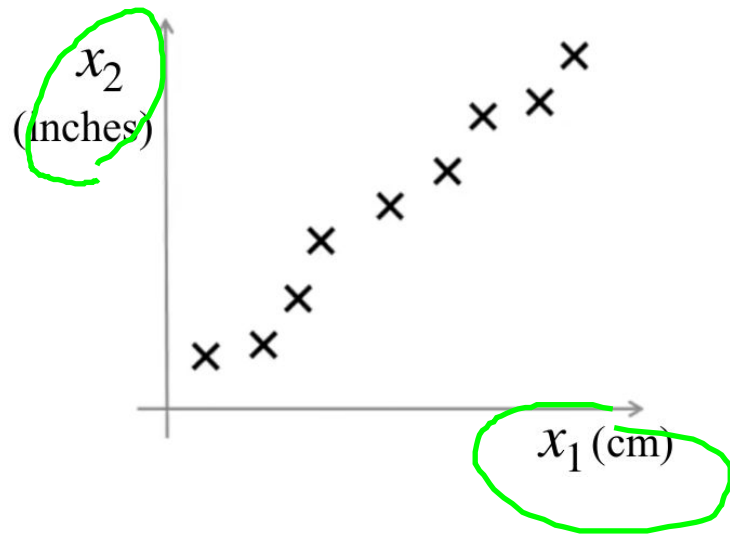


Dimensionality Reduction และ Unsupervised Learning Algorithms อื่น

Krittameth Teachasrisaksakul

แรงจูงใจที่ 1: Data Compression (การบีบอัดข้อมูล)



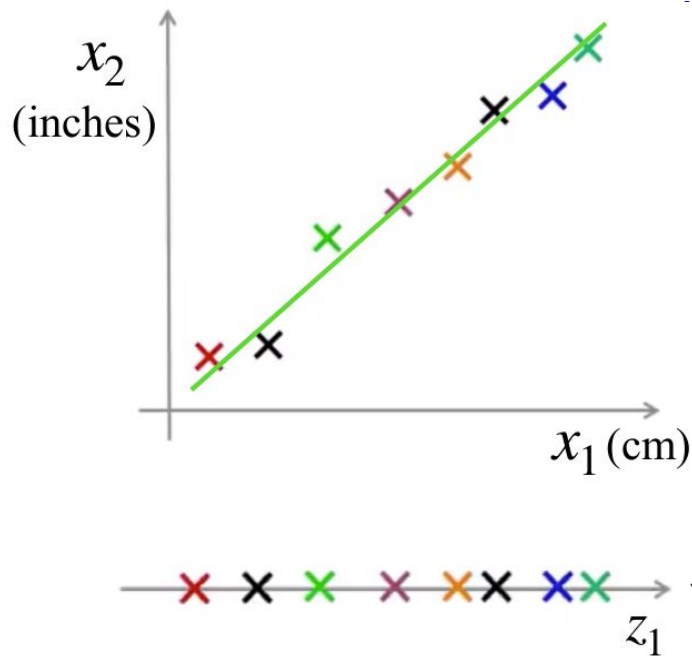
สมมติเราเก็บชุดข้อมูลที่มีหลาย features แต่เรา plot แค่ 2 features

และสมมติว่า เราไม่รู้ว่ามันเป็น ความยาว ในหน่วย cm (เซนติเมตร) และ inches (นิ้ว)

features 2 ตัวนี้ redundant และเราสามารถลดมิติข้อมูลจาก 2D เป็น 1D

คำถาม: reducing dimensions (การลดมิติ) หมายความว่าอะไร?

แรงจูงใจที่ 1: Data Compression (การบีบอัดข้อมูล)

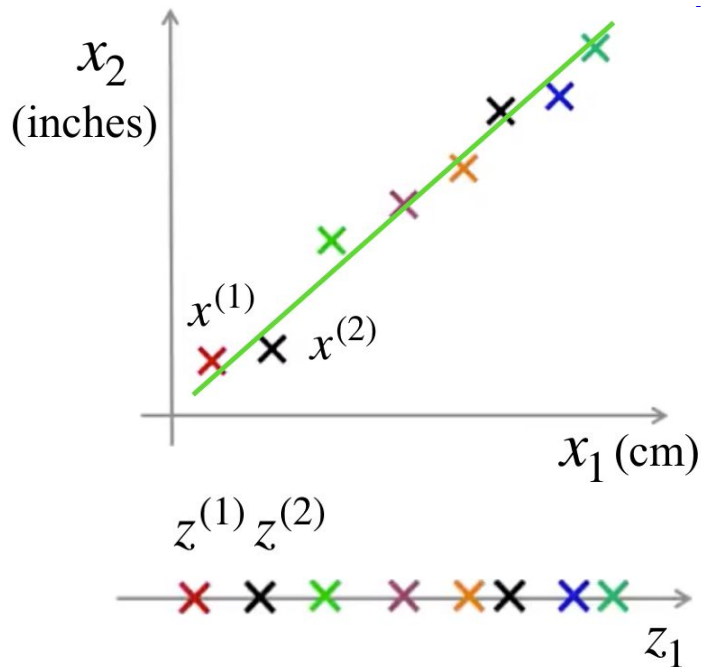


คำถาม: reducing dimensions (การลดมิติ) หมายความว่าอะไร?

หาเส้นที่เหมาะสม (a fitted line) และ project (ฉายภาพ) จุดข้อมูลลงบนอีกแกนหนึ่ง แล้ววัดค่า ตำแหน่งของ example แต่ละตัว บนเส้นนั้น

z_1 เป็น feature ใหม่ ที่บอกตำแหน่งของแต่ละจุดบนเส้นสีเขียว

แรงจูงใจที่ 1: Data Compression (การบีบอัดข้อมูล)



คำถาม: reducing dimensions (การลดมิติ) หมายความว่าอะไร?

$$x^{(1)} \in \mathbb{R}^2 \mapsto z^{(1)} \in \mathbb{R}$$

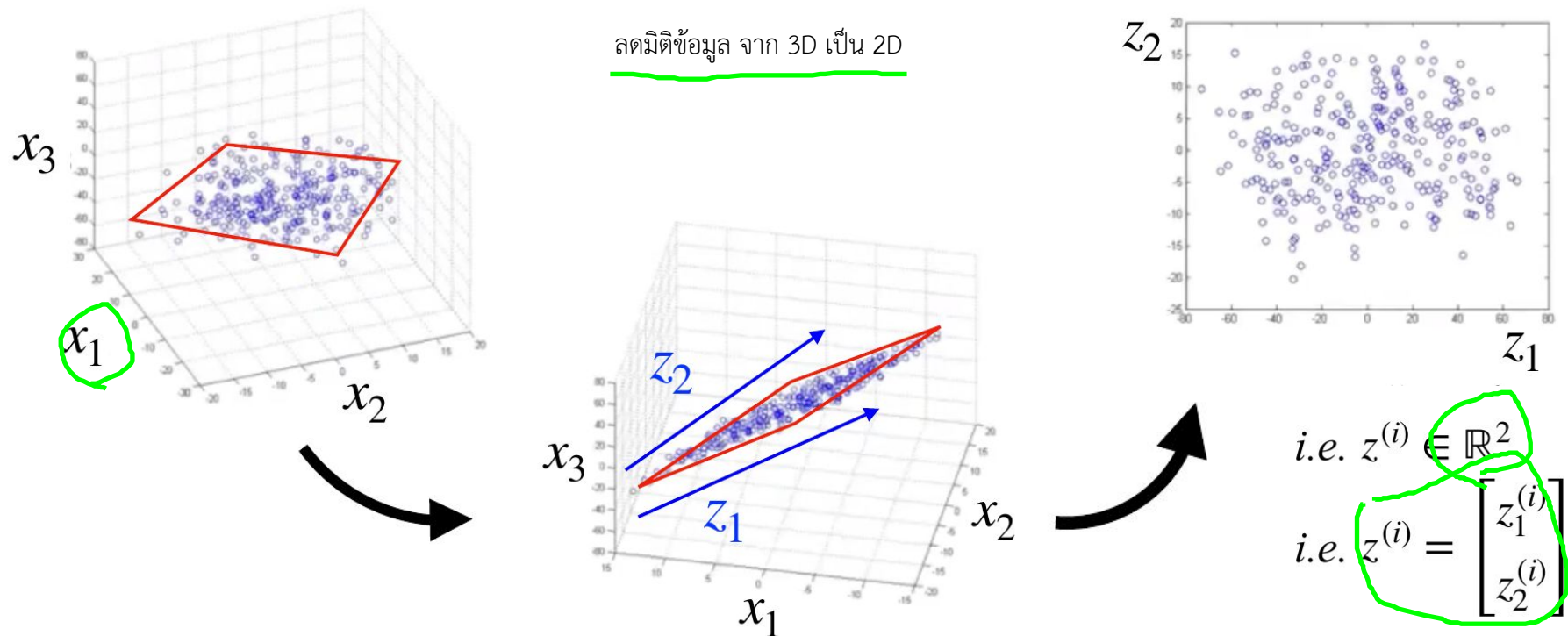
$$x^{(2)} \in \mathbb{R}^2 \mapsto z^{(2)} \in \mathbb{R}$$

