DSC 140B
Representation Learning

Lecture 02 | Part 1

Logistics

http://zhiting.ucsd.edu/teaching/dsc140bwinter2024

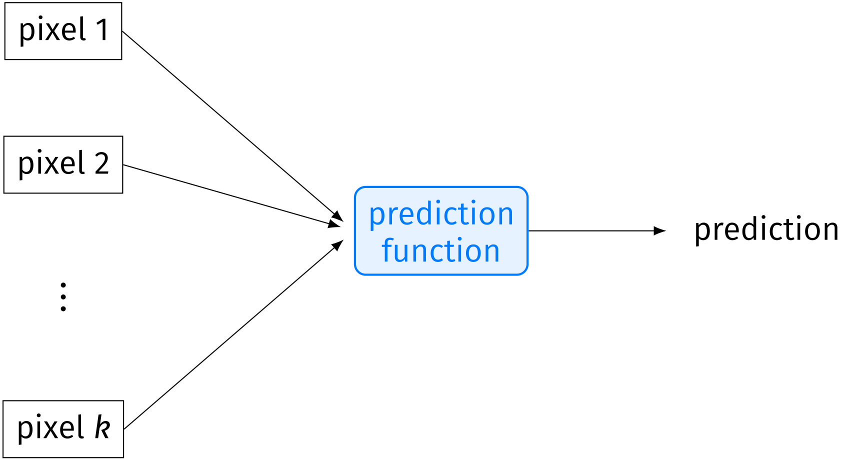
DSC 140B
Representation Learning

Lecture 02 | Part 2

Introduction (Cont'd)

Now: Predict Happiness

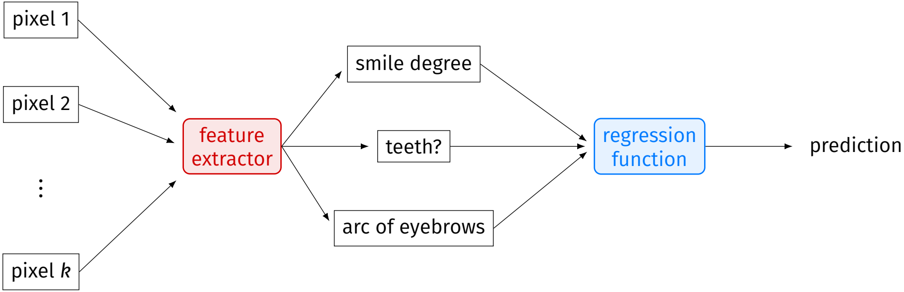
- Given an image, predict happiness on a 1-10 scale.
- This is a **regression** problem.
- Can we use least squares regression?


pixel 1
pixel 2
⋮
pixel k
prediction function
prediction

Handcrafted Representations

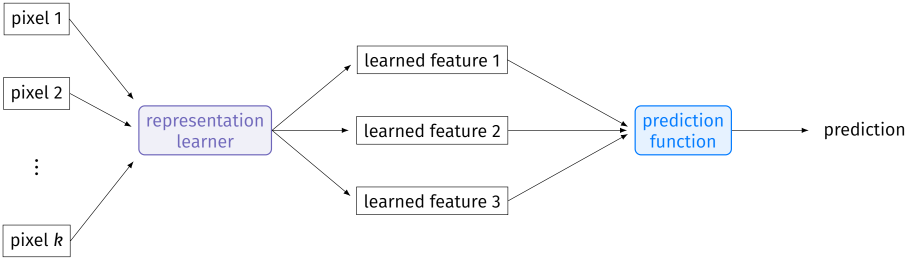
- Idea: build a **feature extractor** to detect:
  - The shape of the eyebrows.
  - Angle of the corners of the mouth.
  - Are teeth visible?

- Use these as high-level features instead.


pixel 1
pixel 2
⋮
pixel k
feature extractor
smile degree
teeth?
arc of eyebrows
regression function
prediction

Problem

- Extractors (may) make good **representations**.
- But building a feature extractor is **hard**.
- Can we **learn** a good representation?


pixel 1
pixel 2
⋮
pixel k
representation learner
learned feature 1
learned feature 2
learned feature 3
prediction function
prediction

DSC 140B

- We'll see how to **learn good representations**.

- Good representations help us when:
  1. making predictions;
  2. doing EDA (better visualizations).

