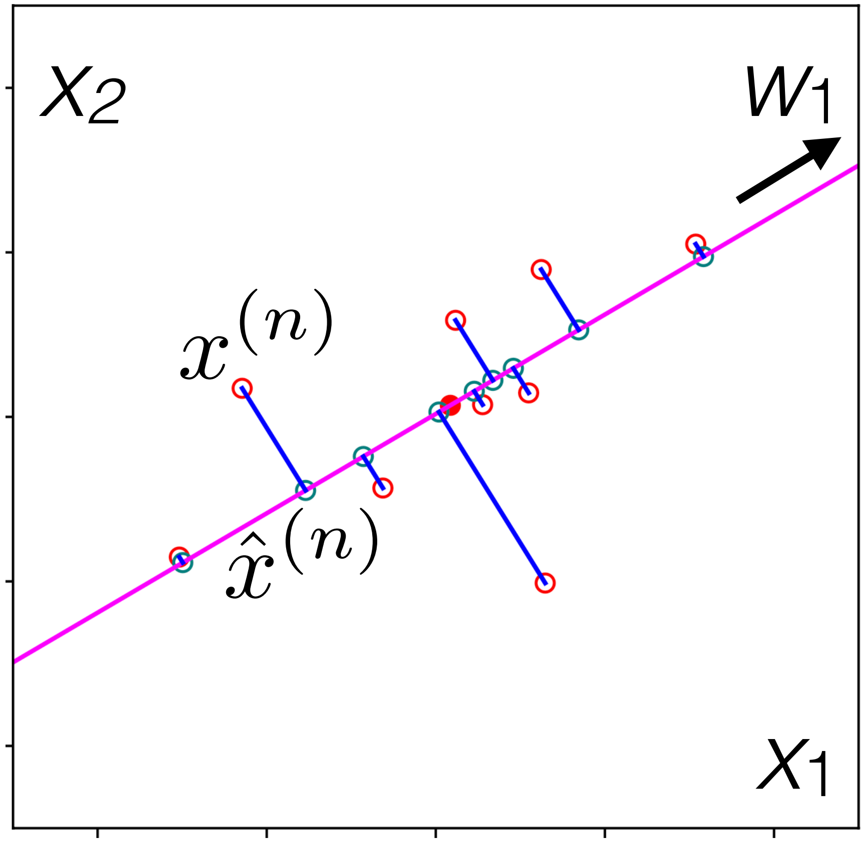
$$-\sum_{n=1}^{N} (w_1^\top x^{(n)})^2 = -w_1^\top \underbrace{\left[\sum_{n=1}^{N} x^{(n)} (x^{(n)})^\top\right]}_{\text{empirical covariance } \hat{\Sigma}} w_1$$

corrected from live lecture!

to include constraint:

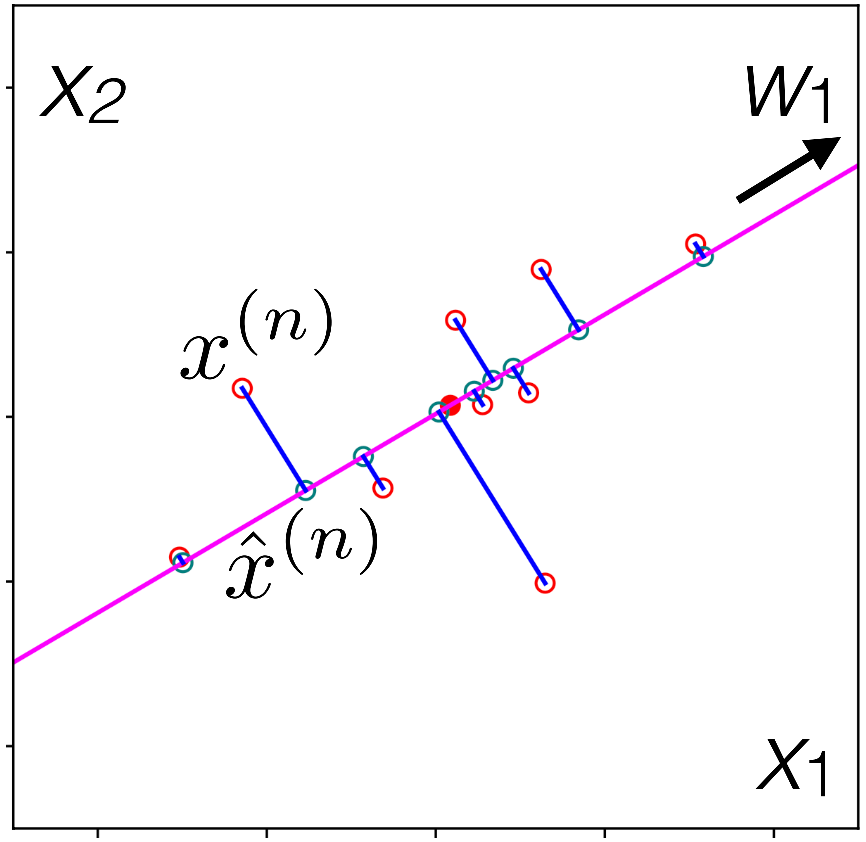
- Lagrangian: $w_1^\top \hat{\Sigma} w_1 - \lambda_1 (w_1^\top w_1 - 1)$
  - Take derivative w.r.t. w_1 & set to 0: $\hat{\Sigma} w_1 = \lambda_1 w_1$
  - Plugging in to the objective: $-w_1^\top \hat{\Sigma} w_1 = -\lambda_1$
  - Best w_1: eigenvec of covariance with largest eigenvalue

