Unsupervised Learning: Principal Component Analysis

KMA Solaiman

Partially Adapted from
Chris Ré and Zilinkas

Our Tour Through Unsupervised Land

Structure	Probabilistic	Not Probabilistic
“Cluster”	GMM	k-Means
“Subspace”	Factor Analysis	PCA

We can impose other structures. These are popular.

Outline

Linear Algebra/Math Review

Two Methods of Dimensionality Reduction

- Linear Discriminant Analysis (LDA, LDiscA)
- Principal Component Analysis (PCA)

Covariance

covariance: how (linearly) correlated are variables

$$\sigma_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ki} - \mu_i)(x_{kj} - \mu_j)$$

covariance of variables *i* and *j*

Value of variable *i* in object k

Mean of variable *i*

Value of variable *j* in object k

Mean of variable *j*

Covariance

covariance: how (linearly) correlated are variables

$$\sigma_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ki} - \mu_i)(x_{kj} - \mu_j)$$

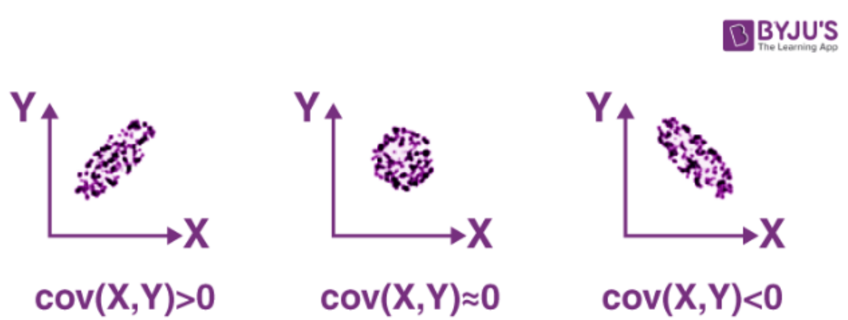
covariance of variables *i* and *j*

Value of variable *i* in object k

Mean of variable *i*

Value of variable *j* in object k

Mean of variable *j*



Covariance

covariance: how (linearly) correlated are variables

$$\sigma_{ij} = \frac{1}{N-1} \sum_{k=1}^{N} (x_{ki} - \mu_i)(x_{kj} - \mu_j)$$

Mean of variable *i*

Mean of variable *j*

covariance of variables *i* and *j*

Value of variable *i* in object k

Value of variable *j* in object k

$$\sigma_{ij} = \sigma_{ji}$$

$$\Sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1K} \\ \vdots & \ddots & \vdots \\ \sigma_{K1} & \cdots & \sigma_{KK} \end{pmatrix}$$

Eigenvalues and Eigenvectors

vector

$$Ax = \lambda x$$

matrix

scalar

for a given matrix operation (multiplication):

what non-zero vector(s) change linearly?

(by a single multiplication)

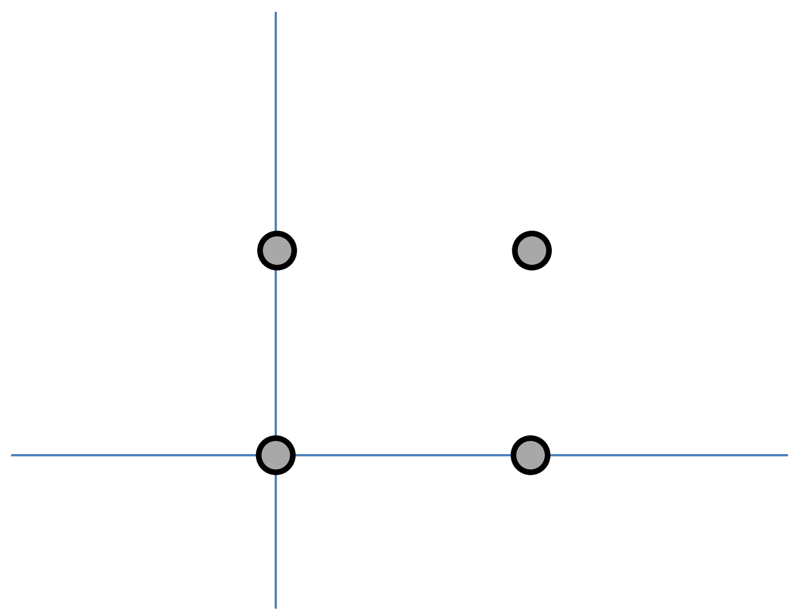
Eigenvalues and Eigenvectors

vector

$$Ax = \lambda x$$

matrix | scalar

$$A = \begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix}$$



Eigenvalues and Eigenvectors

vector

$$Ax = \lambda x$$

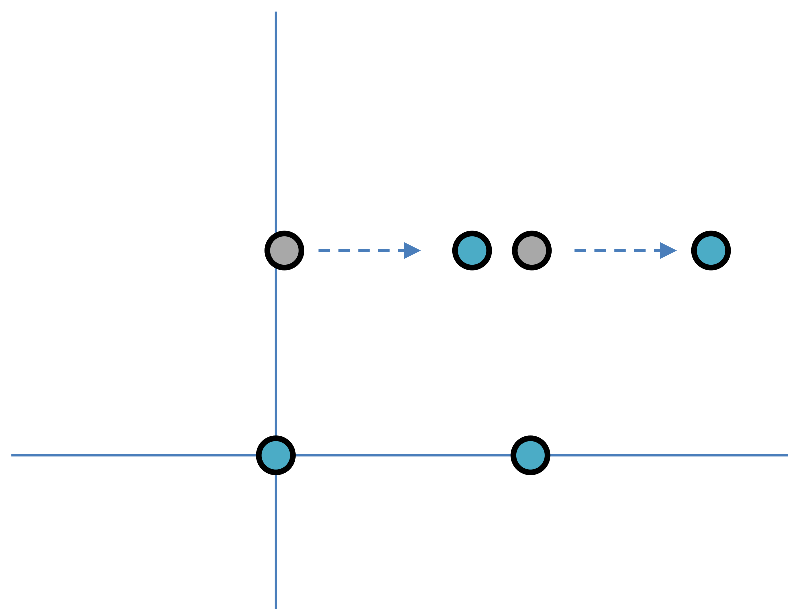
matrix

scalar

$$\begin{pmatrix} x + 5y \\ y \end{pmatrix} = \lambda \begin{pmatrix} x \\ y \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x + 5y \\ y \end{pmatrix}$$



Eigenvalues and Eigenvectors

vector

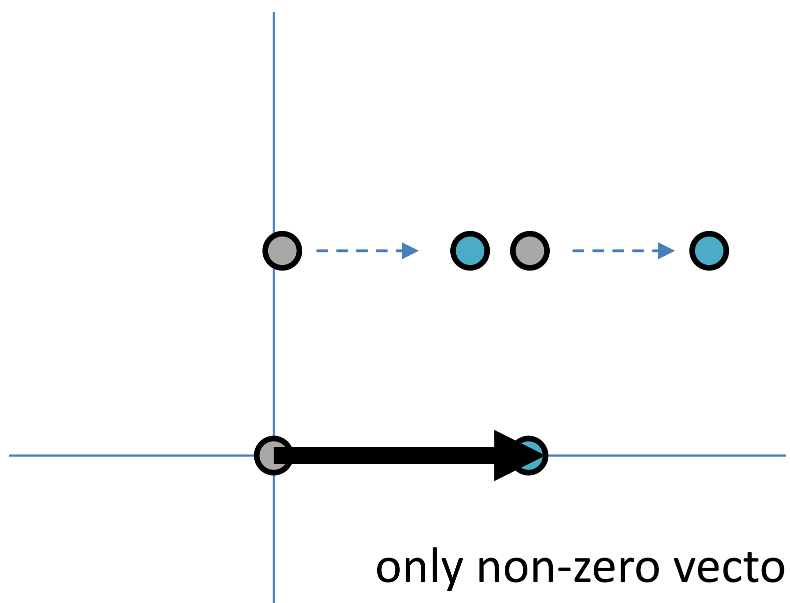
$$Ax = \lambda x$$

matrix scalar

$$\begin{pmatrix} x + 5y \\ y \end{pmatrix} = \lambda \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1 \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 5 \\ 0 & 1 \end{pmatrix}$$