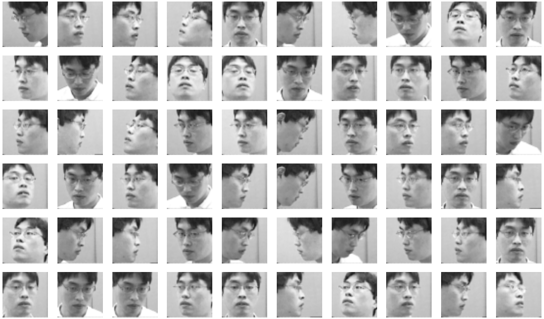
Claim

- Many of the famous recent advancements in AI/ML are due to **representation learning**.

Representations and Structure

- Real world data has structure.
- But “seeing” the structure requires the right representation.

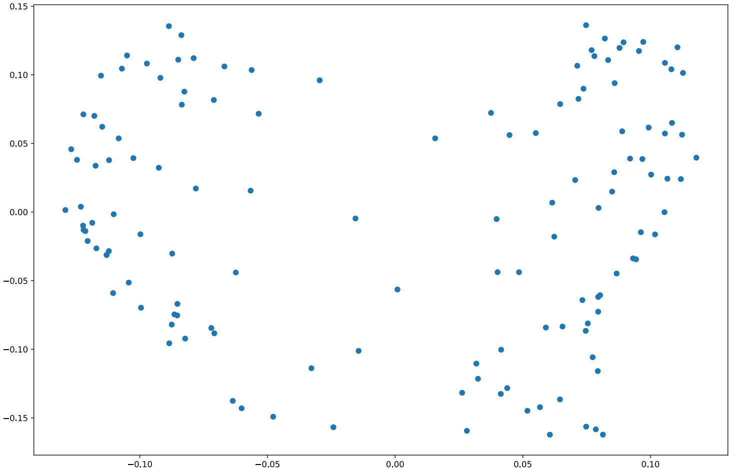
Example: Pose Estimation

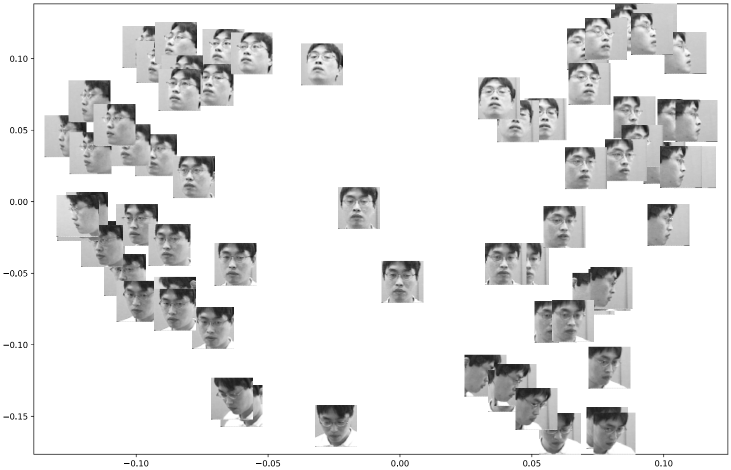


Problem: Classify, is person looking left, right, up, down, netural?

Example: Pose Estimation

- As a “bag of pixels” each image is a vector in $\mathbb{R}^{10,000}$.
- Later: we’ll see how to reduce dimensionality while preserving “closeness”.


0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.10
−0.05
0.00
0.05
0.10


0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.10
−0.05
0.00
0.05
0.10

Main Idea

By learning a better representation, the classification problem can become easy; sometimes trivial.

Example: `word2vec`

- How do we represent a word?
- Google's `word2vec` learned a representation of words as points in 300 dimensional space.
- Two points close $\iff$ words have similar meanings.

Example: `word2vec`

- Fun fact: we can now add and subtract words.
  - They're represented as vectors.

- Surprising results:

$$\vec{v}_{\text{Paris}} - \vec{v}_{\text{France}} + \vec{v}_{\text{China}} \approx \vec{v}_{\text{Beijing}}$$

Example: `word2vec` [2]

Table 8: *Examples of the word pair relationships, using the best word vectors from Table 4 (Skip-gram model trained on 783M words with 300 dimensionality).*

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

[2] "Efficient Estimation of Word Representations in Vector Space" by Mikolov, et al.

Example: Neural Networks

- `word2vec` is an example of a neural network model.
- Deep neural networks have been very successful on certain tasks.
- They **learn** a good representation.