More on the solution



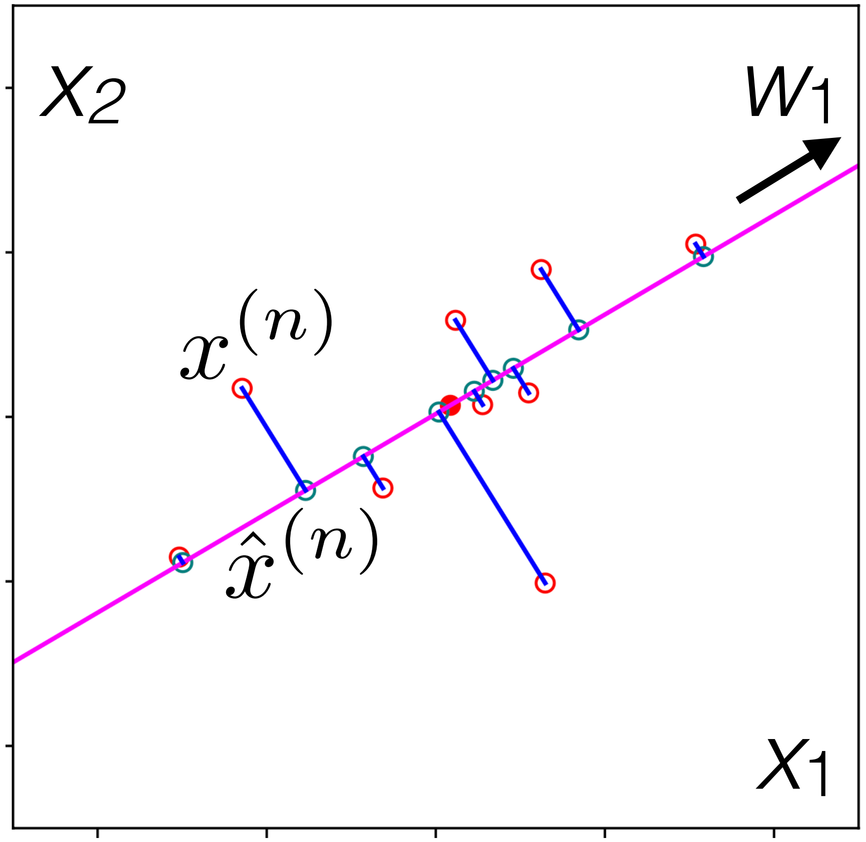
More on the solution

- For $L > 1$, can solve inductively



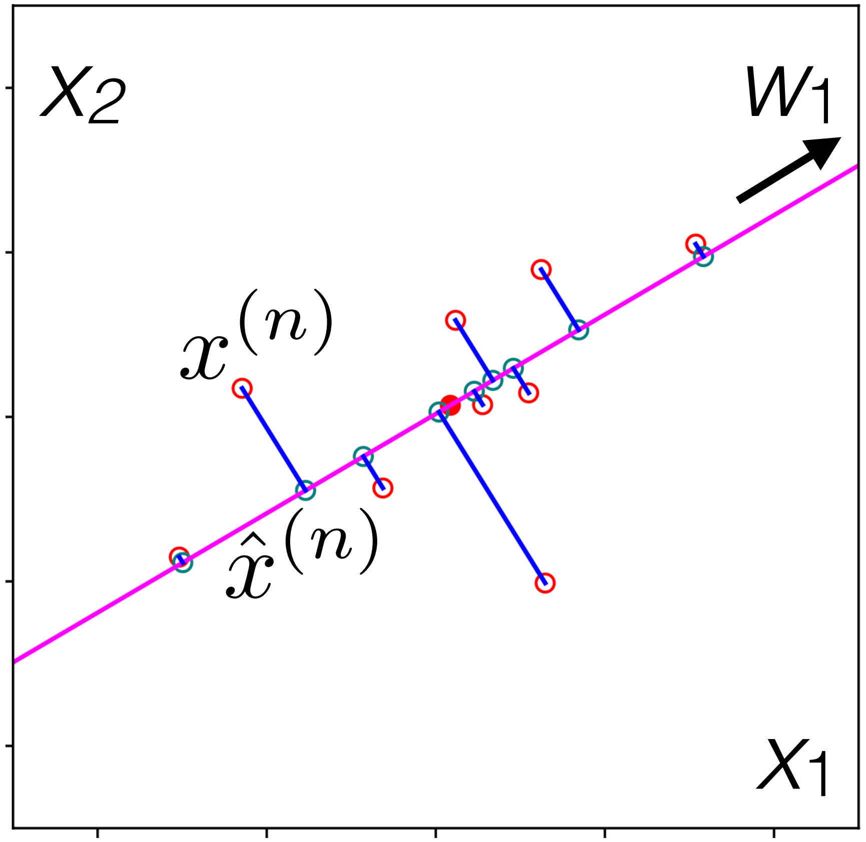
More on the solution

- For $L > 1$, can solve inductively