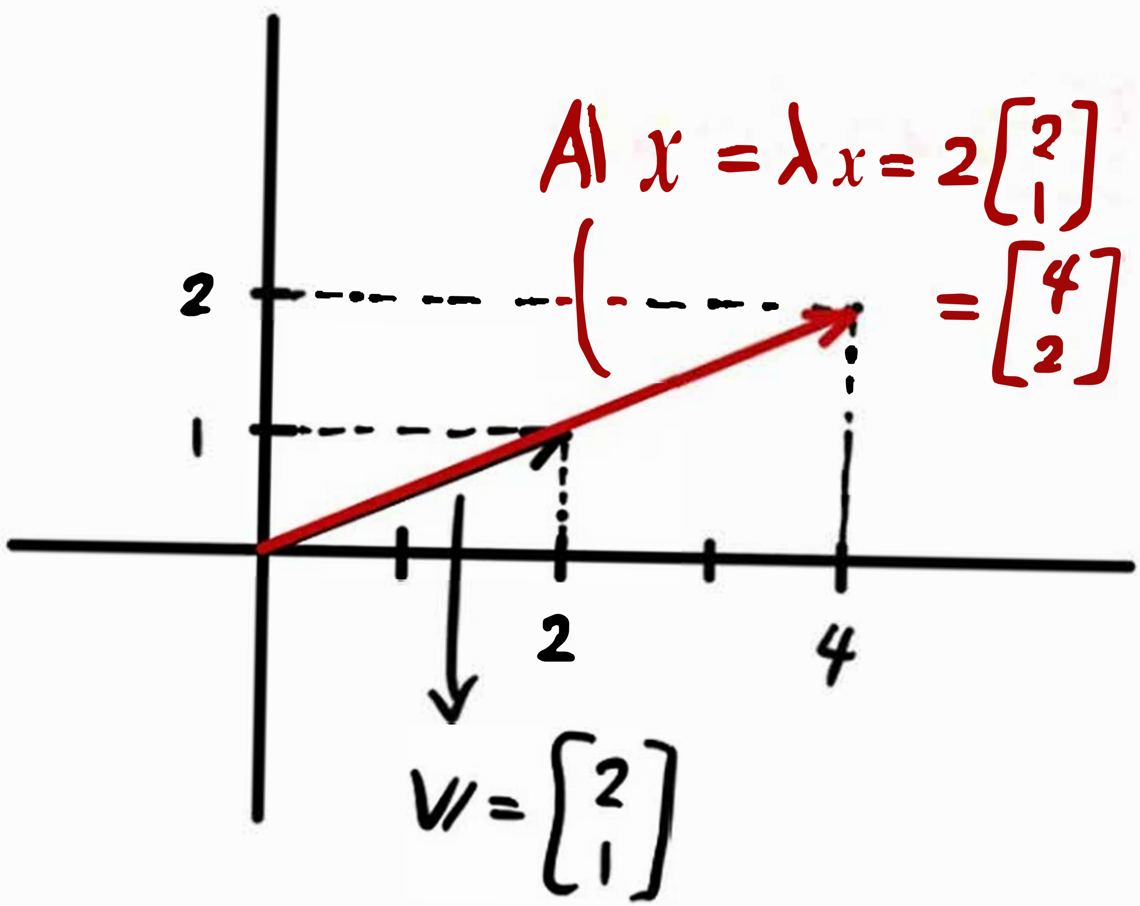


only non-zero vector to scale

(i) $\lambda_1 = 2$

$$\begin{bmatrix} 1 & 2 \\ 3 & -4 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2+2 \\ 6-4 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} = 2\begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$= \lambda_1 v$$



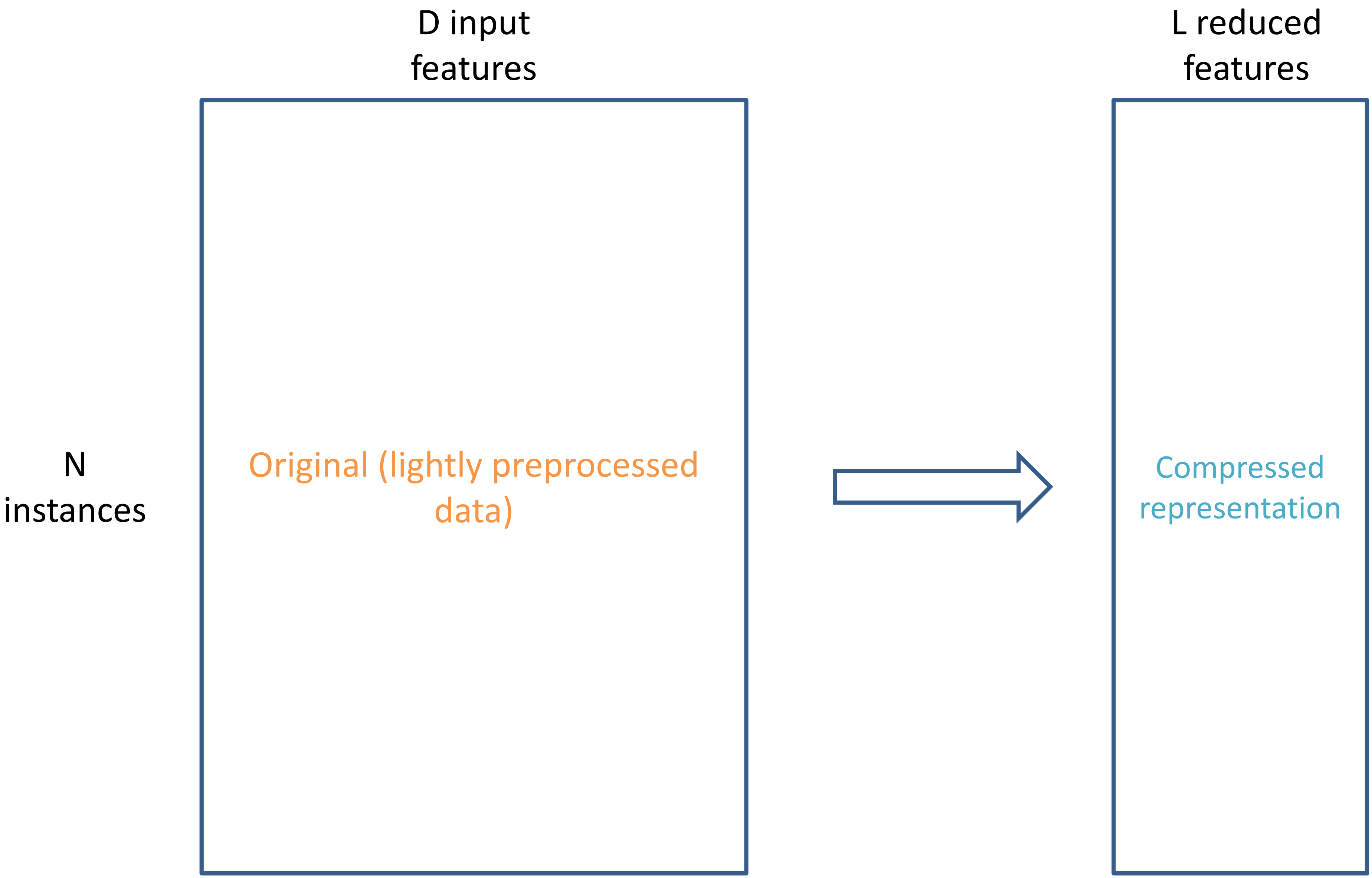
Outline

Linear Algebra/Math Review

Two Methods of Dimensionality Reduction

- Linear Discriminant Analysis (LDA, LDiscA)
- Principal Component Analysis (PCA)

Dimensionality Reduction



Dimensionality Reduction

clarity of representation vs. ease of understanding

oversimplification: loss of important or relevant information

Courtesy Antano Žilinsko

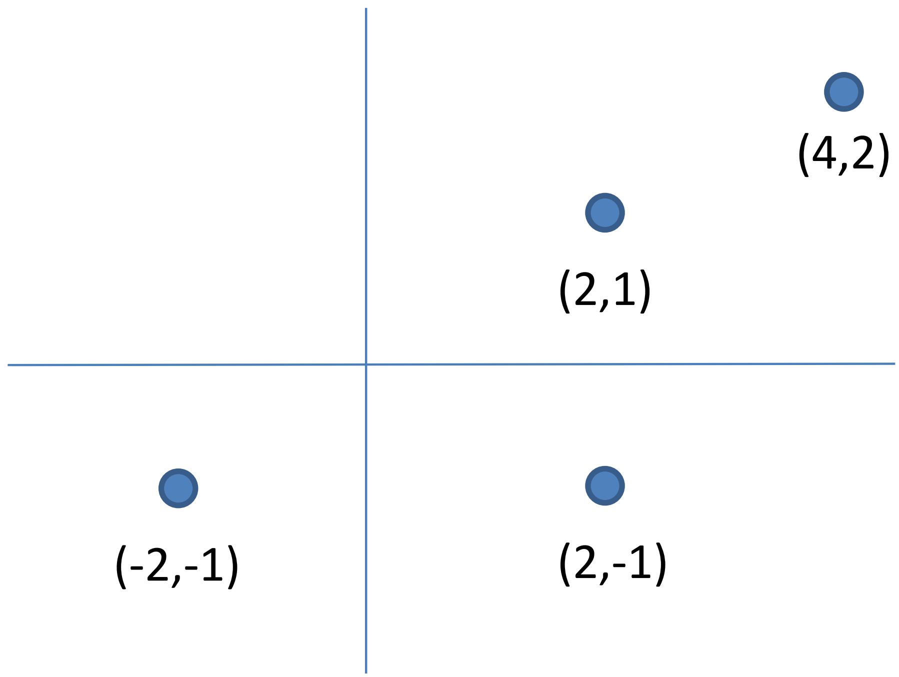
Why “maximize” the variance?