Main Idea

Building a good model requires picking a good **feature representation**.

We can pick features by hand.

Or we can **learn** a good feature representation from data.

DSC 140B is about learning these representations.

Roadmap

- Dimensionality Reduction
- Manifold learning
- Neural Networks
- Autoencoders
- Deep Learning

Practice vs. Theory

- Goal of this class: understand the fundamentals of representation learning.
- Both practical and theoretical.
- Think: more DSC 40A than DSC 80, but a bit of both.

Tools of the Trade

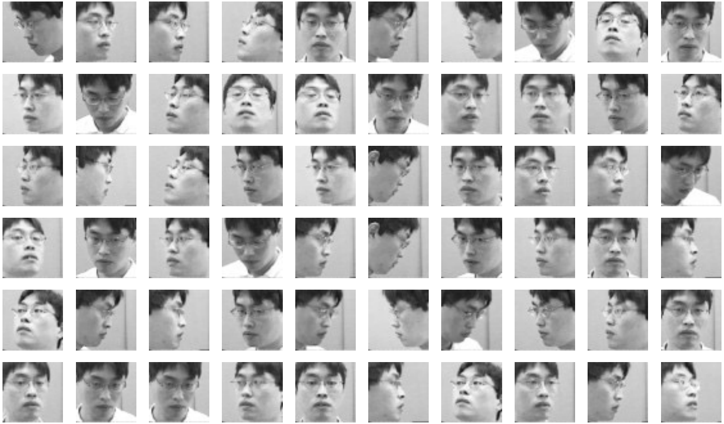
- We’ll see some of the popular Python tools for feature learning.
  - `numpy`
  - `keras`
  - `sklearn`
  - ...

DSC 140B
Representation Learning

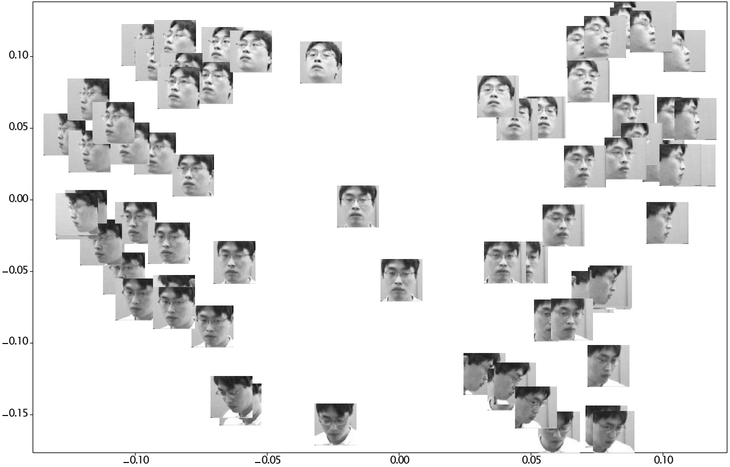
Lecture 02 | Part 3

Why Linear Algebra?

Recall



Recall



Dimensionality Reduction

- This is an example of **dimensionality reduction**:
  - Input: vectors in $\mathbb{R}^{10{,}000}$.
  - Output: vectors in $\mathbb{R}^2$.

- The method which produced this result is called **Laplacian Eigenmaps**.

- How does it work?

A Preview of Laplacian Eigenmaps

To reduce dimensionality from d to d':

1. Create an undirected **similarity graph** G
   - Each vector in $\mathbb{R}^d$ becomes a node in the graph.
   - Make edge (u, v) if u and v are "close"
2. Form the **graph Laplacian matrix**, L:
   - Let A be the adjacency matrix, D be the degree matrix.
   - Define the graph Laplacian matrix, $L = D - A$.
3. Compute d' **eigenvectors** of L.
   - Each eigenvector gives one new feature.

Why eigenvectors?

- We will cover Laplacian Eigenmaps in much greater detail.

- For now: why do eigenvectors appear here?
  - What are eigenvectors?
  - How are they useful?
  - Why is linear algebra important in ML?