  - When get to Lagrangian step, now use constraints:



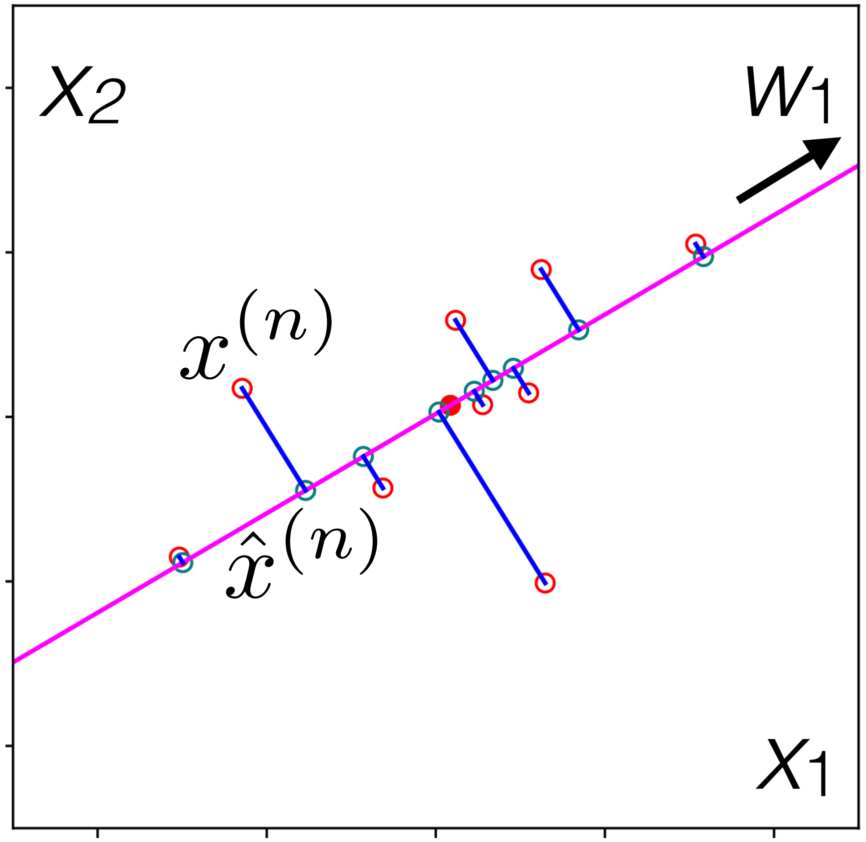
More on the solution

- For $L > 1$, can solve inductively
  - When get to Lagrangian step, now use constraints:
    - Next basis vector is a unit vector