\vdots

$$x^{(m)} \in \mathbb{R}^2 \mapsto z^{(m)} \in \mathbb{R}$$

ก็คือ ถ้าเราประ: $x^{(m)} \in \mathbb{R}^2 \mapsto z^{(m)} \in \mathbb{R}$ ฉายภาพ / ฉาย
เป็นเงา example เดิมลงบนเส้นสีเขียว แล้วเราต้องการแค่จำนวนจริงตัว
เดียว เพื่อระบุจุดนั้นบนเส้น

Data Compression : การบีบอัดข้อมูล



Question

สมมติเราใช้ dimensionality reduction กับชุดข้อมูลที่มี examples m ตัว : $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ เมื่อ $x^{(i)} \in \mathbb{R}^n$ เราจะได้ผลเป็นอะไร

- (i) ชุดข้อมูลที่มีมิติต่ำลง $\{z^{(1)}, z^{(2)}, \dots, z^{(k)}\}$ ที่มี k examples เมื่อ $k \leq n$
- (ii) ชุดข้อมูลที่มีมิติต่ำลง $\{z^{(1)}, z^{(2)}, \dots, z^{(k)}\}$ ที่มี k examples เมื่อ $k > n$
- (iii) ชุดข้อมูลที่มีมิติต่ำลง $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ ที่มี m examples เมื่อ $z^{(i)} \in \mathbb{R}^k$ สำหรับ k บางค่า และ $k \leq n$
- (iv) ชุดข้อมูลที่มีมิติต่ำลง $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ ที่มี m examples เมื่อ $z^{(i)} \in \mathbb{R}^k$ สำหรับ k บางค่า และ $k > n$

Question

สมมติเราใช้ dimensionality reduction กับชุดข้อมูลที่มี examples m ตัว : $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ เมื่อ $x^{(i)} \in \mathbb{R}^n$ เราจะได้ผลเป็นอะไร

- (i) ชุดข้อมูลที่มีมิติต่ำลง $\{z^{(1)}, z^{(2)}, \dots, z^{(k)}\}$ ที่มี k examples เมื่อ $k \leq n$
- (ii) ชุดข้อมูลที่มีมิติต่ำลง $\{z^{(1)}, z^{(2)}, \dots, z^{(k)}\}$ ที่มี k examples เมื่อ $k > n$
- (iii) ชุดข้อมูลที่มีมิติต่ำลง $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ ที่มี m examples เมื่อ $z^{(i)} \in \mathbb{R}^k$
สำหรับ k บางค่า และ $k \leq n$
- (iv) ชุดข้อมูลที่มีมิติต่ำลง $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ ที่มี m examples เมื่อ $z^{(i)} \in \mathbb{R}^k$
สำหรับ k บางค่า และ $k > n$

แรงจูงใจที่ 2: Visualization (การนำเสนอด้วยภาพ)

	x_1	x_2	x_3	x_4	x_5	x_6	
Country	GDP (trillions of US\$)	Per capita GDP (thousands of intl. \$)	Human Develop- ment Index	Life expectancy	Poverty Index (Gini as percentage)	Mean household income (thousands of US\$)	...
Canada	1.577	39.17	0.908	80.7	32.6	67.293	...
China	5.878	7.54	0.687	73	46.9	10.22	...
India	1.632	3.41	0.547	64.7	36.8	0.735	...
Russia	1.48	19.84	0.755	65.5	39.9	0.72	...
Singapore	0.223	56.69	0.866	80	42.5	67.1	...
USA	14.527	46.86	0.91	78.3	40.8	84.3	...
...

เช่น $\mathbf{x}^{(i)} \in \mathbb{R}^{50}$

(source: wikipedia.org)

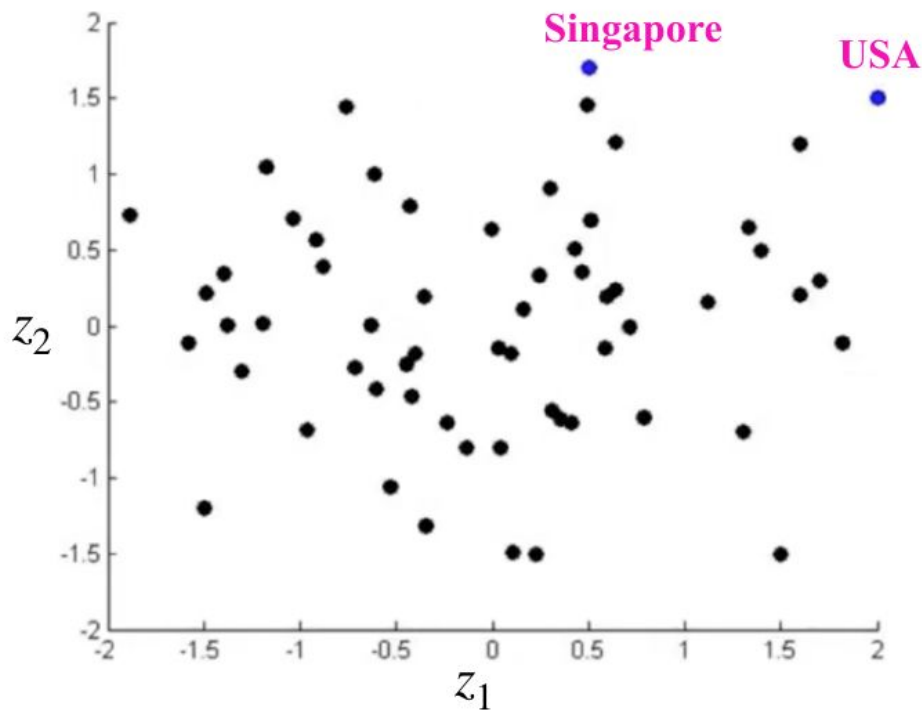
แรงจูงใจที่ 2: Visualization (การนำเสนอด้วยภาพ)

Country	z_1	z_2
Canada	1.6	1.2
China	1.7	0.3
India	1.6	0.2
Russia	1.4	0.5
Singapore	0.5	1.7
USA	2	1.5
...