DSC 140B

Representation Learning

Lecture 02 | Part 4

Coordinate Vectors

Coordinate Vectors

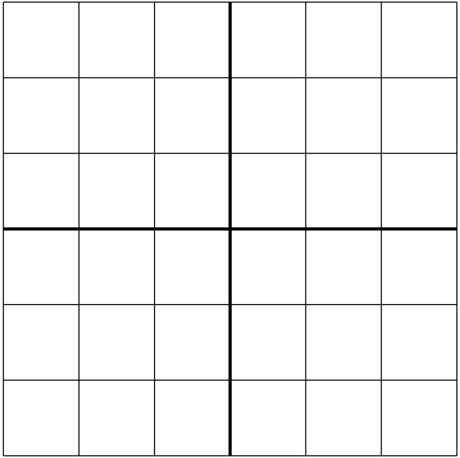
- We can write a vector $\vec{x} \in \mathbb{R}^d$ as a **coordinate vector**:

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$$

Example

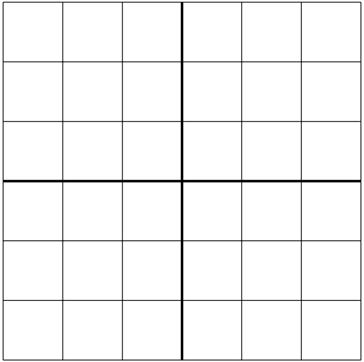
$$\vec{x} = \begin{pmatrix} 2 \\ -3 \end{pmatrix}$$

$$\vec{y} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$



Standard Basis

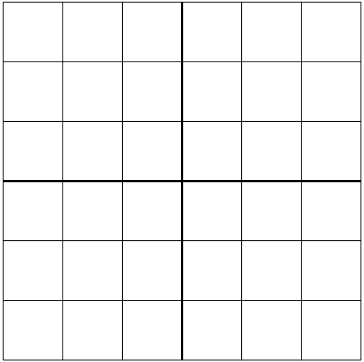
- Writing a vector in coordinate form requires choosing a **basis**.
- The "default" is the **standard basis**: $\hat{e}^{(1)}, \ldots, \hat{e}^{(d)}$.



Standard Basis

- When we write $\vec{x} = (x_1, \ldots, x_d)^T$, we mean that $\vec{x} = x_1 \hat{e}^{(1)} + x_2 \hat{e}^{(2)} + \ldots x_d \hat{e}^{(d)}$.

Example: $\vec{x} = (3, -2)^T$



Standard Basis Coordinates

- In coordinate form:

$$\hat{e}^{(i)} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

where the 1 appears in the ith place.

Exercise

Let $\vec{x} = (3, 7, 2, -5)^T$. What is $\vec{x} \cdot \hat{e}^{(4)}$?

Recall: the Dot Product

- The **dot product** of $\vec{u}$ and $\vec{v}$ is defined as:

  $$\vec{u} \cdot \vec{v} = \|\vec{u}\| \|\vec{v}\| \cos\theta$$

  where θ is the angle between $\vec{u}$ and $\vec{v}$.

- $\vec{u} \cdot \vec{v} = 0$ if and only if $\vec{u}$ and $\vec{v}$ are orthogonal

Dot Product (Coordinate Form)

- In terms of coordinate vectors:

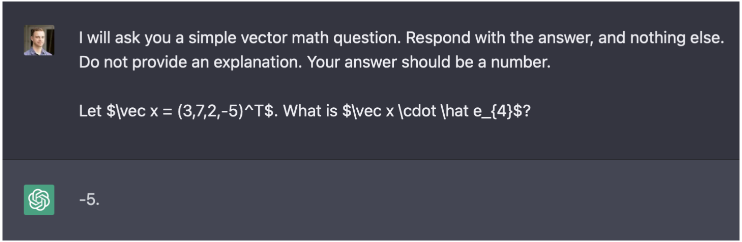
$$\begin{aligned}\vec{u} \cdot \vec{v} &= \vec{u}^T \vec{v} \\ &= \begin{pmatrix} u_1 & u_2 & \cdots & u_d \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \cdots \\ v_d \end{pmatrix} \\ &= \end{aligned}$$

- This definition assumes the standard basis.

Example

$$\begin{pmatrix} 3 \\ 7 \\ 2 \\ -5 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} =$$

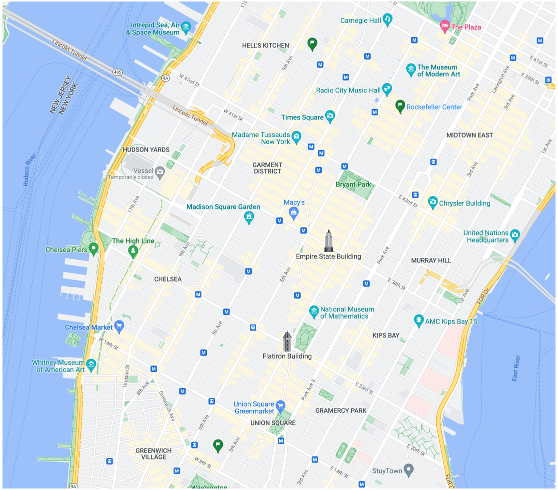
What does ChatGPT say?