How can we efficiently summarize? We **maximize the variance** within our summarization

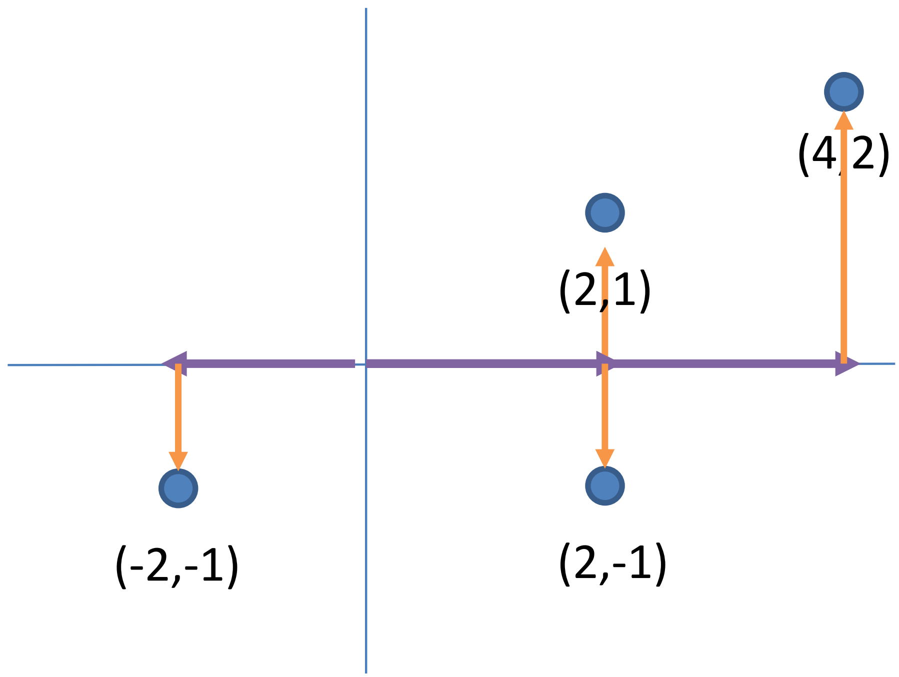
We **don’t increase the variance in the dataset**

How can we capture the most information with the fewest number of axes?

Summarizing Redundant Information

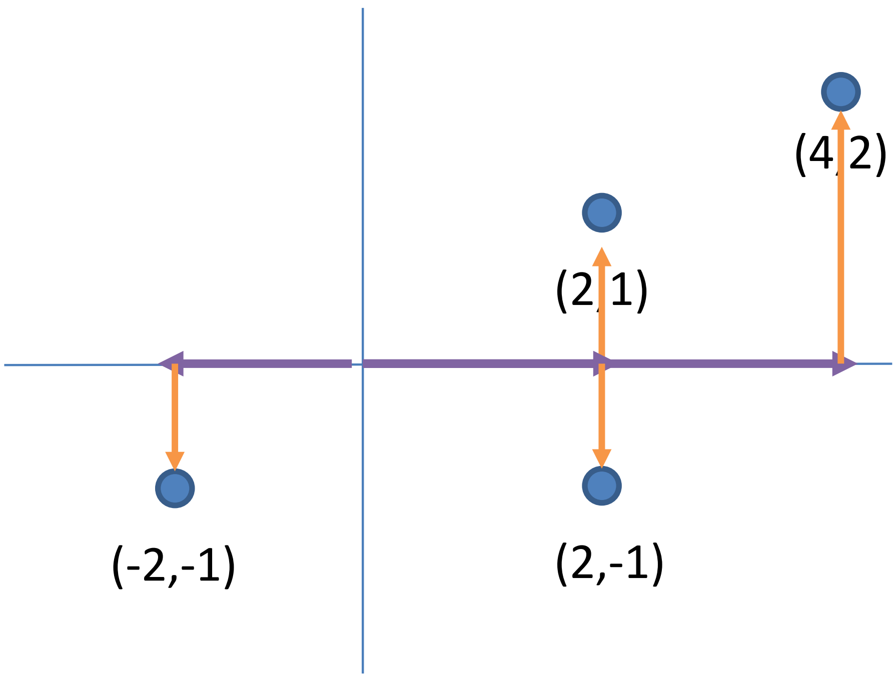


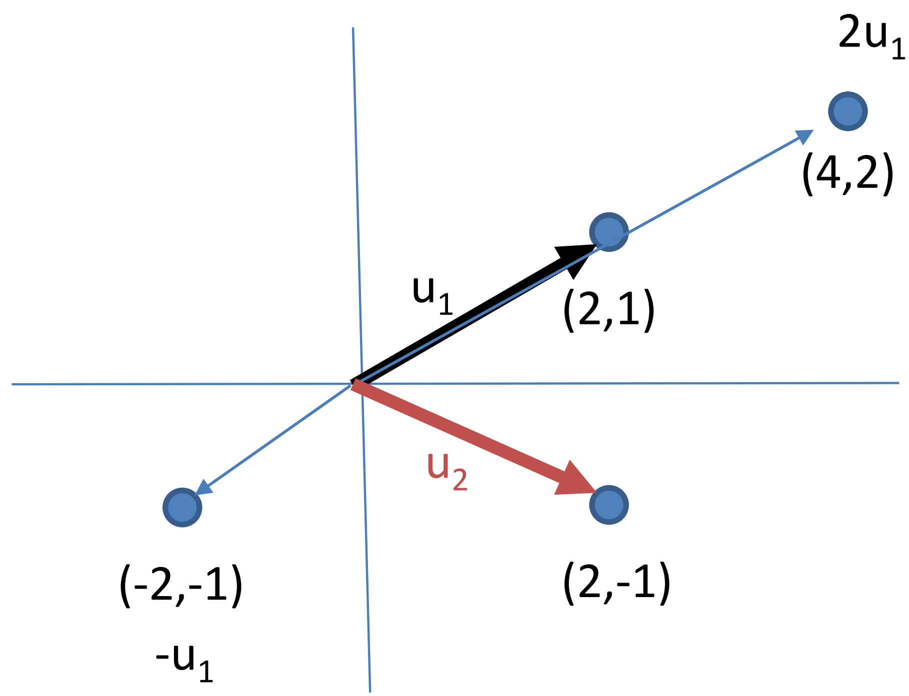
Summarizing Redundant Information



(2,1) = 2*(1,0) + 1*(0,1)