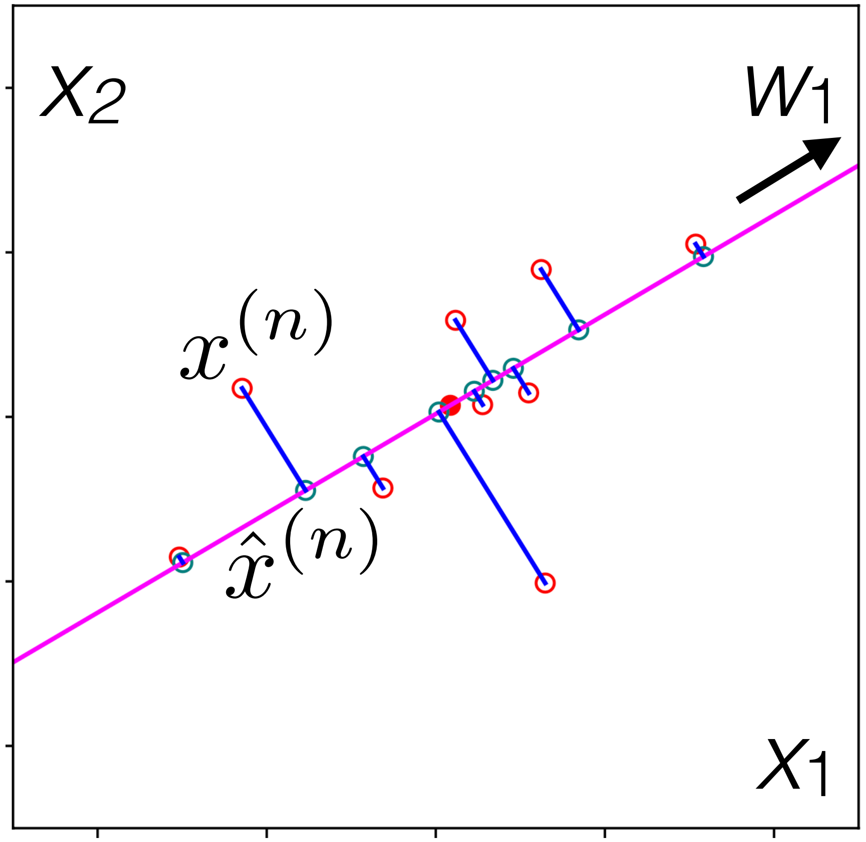


More on the solution

- For $L > 1$, can solve inductively
  - When get to Lagrangian step, now use constraints:
    - Next basis vector is a unit vector
    - Next basis vector is orthogonal to all previous vectors



More on the solution



- For $L > 1$, can solve inductively
  - When get to Lagrangian step, now use constraints:
    - Next basis vector is a unit vector
    - Next basis vector is orthogonal to all previous vectors

$$w_\ell^\top w_\ell = 1 \quad \& \quad \forall k \in \{1, \ldots, \ell - 1\}, w_\ell^\top w_k = 0$$

References (1/1)

Gelman, A. (2007). Data analysis using regression and multilevel/hierarchical models. Cambridge university press.Neal, Radford M. (1994). Priors for infinite networks (tech. rep. no. crg-tr-94-1). University of Toronto.

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

Petito, L. (2023). Unseen Worlds: How Missing Data Impact: Statistical Analyses https://www.feinberg.northwestern.edu/sites/bcc/docs/2022-2023-lectures/petito-slides-unseen-worlds.pdf

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

Tierney, N. (2004). Gallery of Missing Data Visualisations. https://cran.r-project.org/web/packages/naniar/vignettes/naniar-visualisation.html