Other Bases

- The standard basis is not the **only** basis.
- Sometimes more convenient to use another.

Example



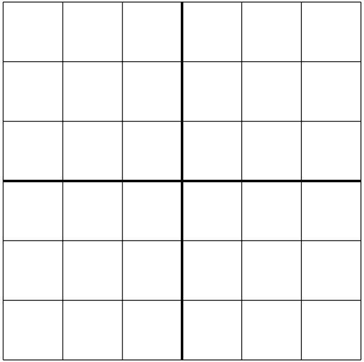
Intrepid Sea, Air & Space Museum
Carnegie Hall
The Plaza
HELL'S KITCHEN
The Museum of Modern Art
Radio City Music Hall
Rockefeller Center
Times Square
Madame Tussauds New York
MIDTOWN EAST
HUDSON YARDS
GARMENT DISTRICT
Vessel
Temporarily closed
Bryant Park
Chrysler Building
Madison Square Garden
Macy's
United Nations Headquarters
The High Line
Chelsea Piers
Empire State Building
MURRAY HILL
CHELSEA
National Museum of Mathematics
AMC Kips Bay 15
Chelsea Market
KIPS BAY
Flatiron Building
Whitney Museum of American Art
Union Square Greenmarket
GRAMERCY PARK
UNION SQUARE
GREENWICH VILLAGE
StuyTown
NEW JERSEY
NEW YORK
Hudson River
East River
Lincoln Tunnel

Orthonormal Bases

- **Orthonormal bases** are particularly nice.
- A set of vectors $\hat{u}^{(1)}, \ldots, \hat{u}^{(d)}$ forms an **orthonomal basis** $\mathcal{U}$ for R^d if:
  - They are mutually orthogonal: $\hat{u}^{(i)} \cdot \hat{u}^{(j)} = 0$.
  - They are all unit vectors: $\|\hat{u}^{(i)}\| = 1$.

Example

$$\hat{u}^{(1)} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \qquad \hat{u}^{(2)} = \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\end{pmatrix}$$



Coordinate Vectors

- A vector's coordinates depend on the basis used.
- If we are using the basis $\mathcal{U} = \{\hat{u}^{(1)}, \hat{u}^{(2)}\}$, then $\vec{x} = (x_1, x_2)^T$ means $\vec{x} = x_1 \hat{u}^{(1)} + x_2 \hat{u}^{(2)}$.
- We will write $[\vec{x}]_{\mathcal{U}} = (x_1, \ldots, x_d)^T$ to denote that the coordinates are with respect to the basis $\mathcal{U}$.

Exercise

Let $\hat{u}^{(1)} = \frac{1}{\sqrt{2}}(1,1)^T$ and $\hat{u}^{(2)} = \frac{1}{\sqrt{2}}(-1,1)^T$. Suppose $[\vec{x}]_{\mathcal{U}} = (3,-4)^T$. What is $\vec{x} \cdot \hat{u}^{(1)}$?

What did ChatGPT say?

Let $\bu{1} = \frac{1}{\sqrt2}(1,1)^T$ and $\bu{2} = \frac{1}{\sqrt2}(-1,1)^T$. Suppose $[\vec x]_\mathcal{U} = (3, -4)^T$. What is $\vec x \cdot \bu{1}$?