เช่น $z^{(i)} \in \mathbb{R}^2$

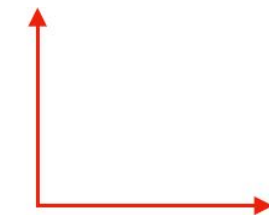
ก็คือ ลดมิติข้อมูลจาก 50D เป็น 2D

แรงจูงใจที่ 2: Visualization (การนำเสนอด้วยภาพ)



แทนประเทศทุกประเทศ ด้วยจุด 1 จุด $z^{(i)} \in \mathbb{R}^2$

GDP ต่อคน



Overall economic
size of a country

(ขนาดเศรษฐกิจของ
ประเทศ)

Question

สมมติ มีชุดข้อมูล $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ เมื่อ $x^{(i)} \in \mathbb{R}^n$ เพื่อจะ visualize ข้อมูล เราใช้ dimensionality reduction (การลดมิติข้อมูล) และได้ $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ เมื่อ $z^{(i)} \in \mathbb{R}^k$ มี k มิติ ในสภาพแวดล้อม (การตั้งค่า) ทั่วไป ข้อใดต่อไปนี่ที่เราคาดว่าเป็นจริง วงทุกข้อที่ถูกต้อง

(i) $k > n$

(ii) $k \leq n$

(iii) $k \geq 4$

(iv) $k = 2$ หรือ $k = 3$

(เพราะเราสามารถ plot ข้อมูล 2D หรือ 3D แต่ไม่มีวิธี visualize ข้อมูลที่มีมิติสูงกว่า)

Question

สมมติ มีชุดข้อมูล $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ เมื่อ $x^{(i)} \in \mathbb{R}^n$ เพื่อจะ visualize ข้อมูล เราใช้ dimensionality reduction (การลดมิติข้อมูล) และได้ $\{z^{(1)}, z^{(2)}, \dots, z^{(m)}\}$ เมื่อ $z^{(i)} \in \mathbb{R}^k$ มี k มิติ ในสภาพแวดล้อม (การตั้งค่า) ทั่วไป ข้อใดต่อไปนี่ที่เราคาดว่าเป็นจริง วงทุกข้อที่ถูกต้อง

(i) $k > n$

(ii) $k \leq n$

(iii) $k \geq 4$

(iv) $k = 2$ หรือ $k = 3$

(เพราะเราสามารถ plot ข้อมูล 2D หรือ 3D แต่ไม่มีวิธี visualize ข้อมูลที่มีมิติสูงกว่า)

Dimensionality Reduction

Principal Component Analysis (PCA)

Krittameth Teachasrisaksakul

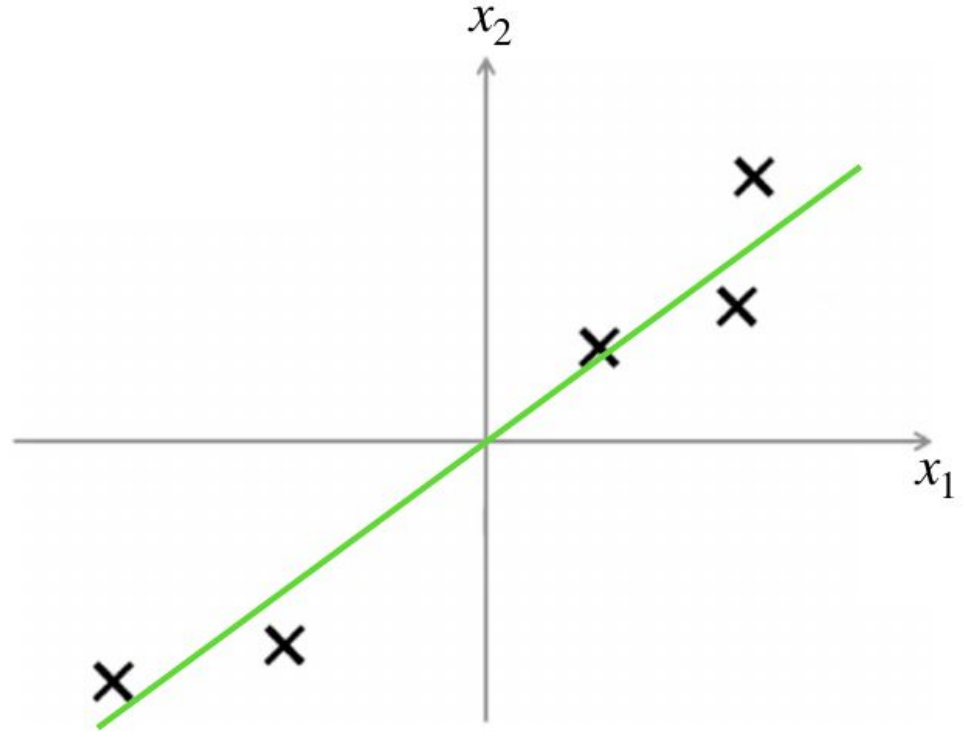
PCA: Problem Formulation (การเขียนสูตรของปัญหา)

$$\mathbf{x}_i \in \mathbb{R}^2$$

อยากลดมิติข้อมูลจาก 2D เป็น 1D

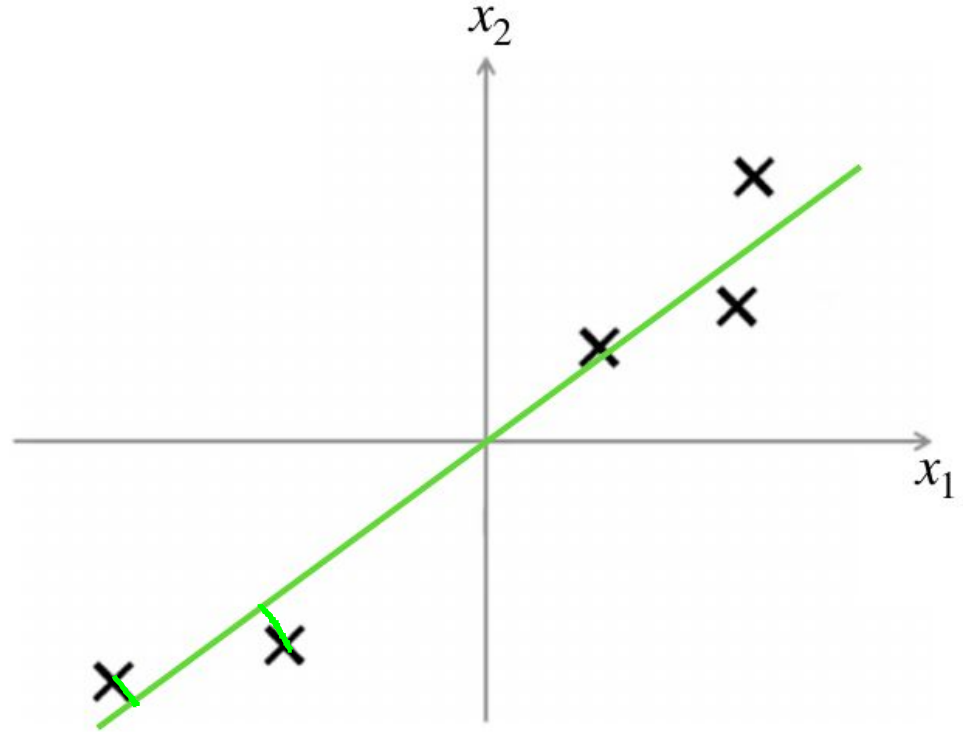
ก็คือ หาเส้นที่ดีที่สุดที่สามารถ project

(ฉายภาพ) ข้อมูลลงไปได้



PCA: Problem Formulation (การเขียนสูตรของปัญหา)