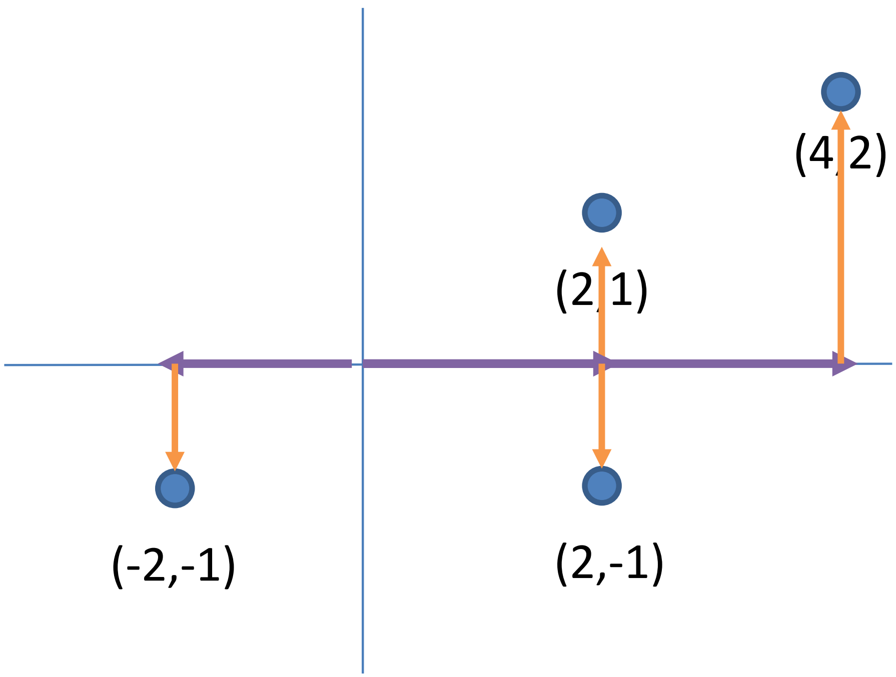
Summarizing Redundant Information

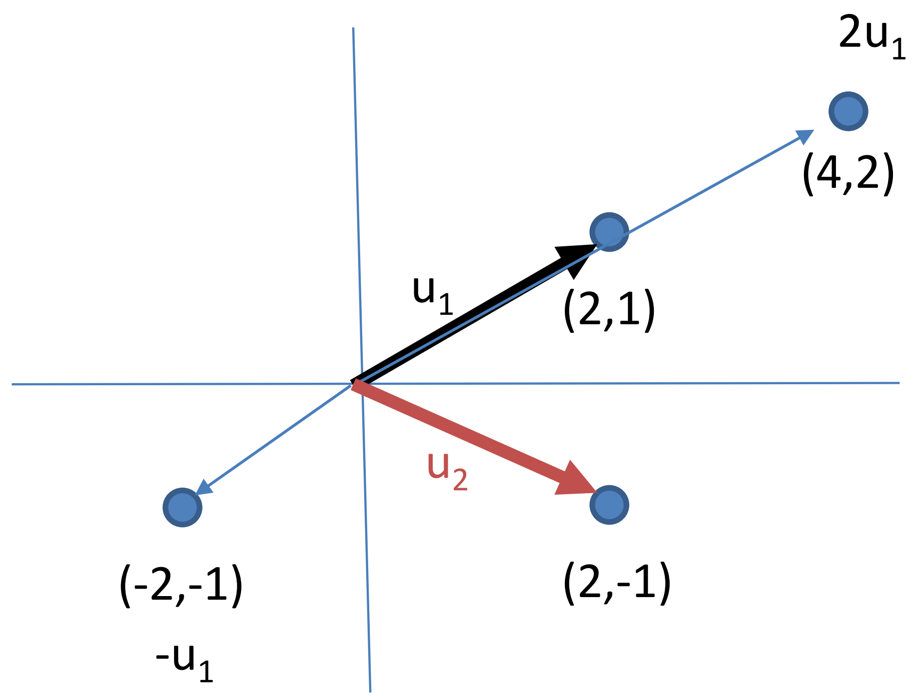




(2,1) = 1*(2,1) + 0*(2,-1)
(4,2) = 2*(2,1) + 0*(2,-1)

Summarizing Redundant Information





$(2,1) = 1*(2,1) + 0*(2,-1)$
$(4,2) = 2*(2,1) + 0*(2,-1)$

(Is it the most general? These vectors aren't orthogonal)

Algorithm 37 PCA($\mathbf{D}$, K)

$\boldsymbol{\mu} \leftarrow$ MEAN($\mathbf{X}$) // compute data mean for centering

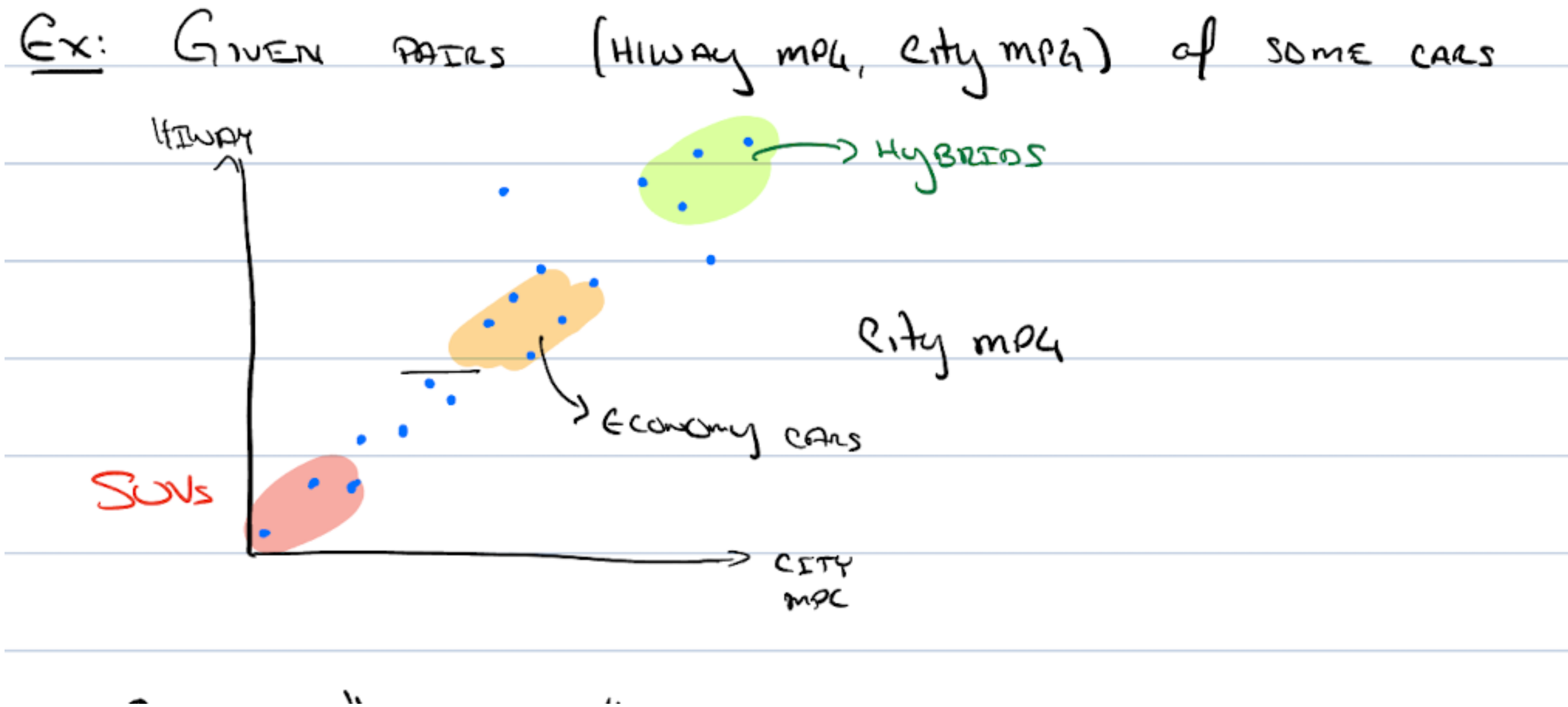
$\mathbf{D} \leftarrow \left(\mathbf{X} - \boldsymbol{\mu}\mathbf{1}^\top\right)^\top \left(\mathbf{X} - \boldsymbol{\mu}\mathbf{1}^\top\right)$ // compute covariance, **1** is a vector of ones

$\{\lambda_k, \boldsymbol{u}_k\} \leftarrow$ top K eigenvalues/eigenvectors of $\mathbf{D}$

return $(\mathbf{X} - \boldsymbol{\mu}\mathbf{1})\,\mathbf{U}$ // project data using **U**