-0.5. (Note: This answer assumes that $[\vec x]_\mathcal{U}$ represents the coordinate vector of $\vec x$ with respect to the basis $\mathcal{U}$, and that $\bu{1}$ and $\bu{2}$ are orthonormal basis vectors.) The dot product between $\vec x$ and $\bu{1}$ represents the projection of $\vec x$ onto the direction of $\bu{1}$. In this case, $\bu{1}$ is a unit vector, so the dot product gives the signed length of the projection of $\vec x$ onto the direction of $\bu{1}$. The result of -0.5 indicates that the projection of $\vec x$ onto the direction of $\bu{1}$ is in the opposite direction of $\bu{1}$, and the magnitude of the projection is 0.5 times the length of $\bu{1}$. Note that the negative sign indicates that the projection is in the opposite direction of $\bu{1}$. The dot product can be used to determine the alignment or misalignment of vectors, and is a useful tool in various applications, such as in physics, engineering, and computer graphics. It is important to understand how to compute and interpret dot products in the context of vector projections and geometric relationships between vectors. Therefore, the result of -0.5 indicates that $\vec x$ is projected onto the opposite direction of $\bu{1}$ with a magnitude of 0.5 times the length of $\bu{1}$. Keep practicing vector math problems to further strengthen your understanding of dot products and their applications!

Exercise

Consider $\vec{x} = (2, 2)^T$ and let $\hat{u}^{(1)} = \frac{1}{\sqrt{2}}(1, 1)^T$ and $\hat{u}^{(2)} = \frac{1}{\sqrt{2}}(-1, 1)^T$. What is $[\vec{x}]_{\mathcal{U}}$?

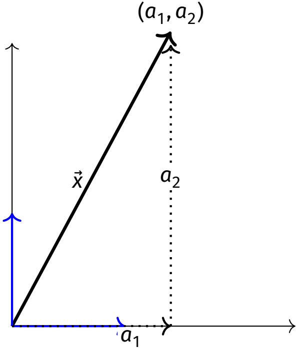
Change of Basis

- How do we compute the coordinates of a vector in a new basis, $\mathcal{U}$?

- Some trigonometry is involved.

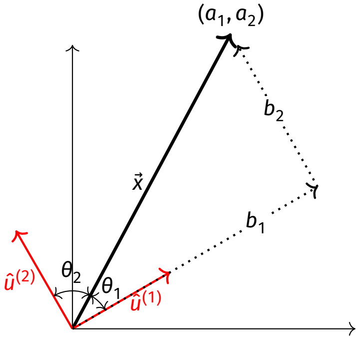
- **Key Fact**: $\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos \theta$

Change of Basis



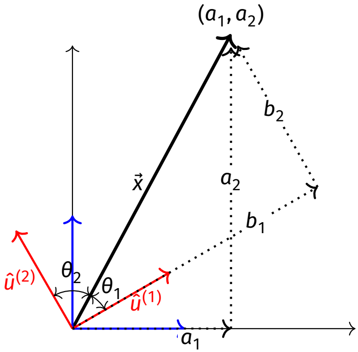
- Suppose we know $\vec{x} = (a_1, a_2)^T$ w.r.t. standard basis.
- Then $\vec{x} = a_1 \hat{e}^{(1)} + a_2 \hat{e}^{(2)}$

Change of Basis



- Want to write:
  $\vec{x} = b_1\hat{u}^{(1)} + b_2\hat{u}^{(2)}$
- Need to find b_1 and b_2.

Change of Basis



- **Exercise:** Solve for b_1, writing the answer as a dot product.
- Hint: $\cos \theta$ = adjacent/hypotenuse

Change of Basis

- Let $\mathcal{U} = \{\hat{u}^{(1)}, \ldots, \hat{u}^{(d)}\}$ be an orthonormal basis.
- The coordinates of $\vec{x}$ w.r.t. $\mathcal{U}$ are:

$$[\vec{x}]_{\mathcal{U}} = \begin{pmatrix} \vec{x} \cdot \hat{u}^{(1)} \\ \vec{x} \cdot \hat{u}^{(2)} \\ \vdots \\ \vec{x} \cdot \hat{u}^{(d)} \end{pmatrix}$$

Exercise

Suppose $\vec{x} = (2, 1)^T$ and let $\hat{u}^{(1)} = \frac{1}{\sqrt{2}}(1, 1)^T$ and $\hat{u}^{(2)} = \frac{1}{\sqrt{2}}(-1, 1)^T$. What is $[\vec{x}]_{\mathcal{U}}$?

Exercise

Let $\vec{x} = (-1, 4)^T$ and suppose:

$$\hat{u}^{(1)} \cdot \hat{e}^{(1)} = 3 \qquad \hat{u}^{(2)} \cdot \hat{e}^{(1)} = -1$$

$$\hat{u}^{(1)} \cdot \hat{e}^{(2)} = -2 \qquad \hat{u}^{(2)} \cdot \hat{e}^{(2)} = 5$$

What is $[\vec{x}]_{\mathcal{U}}$?