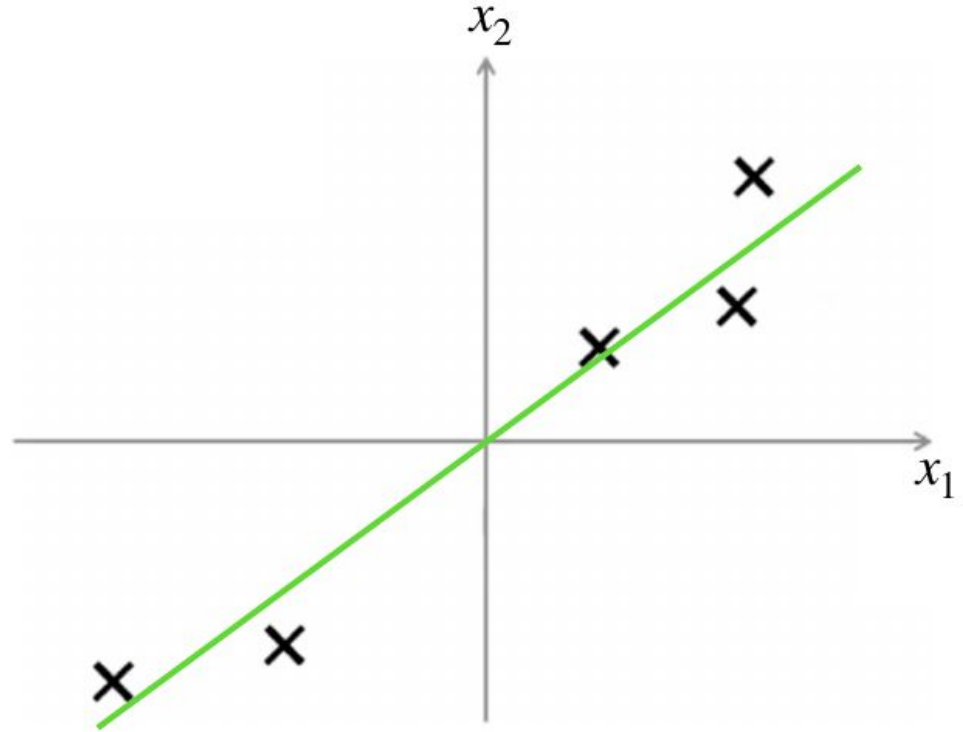
ความเข้าใจพื้นฐาน: หาพื้นผิว (surface) ที่มีมิติต่ำกว่า เช่น เส้นสีเขียว
ที่สามารถ project (ฉายภาพ) ข้อมูลลงไปได้ และ ต้องทำให้ ผลรวมของ
กำลังสอง (sum of squares) ของเส้นสีแดง น้อยที่สุด



PCA: Problem Formulation (การเขียนสูตรของปัญหา)

ความเข้าใจพื้นฐาน: หาพื้นผิว (surface) ที่มีมิติต่ำกว่า เช่น เส้นสีเขียว
ที่สามารถ project (ฉายภาพ) ข้อมูลลงไปได้ และ ต้องทำให้ ผลรวมของ
กำลังสอง (sum of squares) ของเส้นสีแดง น้อยที่สุด

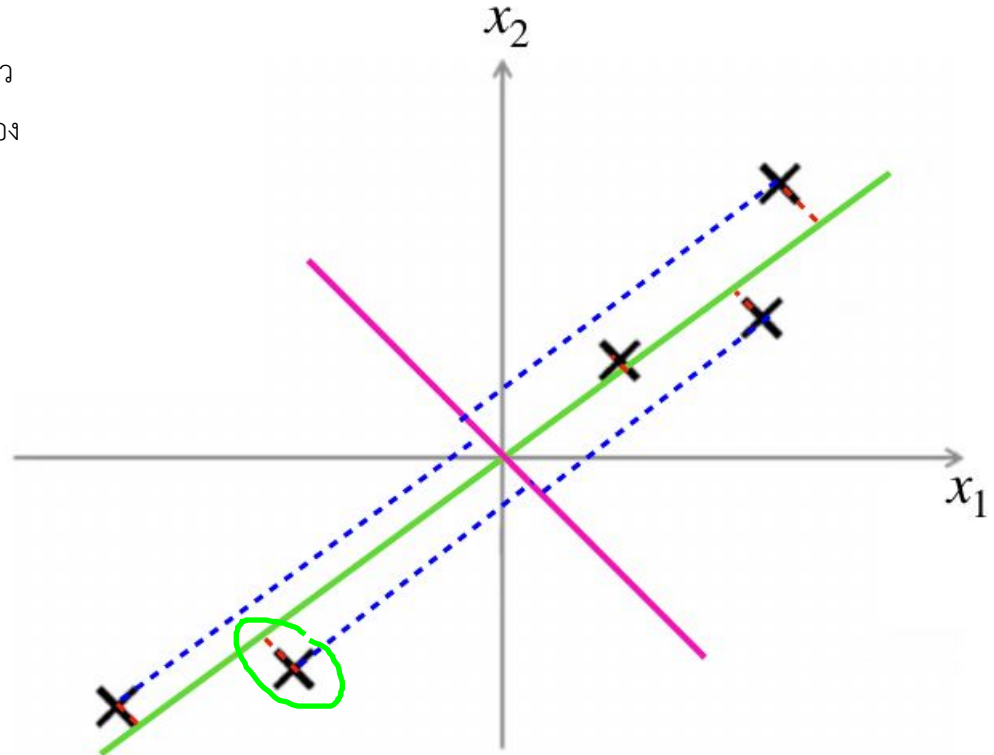
หมายเหตุ: ทำ feature scaling ก่อนใช้ PCA



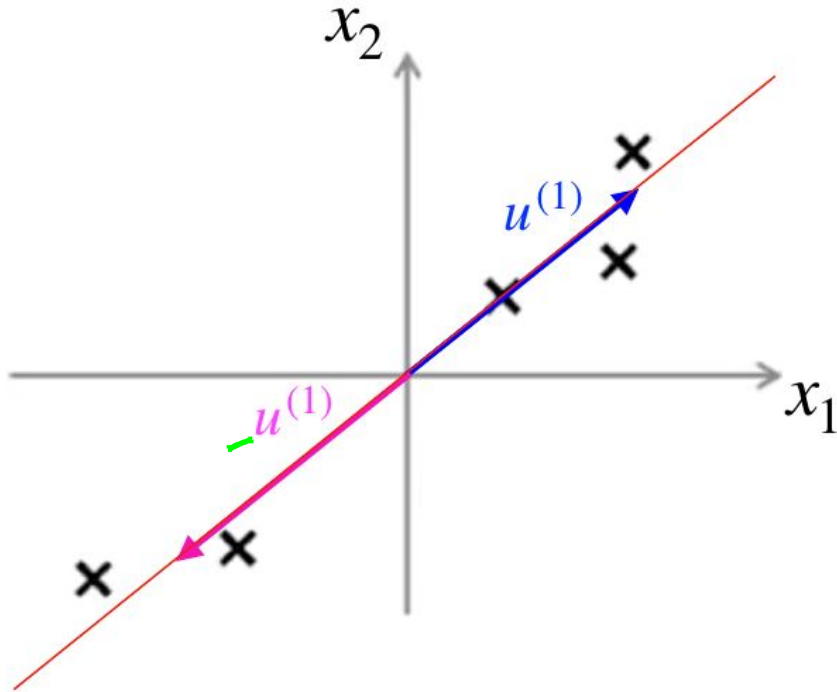
PCA: Problem Formulation (การเขียนสูตรของปัญหา)

ความเข้าใจพื้นฐาน: หาพื้นผิว (surface) ที่มีมิติต่ำกว่า เช่น เส้นสีเขียว
ที่สามารถ project (ฉายภาพ) ข้อมูลลงไปได้ และ ต้องทำให้ ผลรวมของ
กำลังสอง (sum of squares) ของเส้นสีแดง น้อยที่สุด