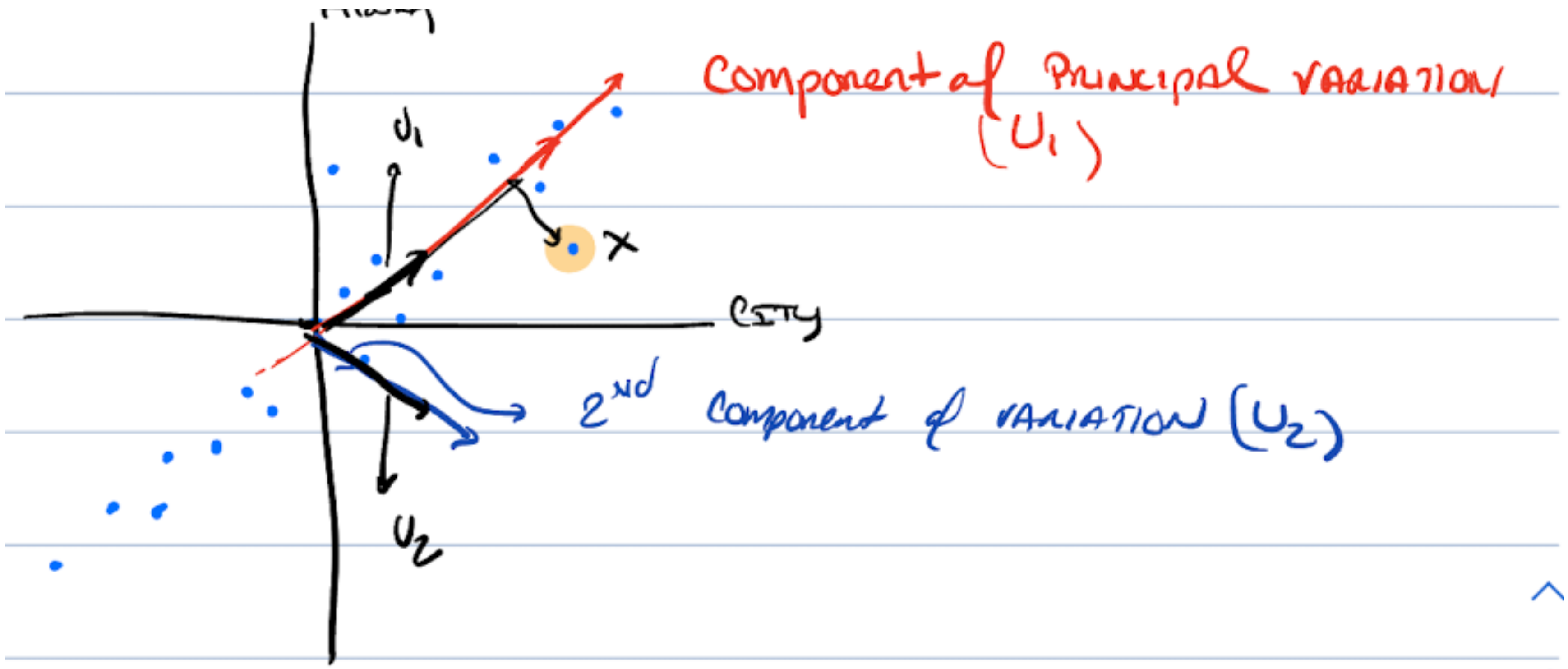
PCA Example: MPG

Given pairs (Highway MPG, City MPG) of some cars.



Question: What is "good" MPG?

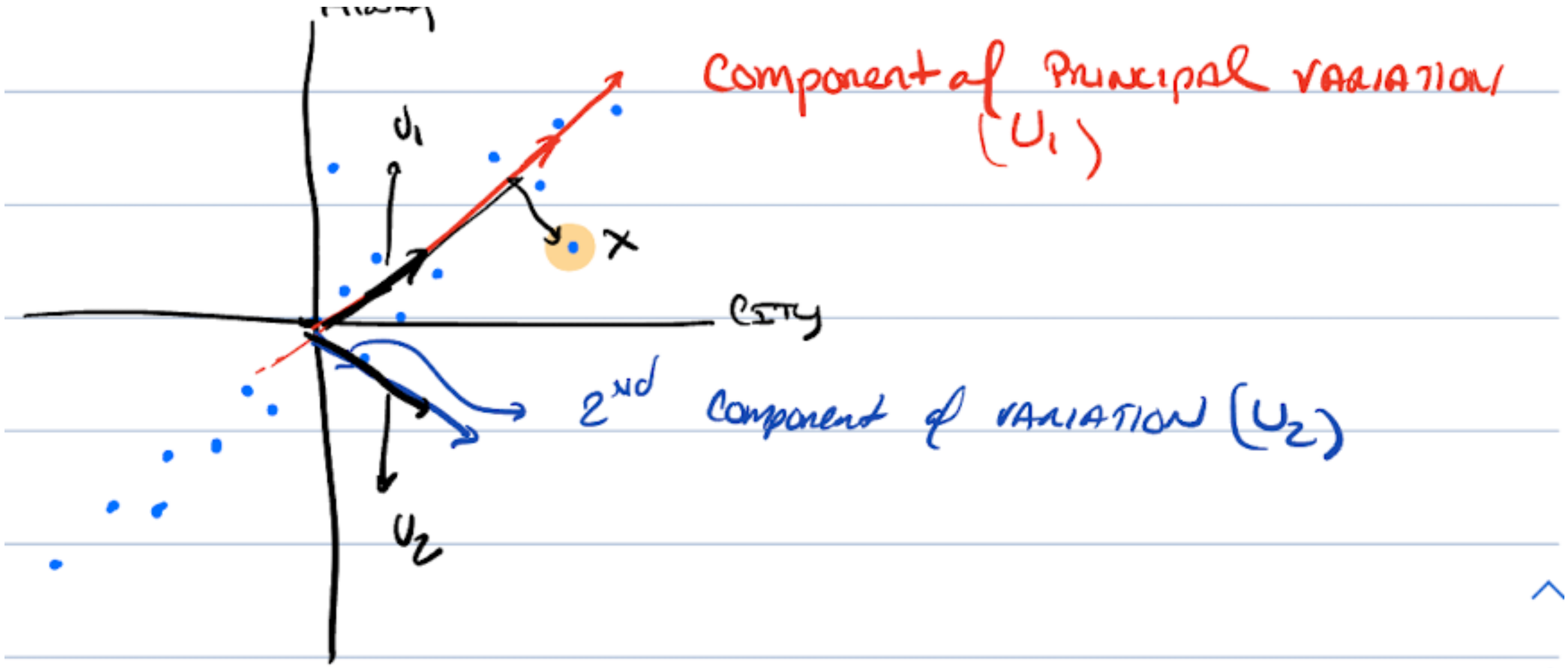
Center the data



We *center* the data, i.e., as preprocessing.

$$x^{(i)} \mapsto x^{(i)} - \mu \text{ where } \mu = \frac{1}{n}\sum_{i=1}^{n} x^{(i)}.$$

Finding Components



By convention, $\|u_1\| = \|u_2\| = 1$ by convention.

- u_1 is the first **principal component** “how good is the MPG”
- u_2 is the second, and roughly the difference.

Recall: any point can be written in an orthogonal basis:

$$x = \alpha_1 u_1 + \alpha_2 u_2$$

Goals

- How do we find these directions?
- Some caveats about how to use these?
- Reduce dimensions: Think about $D = 1000$ reduced to $d = 10$.

Preprocessing

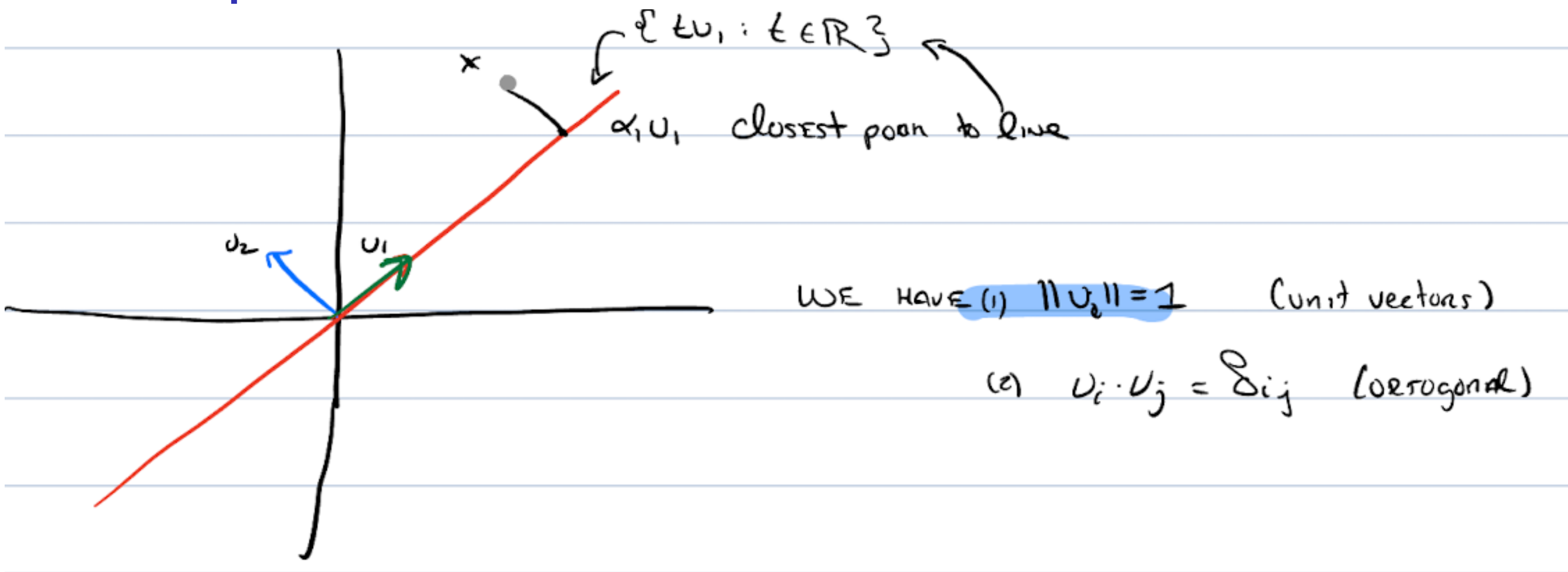
Given $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ we preprocess:

- **Center the data** $x^{(i)} \mapsto x^{(i)} - \mu$
- **Rescale the data** May need to rescale components, e.g., "Feet per gallon" v. "Miles per Gallon"

$$x^{(i)} \mapsto \frac{x^{(i)} - \mu}{\sigma}.$$

We will assume from now on that the data is preprocessed.