หมายเหตุ: ทำ feature scaling ก่อนใช้ PCA



PCA: Problem Formulation (การเขียนสูตรของปัญหา)



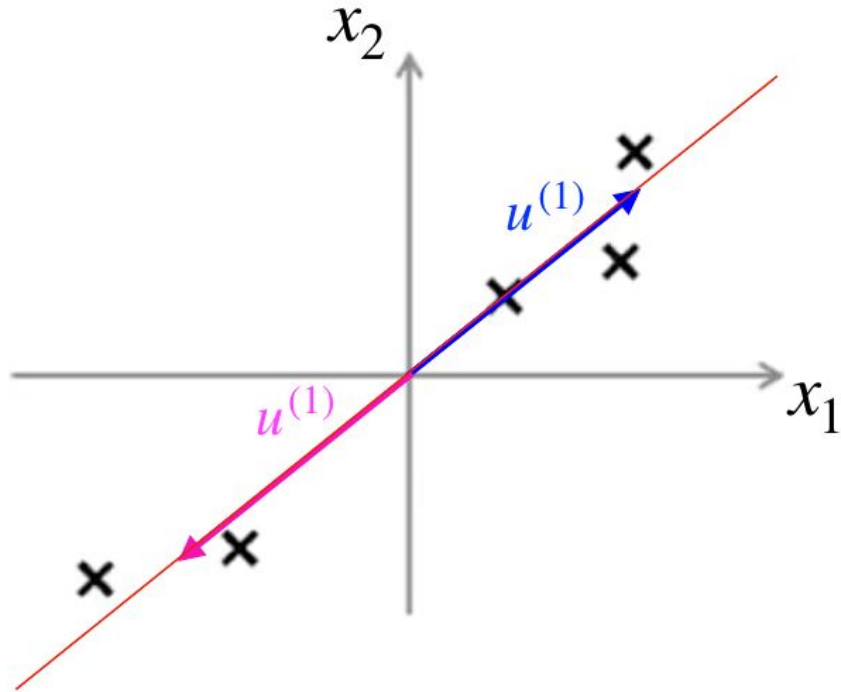
นิยาม (ลดมิติจาก 2D เป็น 1D):

หาทิศทาง (vector $u^{(j)} \in \mathbb{R}^n$) ที่ project ข้อมูลลงได้ เพื่อให้ projection error น้อยที่สุด

ในกรณีนี้ PCA ควรหา $u^{(1)}$ ให้เรา !

หมายเหตุ: ไม่ว่า PCA จะให้ค่า $u^{(1)}$ หรือ $-u^{(1)}$ ก็ไม่สำคัญ)

PCA: Problem Formulation (การเขียนสูตรของปัญหา)



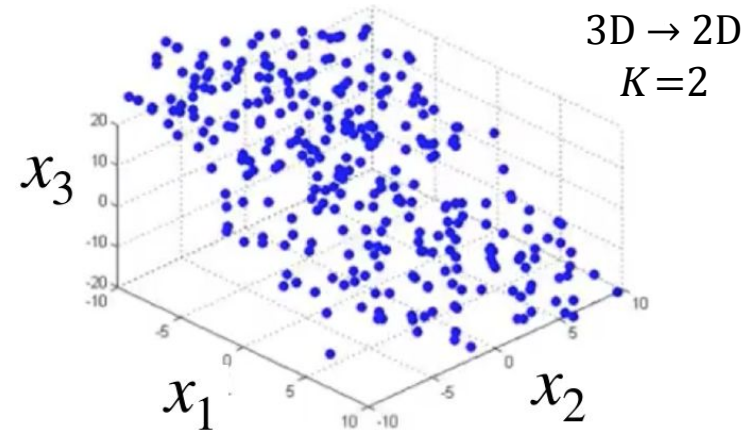
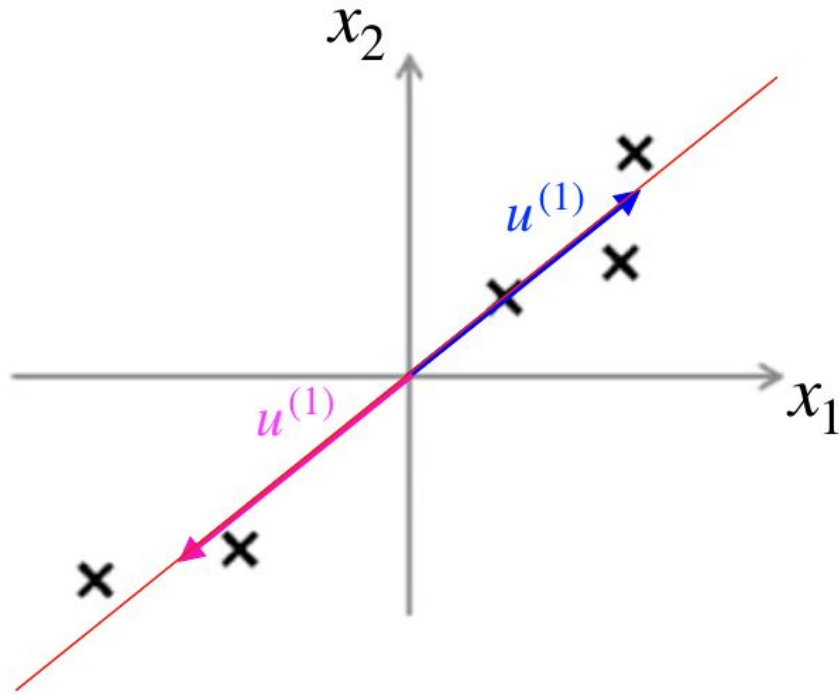
นิยาม (ลดมิติจาก 2D เป็น 1D):

หาทิศทาง (vector $u^{(1)} \in \mathbb{R}^n$) ที่ project ข้อมูลลงได้ เพื่อให้ projection error น้อยที่สุด

นิยาม (ลดมิติจาก nd เป็น kd):

หา vectors k ตัว $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ ที่ project ข้อมูลลงได้ เพื่อให้ projection error น้อยที่สุด