PCA As Optimization



How do you find the closest point to the line?

$$
\begin{aligned}
\alpha_1 &= \underset{\alpha}{\operatorname{argmin}} \|x - \alpha u_1\|^2 \\
&= \underset{\alpha}{\operatorname{argmin}} \|x\|^2 + \alpha^2 \|u_1\|^2 - 2\alpha u_1^T x
\end{aligned}
$$

Then, differentiate wrt α, set to 0, and use $\|u_1\|^2$, which leads to:

$$
2\alpha - 2u_1^T x = 0 \implies \alpha = u_i^T x.
$$

Generalize to higher dimensions

Suppose we have a $u_1, \ldots, u_k \in \mathbb{R}^d$ with $u_i \cdot u_j = \delta_{i,j}$. Then,

$$= \underset{\alpha_1,\ldots,\alpha_k \in R}{\operatorname{argmin}} \|x - \sum_{i=1}^{k} \alpha_i u_i\|^2$$

$$= \underset{\alpha_1,\ldots,\alpha_k \in R}{\operatorname{argmin}} \|x\|^2 + \sum_{i=1}^{k} \alpha_i^2 - 2\alpha_i (u_i \cdot x)$$

These are k independent minimizations, so $\alpha_i = u_i \cdot x$.

- This process is also known as **projecting** on to the set spanned by the vectors $\{u_1, \ldots, u_k\}$.
- We call $\|x - \sum_{i=1}^{k} \alpha_i u_i\|^2$ the **residual**.

Finding PCA

There are two ways you can find PCA:

- Maximize the projected subspace of the data. (we see more)

$$\max_{u \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (u \cdot x^{(i)})^2.$$

- Minimize the residual

$$\min_{u \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (x^{(i)} - u \cdot x^{(i)})^2.$$

We need to recall some more linear algebra to solve this.

Recall: Eigenvalue decomposition

Let $A \in \mathbb{R}^{d \times d}$ be symmetric (and square) then there exists $U, \Lambda \in \mathbb{R}^{d \times d}$ such that

$$A = U \Lambda U^T \text{ in which } UU^T = I \text{ and } \Lambda \text{ is diagonal.}$$

- If $U = [u_1, \ldots, u_d]$, $UU^T = I$ can also be written $u_i \cdot u_j = \delta_{i,j}$.
- In this decomposition,

$$\Lambda_{i,i} = \lambda_i \text{ is called an } \textbf{eigenvalue}.$$

  and by convention, we order them $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$.
- For $i = 1, \ldots, d$, u_i is the eigenvector associated with λ_i:

$$Au_i = \lambda u_i \text{ since } Au_i = U \Lambda U^T u_i = \lambda_i U e_i = \lambda u_i$$

  here e_i is the ith standard basis vector.

Back to PCA!

$$\max_{u \in \mathbb{R}^d : \|u\|^2 = 1} \frac{1}{n} \sum_{i=1}^{n} (u \cdot x^{(i)})^2$$

We can write:

$$\frac{1}{n} \sum_{i=1}^{n} (u \cdot x^{(i)})^2 = \frac{1}{n} \sum_{i=1}^{n} u^T x^{(i)} (x^{(i)})^T u = u^T \left(\underbrace{\frac{1}{n} \sum_{i=1}^{n} x^{(i)} (x^{(i)})^T}_{C} \right) u.$$

C is the covariance of the data, since we subtracted the mean.

> The first eigenvector of the data's covariance matrix is the principal component

More PCA

- **Multiple Dimensions** What if we want multiple dimensions? We keep the top-k.

$$\max_{U \in \mathbb{R}^{k \times d} : UU^T = I_k} \frac{1}{n} \sum_{u=1}^{n} \| Ux^{(i)} \|^2.$$

More PCA

- **Multiple Dimensions** What if we want multiple dimensions? We keep the top-k.

$$\max_{U \in \mathbb{R}^{k \times d} : UU^T = I_k} \frac{1}{n} \sum_{u=1}^{n} \| U x^{(i)} \|^2.$$

- **Reduce dimensionality**. How do we represent data with just those $k < d$ scalars α_j for $j = 1, \ldots, k$

$$x = \alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_d u_d \text{ keep only } (\alpha_1, \ldots, \alpha_k)$$

  - Lurking instability: what if $\lambda_j = \lambda_{j+1}$?

More PCA

- **Multiple Dimensions** What if we want multiple dimensions? We keep the top-k.

$$\max_{U \in \mathbb{R}^{k \times d} : UU^T = I_k} \frac{1}{n} \sum_{u=1}^{n} \| U x^{(i)} \|^2 .$$

- **Reduce dimensionality**. How do we represent data with just those $k < d$ scalars α_j for $j = 1, \ldots, k$

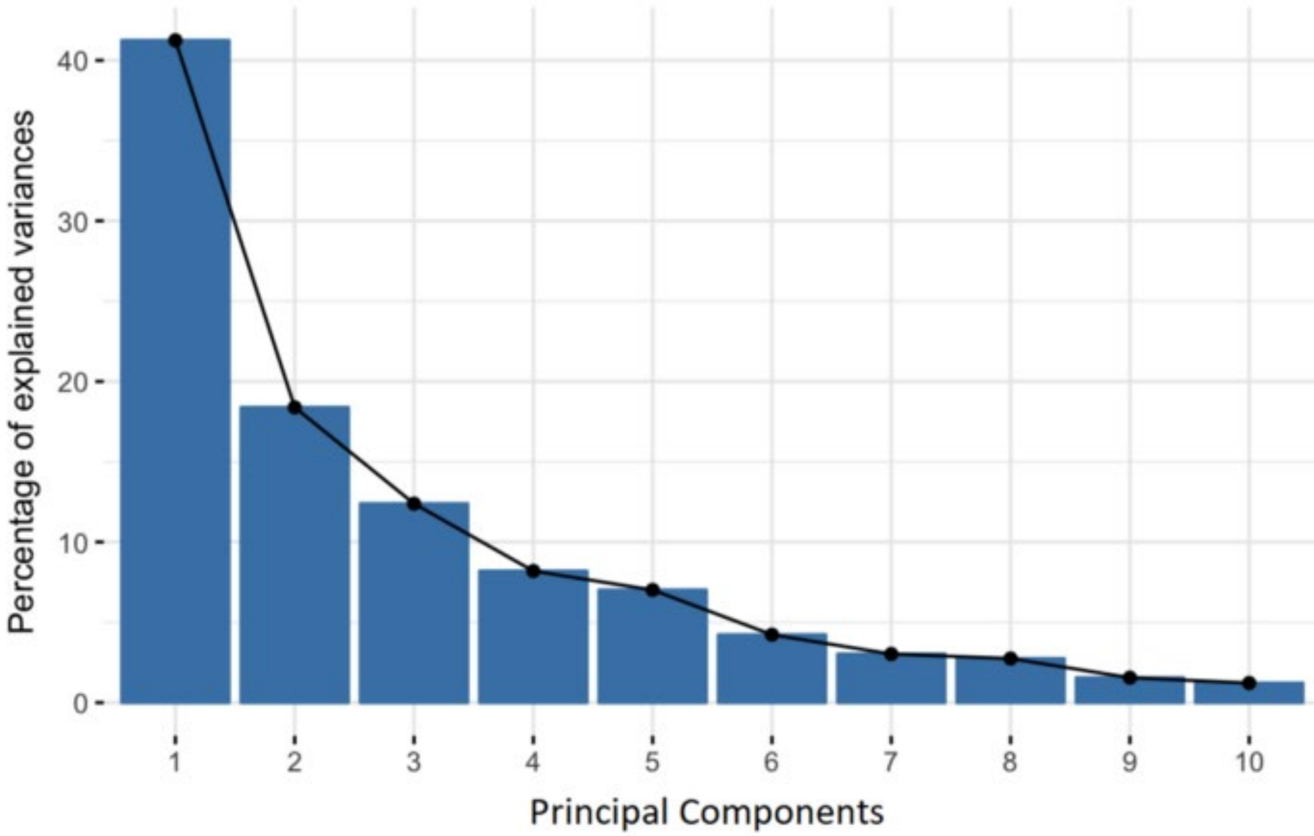
$$x = \alpha_1 u_1 + \alpha_2 u_2 + \cdots + \alpha_d u_d \text{ keep only } (\alpha_1, \ldots, \alpha_k)$$

  - Lurking instability: what if $\lambda_j = \lambda_{j+1}$?

- **Choose k?** One approach is "amount of explained variance"

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{i=1}^{n} \lambda_i} \geq 0.9 \text{ note } \mathrm{tr}(C) = \sum_{i=1}^{n} C_{i,i} = \sum_{i=1}^{n} \lambda_i$$

Recall $\lambda_j \geq 0$ since C is a covariance matrix.


Percentage of explained variances
0
10
20
30
40
1
2
3
4
5
6
7
8
9
10
Principal Components

Recap of PCA

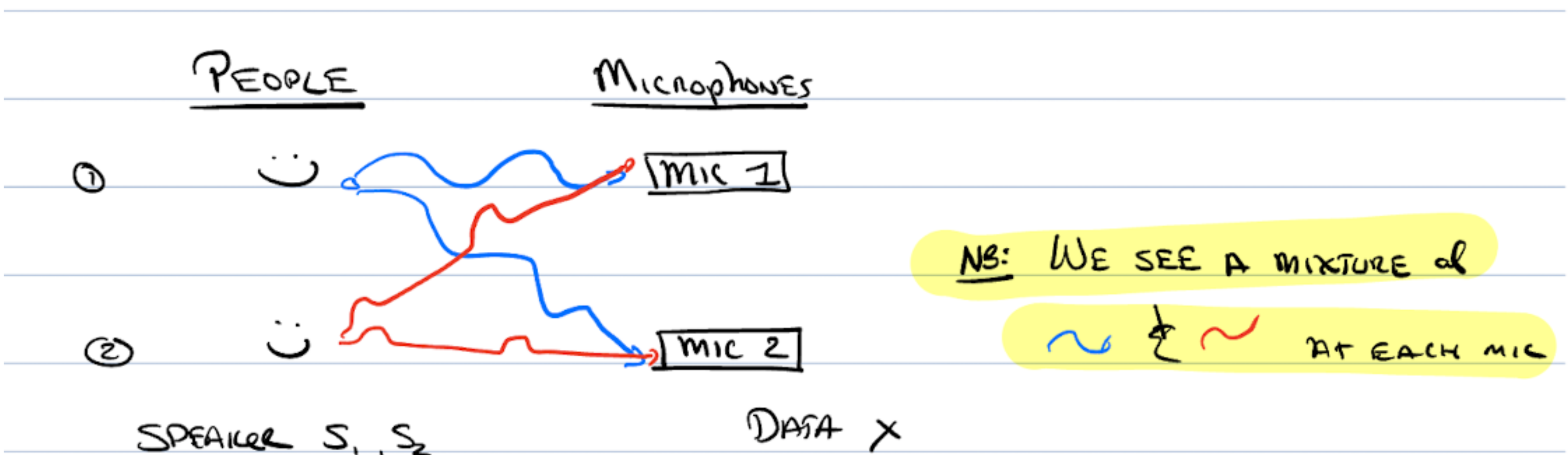
- Project the data onto a subspace: Find the subspace that captures as much of the data as possible (or doesn't explain the least amount).
- Dimensionality reduction and visualization
- Note: The preprocessing (especially centering) featured in our interpretation.

Independent Component Analysis

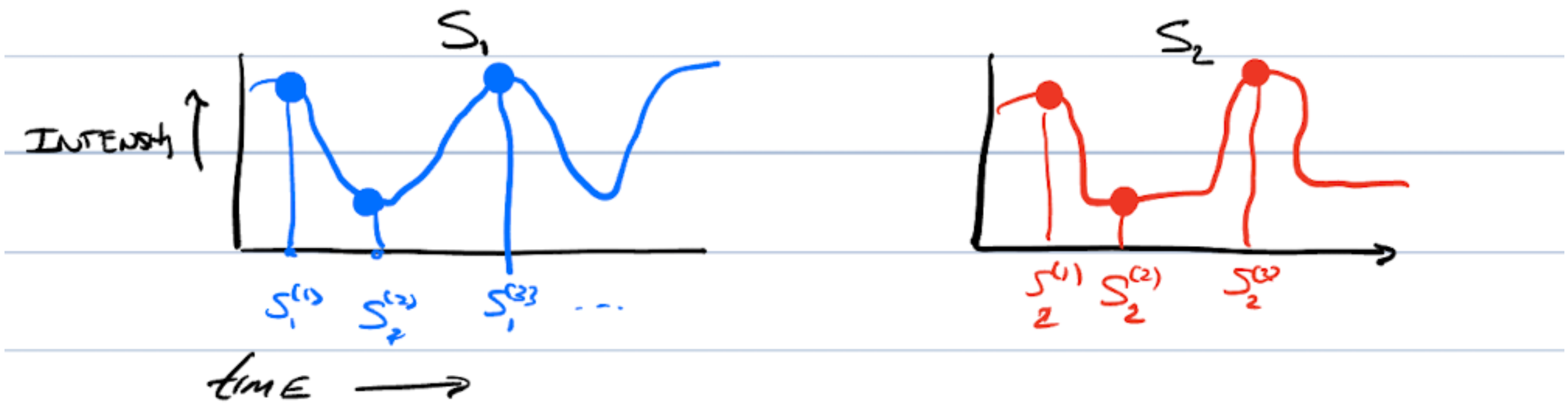
ICA: Independent Component Analysis

- The high-level story (the cocktail party problem)
- The key technical issues (on distributions) and likelihoods
- Model

Cocktail Party Problem



The Data



$S_j^{(t)}$ is the intensity at time t from speaker j.

We do **not** observe $S^{(t)}$ directly, only $x^{(t)}$ the microphones.

Our model is.

$$x_j^{(t)} = a_{j,1} S_1^{(t)} + a_{j,2} S_2^{(t)}.$$

"Microphone j at time t $\left(x_j^{(t)}\right)$ receives a mixture of speaker 1 at time t $\left(S_1^{(t)}\right)$ and speaker 2 at time t $\left(S_2^{(t)}\right)$."

Our Model

We can write out model succinctly as:

$$x^{(t)} = As^{(t)} \text{ for } t = 1, \ldots, n$$

- The blue values are observed: $x^{(t)}$.
- The red values are latent: A and $s^{(t)}$.
- Given x, our goal is to estimate s and A.

For simplicity, we assume number of speakers equals the number of microphones.

More formal model

- **Given:** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ where d is the number of speakers and microphones.
- **Do:** Find $s^{(1)}, \ldots, s^{(n)} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$

$$x^{(t)} = As^{(t)}.$$

We call A the **mixing matrix** and $W = A^{-1}$ is the unmixing matrix.

More formal model

- **Given:** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ where d is the number of speakers and microphones.
- **Do:** Find $s^{(1)}, \ldots, s^{(n)} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$

$$x^{(t)} = As^{(t)}.$$

We call A the **mixing matrix** and $W = A^{-1}$ is the unmixing matrix. We write

$$W = \begin{pmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_d^T \end{pmatrix} \text{ so that } S_j^{(t)} = w_j \cdot x^{(t)}.$$

More formal model

- **Given:** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ where d is the number of speakers and microphones.
- **Do:** Find $s^{(1)}, \ldots, s^{(n)} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$

$$x^{(t)} = As^{(t)}.$$

Some caveats:

- We assume A does **not** vary with time and is full rank.

More formal model

- **Given:** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ where d is the number of speakers and microphones.
- **Do:** Find $s^{(1)}, \ldots, s^{(n)} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$

$$x^{(t)} = As^{(t)}.$$

Some caveats:

- We assume A does **not** vary with time and is full rank.
- There are *inherent ambiguities*:
  - We can't determine speaker id (could swap 1 and 2!)