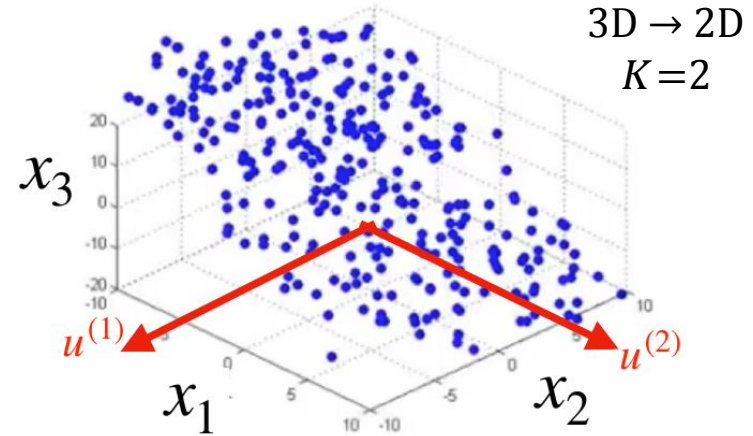
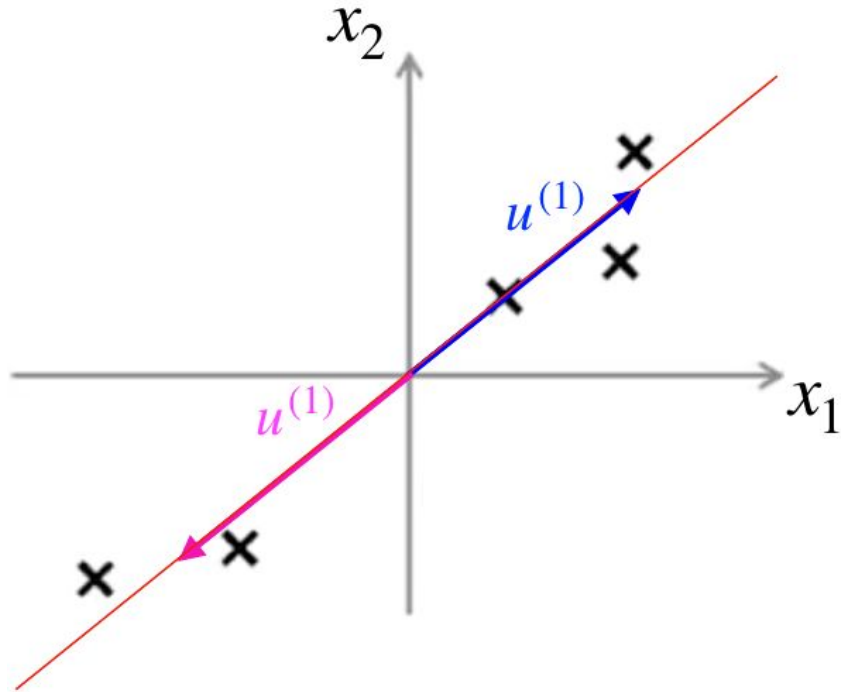
PCA: Problem Formulation (การเขียนสูตรของปัญหา)



นิยาม (ลดมิติจาก nD เป็น kD):

หา vectors k ตัว $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ ที่ project ข้อมูลลงได้ เพื่อให้ projection error น้อยที่สุด

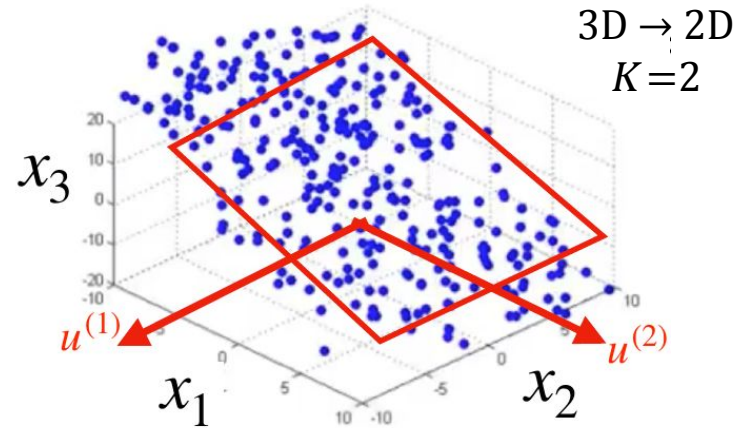
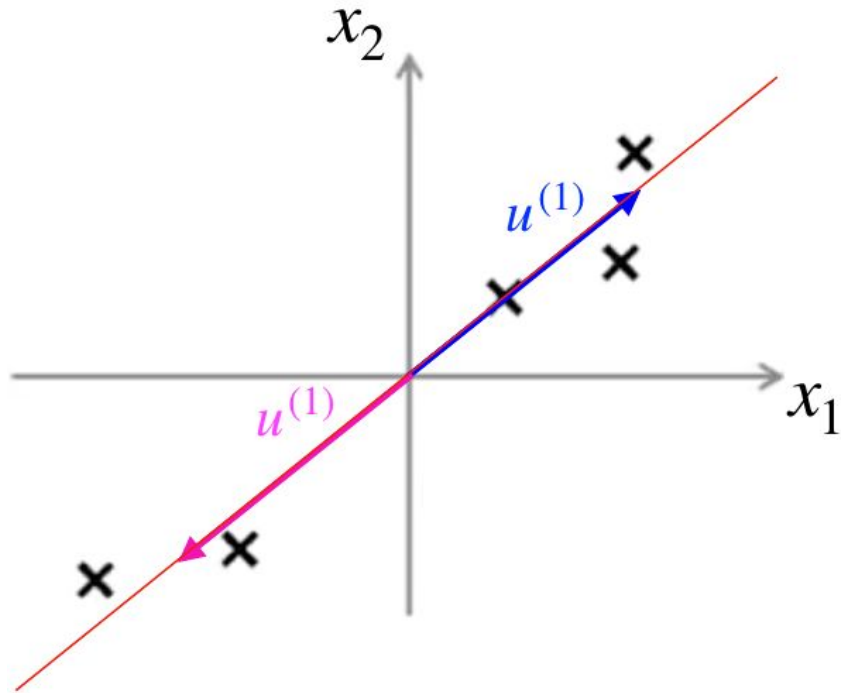
PCA: Problem Formulation (การเขียนสูตรของปัญหา)



นิยาม (ลดมิติจาก nD เป็น kD):

หา vectors k ตัว $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ ที่ project ข้อมูลลงได้ เพื่อให้ projection error น้อยที่สุด

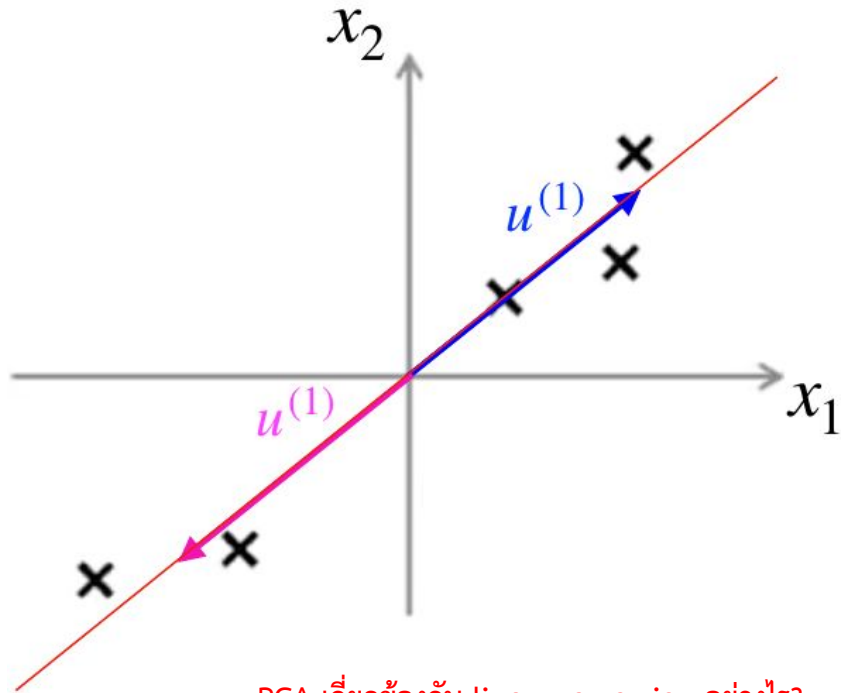
PCA: Problem Formulation (การเขียนสูตรของปัญหา)



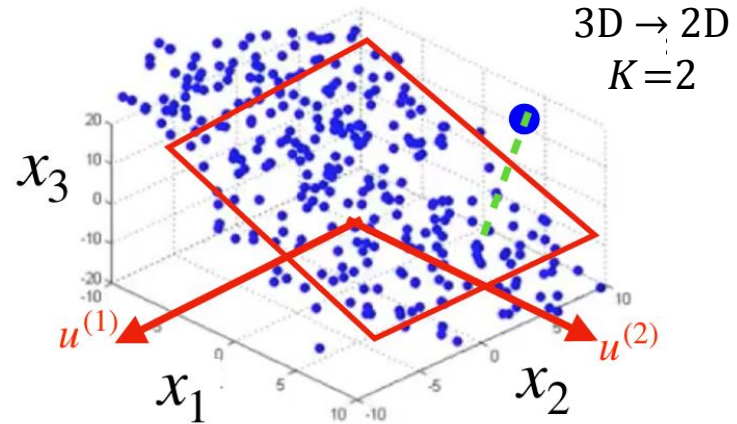
นิยาม (ลดมิติจาก nD เป็น kD):