More formal model

- **Given:** $x^{(1)}, \dots, x^{(n)} \in \mathbb{R}^d$ where d is the number of speakers and microphones.
- **Do:** Find $s^{(1)}, \dots, s^{(n)} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$

$$x^{(t)} = As^{(t)}.$$

Some caveats:

- We assume A does **not** vary with time and is full rank.
- There are *inherent ambiguities*:
  - We can't determine speaker id (could swap 1 and 2!)
  - We can't determine absolute intensity:

$$(cA)(c^{-1}s^{(t)}) = As^{(t)} \text{ for any } c \neq 0.$$

More formal model

- **Given:** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ where d is the number of speakers and microphones.
- **Do:** Find $s^{(1)}, \ldots, s^{(n)} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$

$$x^{(t)} = As^{(t)}.$$

Some caveats:

- We assume A does **not** vary with time and is full rank.
- There are *inherent ambiguities*:
  - We can't determine speaker id (could swap 1 and 2!)
  - We can't determine absolute intensity:

$$(cA)(c^{-1}s^{(t)}) = As^{(t)} \text{ for any } c \neq 0.$$

- Speakers cannot be Gaussian! Maybe surprising:

  $x^{(t)} \sim \mathcal{N}(\mu, AA^T)$ then if $U^T U = I$ then AU generates same data.

Nevertheless, we can recover something meaningful–and the whole algorithm is just MLE with gradient descent.

More formal model

- **Given:** $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ where d is the number of speakers and microphones.
- **Do:** Find $s^{(1)}, \ldots, s^{(n)} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{d \times d}$

$$x^{(t)} = As^{(t)}.$$

Some caveats:

- We assume A does **not** vary with time and is full rank.
- There are *inherent ambiguities*:
  - We can't determine speaker id (could swap 1 and 2!)
  - We can't determine absolute intensity:

$$(cA)(c^{-1}s^{(t)}) = As^{(t)} \text{ for any } c \neq 0.$$

- Speakers cannot be Gaussian! Maybe surprising:

$$x^{(t)} \sim \mathcal{N}(\mu, AA^T) \text{ then if } U^T U = I \text{ then } AU \text{ generates same data.}$$

Nevertheless, we can recover something meaningful–and the whole algorithm is just MLE with gradient descent.We need one fact first.

Now the ICA Model is MLE

Goal: write signals in terms of observed quantities:

$$p(s) = \prod_{j=1}^{d} p_s(s_j) \qquad \text{sources are iid.}$$

Now the ICA Model is MLE

Goal: write signals in terms of observed quantities:

$$p(s) = \prod_{j=1}^{d} p_s(s_j) \qquad \text{sources are iid.}$$

$$p(x) = \prod_{j=1}^{d} p_s(w_j \cdot x) \left|\det(W)\right| \qquad \text{Use the previous slide}$$

Technical: Use non-rotationally invariant distribution. We set

$$p_s(x) \propto g'(x) \text{ for } g(x) = \frac{1}{1+e^{-x}}.$$

With this, we can solve the following with gradient descent:

$$\ell(W) = \sum_{t=1}^{n} \sum_{j=1}^{d} \log g'\left(w_j \cdot x^{(t)}\right) + \log\left|\det(W)\right|.$$

Summary of Lecture

- We saw PCA: workhorse of dimensionality reduction. The structure was "subspaces"
- We saw ICA: Key idea for homework, and introduced this concept of up to symmetry.

Recall: Eigenvalue decompositions

Given $x \in \mathbb{R}^d$ and $A = U\Lambda U^T$ we can express x in the basis:

$$x = \sum_{j=1}^{d} \alpha_j u_j$$

As before, using $u_i \cdot u_j = \delta_{i,j}$, we compute $x^T A x$

$$= x^T U\Lambda \sum_{j=1}^{d} \alpha_j e_j = x^T U \sum_{j=1}^{d} \lambda_j \alpha_j e_j = x^T \left(\sum_{j=1}^{d} \lambda_j \alpha_j u_j \right) = \sum_{j=1}^{d} \lambda_j \alpha_j^2$$

Since $\|x\|^2 = x^T x = \sum_{j=1}^{d} \alpha_j^2 = \|\alpha\|^2$, we can write:

$$\max_{x:\|x\|^2=1} x^T A x \text{ is equivalent to } \max_{\alpha:\|\alpha\|^2=1} \sum_{j=1}^{d} \alpha_j^2 \lambda_j.$$

Eigenvectors

So which x attains a maximum?

$$\max_{x:\|x\|^2=1} x^T A x \text{ is equivalent to } \max_{\alpha:\|\alpha\|^2=1} \sum_{j=1}^{d} \alpha_j^2 \lambda_j.$$

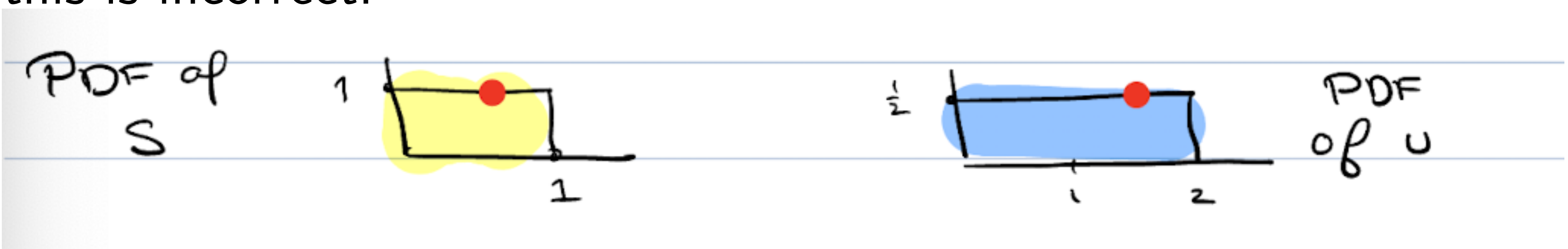
- Taking $x = u_1$ works, why?
- What if $\lambda_1 = \lambda_2$, is it unique?
  - Potential instability, when λ_1 is close to λ_2 issues can happen!

Detour: Density under linear transformations

Consider

$$s \sim \text{Uniform}[0,1] \text{ and } u = 2s.$$

What is the PDF of u? Tempted to write $P_u(x/2) = P_s(x)$ – but this is incorrect:



$$P_s(x) = \begin{cases} 1 & \text{if } x \in [0,1] \\ 0 & \text{otherwise} \end{cases} \text{ and } P_u(x) = \frac{1}{2} p_s\left(\frac{x}{2}\right).$$

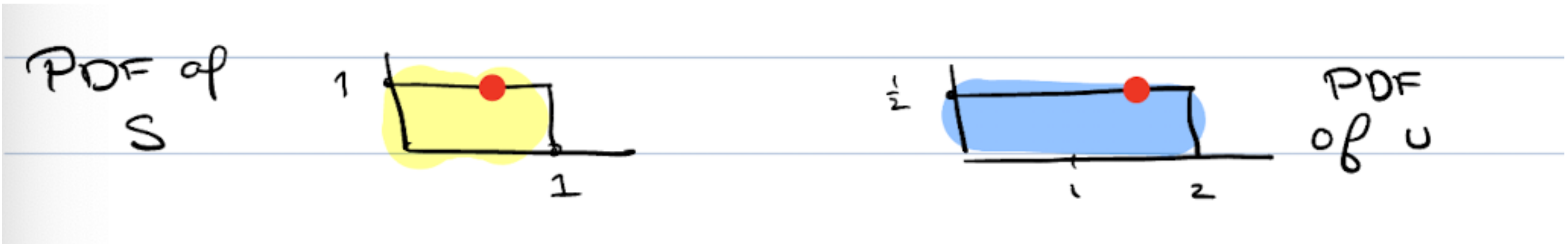
The key issue is the *normalization constant* here $\frac{1}{2}$.

Detour: Density under linear transformations

Consider

$$s \sim \text{Uniform}[0,1] \text{ and } u = 2s.$$

What is the PDF of u? Tempted to write $P_u(x/2) = P_s(x)$ – but this is incorrect:



$$P_s(x) = \begin{cases} 1 & \text{if } x \in [0,1] \\ 0 & \text{otherwise} \end{cases} \text{ and } P_u(x) = \frac{1}{2} p_s\left(\frac{x}{2}\right).$$

The key issue is the *normalization constant* here $\frac{1}{2}$. For matrix A:

$$P_u(x) = p_s(A^{-1}x)\left|\det(A^{-1})\right| = P_s(Wx)\left|\det(W)\right|.$$

Here, $\det(A^{-1}) = \frac{1}{\det(A)}$