หา vectors k ตัว $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ ที่ project ข้อมูลลงได้ เพื่อให้ projection error น้อยที่สุด

PCA: Problem Formulation (การเขียนสูตรของปัญหา)



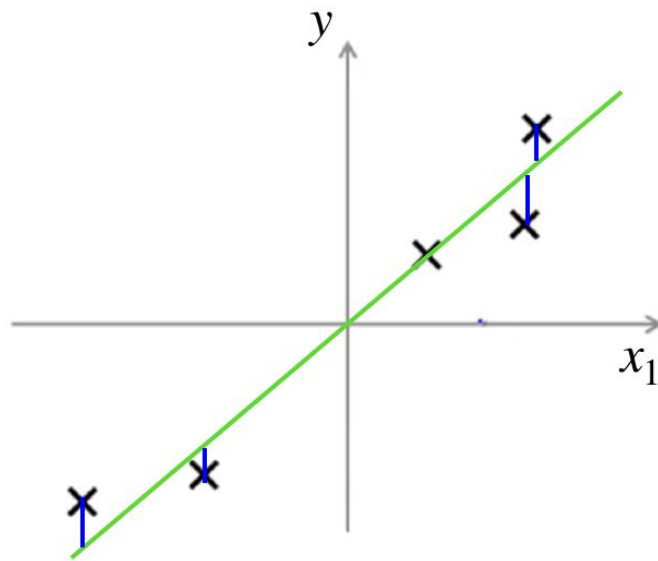
PCA เกี่ยวข้องกับ linear regression อย่างไร?



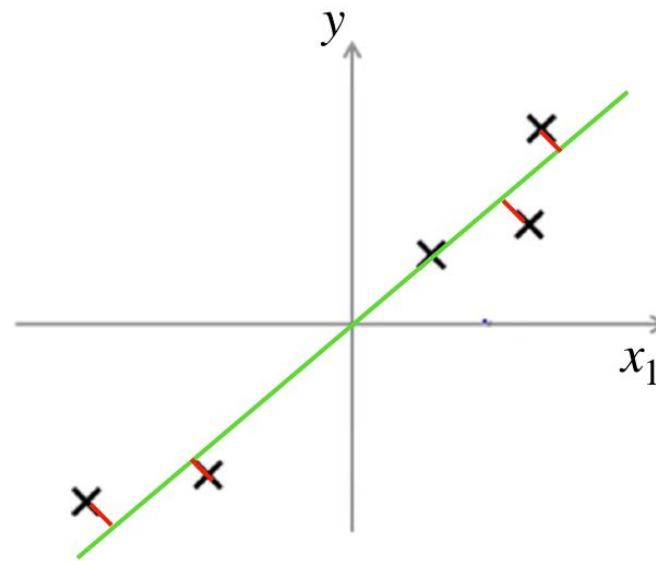
นิยาม (ลดมิติจาก nD เป็น kD):

หา vectors k ตัว $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ ที่ project ข้อมูลลงได้ เพื่อให้ projection error น้อยที่สุด

PCA ไม่ใช่ Linear Regression



Linear regression



PCA

สอง algorithm นี้ เป็น algorithm ที่ต่างกันโดยสิ้นเชิง !

Question

สมมติ run PCA กับชุดข้อมูลด้านล่าง ข้อใดต่อไปนี้น่าจะเป็น vector ที่เหมาะสม $u^{(1)}$ ที่จะ project ข้อมูล ลงไป ? (เราเลือก $u^{(1)}$ เพื่อให้

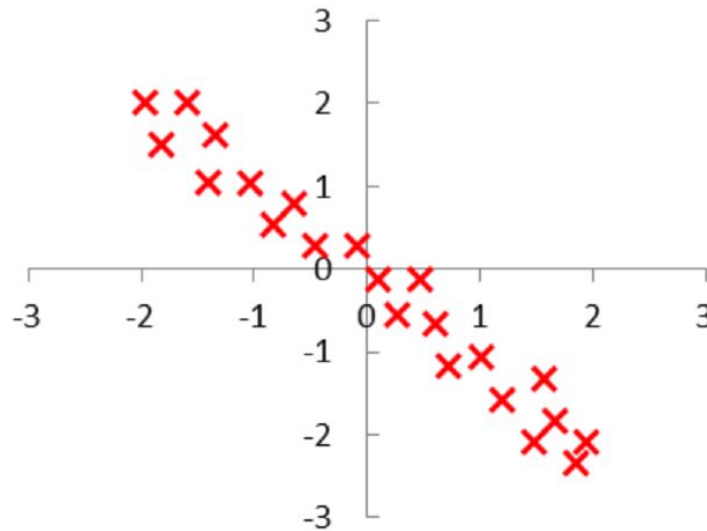
$$\text{และ ความยาวของ } \|u^{(1)}\| = \sqrt{(u_1^{(1)})^2 + (u_2^{(1)})^2}$$

$$(i) \quad u^{(1)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$(ii) \quad u^{(1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$(iii) \quad u^{(1)} = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

$$(iv) \quad u^{(1)} = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

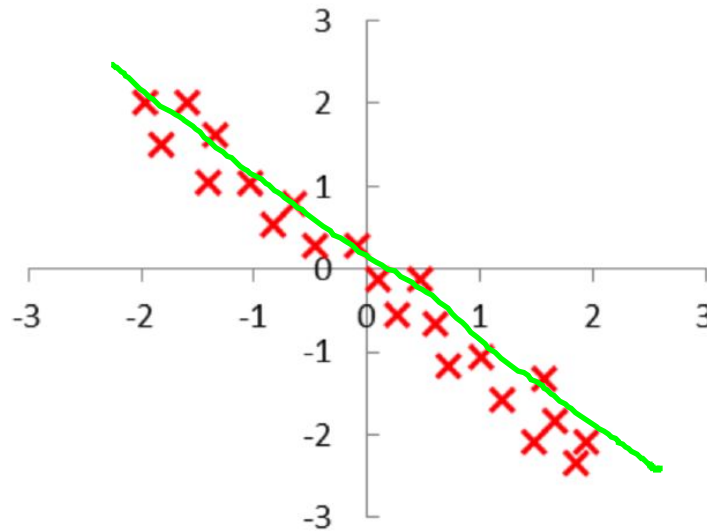


Question

สมมติ run PCA กับชุดข้อมูลด้านล่าง ข้อใดต่อไปนี้น่าจะเป็น vector ที่เหมาะสม $u^{(1)}$ ที่จะ project ข้อมูล ลงไป ? (เราเลือก $u^{(1)}$ เพื่อให้

และ ความยาวของ $v \|u^{(1)}\| = \sqrt{(u_1^{(1)})^2 + (u_2^{(1)})^2}$

- (i) $u^{(1)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- (ii) $u^{(1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- (iii) $u^{(1)} = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$
- (iv) $u^{(1)} = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$



Dimensionality Reduction

PCA: Algorithm

Krittameth Teachasrisaksakul

Data Preprocessing

ก่อนทำ PCA เราควรทำขั้นตอน data pre-processing ดังนี้ เสมอ

ชุดข้อมูล training set: $X^{(1)}, X^{(2)}, \dots, X^{(m)}$

Preprocessing (feature scaling / mean normalization):

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}$$

แทนที่ $x_j^{(i)}$ แต่ละตัว ด้วย $x_j - \mu_j$

ถ้า features แต่ละตัว อยู่ใน scale ที่ต่างกัน

(เช่น X_1 = ขนาดพื้นที่บ้าน, X_2 = จำนวนห้องนอน)

scale feature เพื่อให้ feature มีค่าอยู่ในช่วงที่เทียบกันได้

} เพื่อให้ feature แต่ละตัว มี mean เป็น 0

}
$$\frac{x_j - \mu_j}{S_j}$$

PCA Algorithm

เป้าหมาย: ลดมิติของข้อมูลจาก n เป็น k

$$\Sigma = \frac{1}{m} \sum_{i=1}^n \underbrace{(x^{(i)})}_{n \times 1} \underbrace{(x^{(i)})^T}_{1 \times n}$$

1. คำนวณ 'covariance matrix':

2. คำนวณ 'eigenvector' ของ matrix Σ :

เช่น โดยเรียก function 'singular value decomposition' หรือ **svd**

$$(U, S, V) = \text{svd}(\Sigma)$$

หมายเหตุ:

$$U = \begin{bmatrix} | & | & | & | & \dots & | \\ u^{(1)} & u^{(2)} & u^{(2)} & u^{(3)} & \dots & u^{(m)} \\ | & | & | & | & \dots & | \\ | & | & | & | & \dots & | \end{bmatrix}$$

เพื่อให้ $U \in \mathbb{R}^{n \times n}$

PCA Algorithm

เป้าหมาย: ลดมิติของข้อมูลจาก n เป็น k

$$\Sigma = \frac{1}{m} \sum_{i=1}^n \underbrace{(x^{(i)})}_{n \times 1} \underbrace{(x^{(i)})^T}_{1 \times n}$$

1. คำนวณ 'covariance matrix':

$n \times n$

2. คำนวณ 'eigenvector' ของ matrix Σ :

เช่น โดยเรียก function 'singular value decomposition' หรือ **svd**

$$(U, S, V) = \text{svd}(\Sigma)$$

หมายเหตุ:

$$U = \begin{bmatrix} | & | & | & | & \dots & | \\ | & | & | & | & \dots & | \\ u^{(1)} & u^{(2)} & u^{(2)} & u^{(3)} & \dots & u^{(m)} \\ | & | & | & | & \dots & | \\ | & | & | & | & \dots & | \end{bmatrix}$$

เพื่อให้ $U \in \mathbb{R}^{n \times n}$

เลือก k ตัวแรก !

ก็คือ $u^{(1)}, \dots, u^{(k)}$

PCA Algorithm

เป้าหมาย: ลดมิติของข้อมูลจาก n เป็น k

ก็คือ $X \in \mathbb{R}^n \mapsto z \in \mathbb{R}^k$

Solution (คำตอบ):

$$z = \left(\begin{bmatrix} | & | & | & | & \dots & | \\ | & | & | & | & \dots & | \\ u^{(1)} & u^{(2)} & u^{(2)} & u^{(3)} & \dots & u^{(k)} \\ | & | & | & | & \dots & | \\ | & | & | & | & \dots & | \end{bmatrix}_{n \times k}^T \right) \left(X_{n \times 1} \right)$$
$$= \left(\begin{bmatrix} - & - & (u^{(1)})^T & - & - \\ & & \vdots & & \\ - & - & (u^{(k)})^T & - & - \end{bmatrix}_{k \times n} \right) \left(X_{n \times 1} \right)$$

บางครั้งเรียกว่า 'U-reduced'

$\therefore z \in \mathbb{R}^k$

PCA Algorithm

เป้าหมาย: ลดมิติของข้อมูลจาก n เป็น k

ก็คือ $X \in \mathbb{R}^n \mapsto z \in \mathbb{R}^k$

Solution (คำตอบ):

$$z^{(i)} = \left(\begin{bmatrix} | & | & | & | & \dots & | \\ u^{(1)} & u^{(2)} & u^{(2)} & u^{(3)} & \dots & u^{(k)} \\ | & | & | & | & \dots & | \\ | & | & | & | & \dots & | \end{bmatrix}_{n \times k}^T \right) \left(X_{n \times 1}^{(i)} \right)$$

$$= \left(\begin{bmatrix} - & - & (u^{(1)})^T & - & - \\ & & \vdots & & \\ - & - & (u^{(k)})^T & - & - \end{bmatrix}_{k \times n} \right) \left(X_{n \times 1}^{(i)} \right) \quad \therefore z^{(i)} \in \mathbb{R}^k$$

บางครั้งเรียกว่า 'U-reduced'

PCA Algorithm (Vectorizing)

เป้าหมาย:

ลดมิติของข้อมูลจาก n เป็น k

1. คำนวณ 'covariance matrix':

$$\Sigma = \frac{1}{m} \sum_{i=1}^n \underbrace{(x^{(i)})}_{n \times 1} \underbrace{(x^{(i)})^T}_{1 \times n}$$

2. คำนวณ 'eigenvector' ของ matrix Σ ($n \times n$):

เช่น โดยเรียก function 'singular value decomposition' หรือ **svd**

$$(U, S, V) = \text{svd}(\Sigma)$$

3. $U_{\text{reduce}} := U(:, 1:k)$

4. $z := (U_{\text{reduce}}^T)(X)$

Note: $X \in \mathbb{R}^n$ (not $X \in \mathbb{R}^{n+1}$)

$$X \in \mathbb{R}^n \mapsto z \in \mathbb{R}^k$$

$$\therefore X = \begin{bmatrix} - & - & (x^{(1)})^T & - & - \\ - & - & \vdots & - & - \\ - & - & (x^{(m)})^T & - & - \end{bmatrix}$$

$$\therefore \Sigma = \frac{1}{m} X^T X$$

Question

ใน PCA : เราได้ $z \in \mathbb{R}^k$ จาก $x \in \mathbb{R}^n$ โดยใช้

$$z = \left(\begin{bmatrix} | & | & \dots & | \\ u^{(1)} & u^{(2)} & \dots & u^{(k)} \\ | & | & \dots & | \end{bmatrix}^T \right) (x) = \left(\begin{bmatrix} - & - & (u^{(1)})^T & - & - \\ - & - & (u^{(2)})^T & - & - \\ - & - & \vdots & - & - \\ - & - & (u^{(k)})^T & - & - \end{bmatrix} \right) (x)$$

ข้อใดต่อไปนี้เป็น expression ที่ถูกต้องของ z_j ? j = 2

- (i) $z_j = (u^{(k)})^T x$
- (ii) $z_j = (u^{(j)})^T x_j$
- (iii) $z_j = (u^{(j)})^T x_k$
- (iv) $z_j = (u^{(j)})^T x$

Question

ใน PCA : เราได้ $z \in \mathbb{R}^k$ จาก $x \in \mathbb{R}^n$ โดยใช้

$$z = \left(\begin{bmatrix} | & | & \dots & | \\ u^{(1)} & u^{(2)} & \dots & u^{(k)} \\ | & | & \dots & | \end{bmatrix}^T \right) (x) = \left(\begin{bmatrix} - & - & (u^{(1)})^T & - & - \\ - & - & (u^{(2)})^T & - & - \\ - & - & \vdots & - & - \\ - & - & (u^{(k)})^T & - & - \end{bmatrix} \right) (x)$$

ข้อใดต่อไปนี้เป็น expression ที่ถูกต้องของ z_j ?

- (i) $z_j = (u^{(k)})^T x$
- (ii) $z_j = (u^{(j)})^T x_j$
- (iii) $z_j = (u^{(j)})^T x_k$
- (iv) $z_j = (u^{(j)})^T x$

Dimensionality Reduction

Reconstruction from Compressed Representation

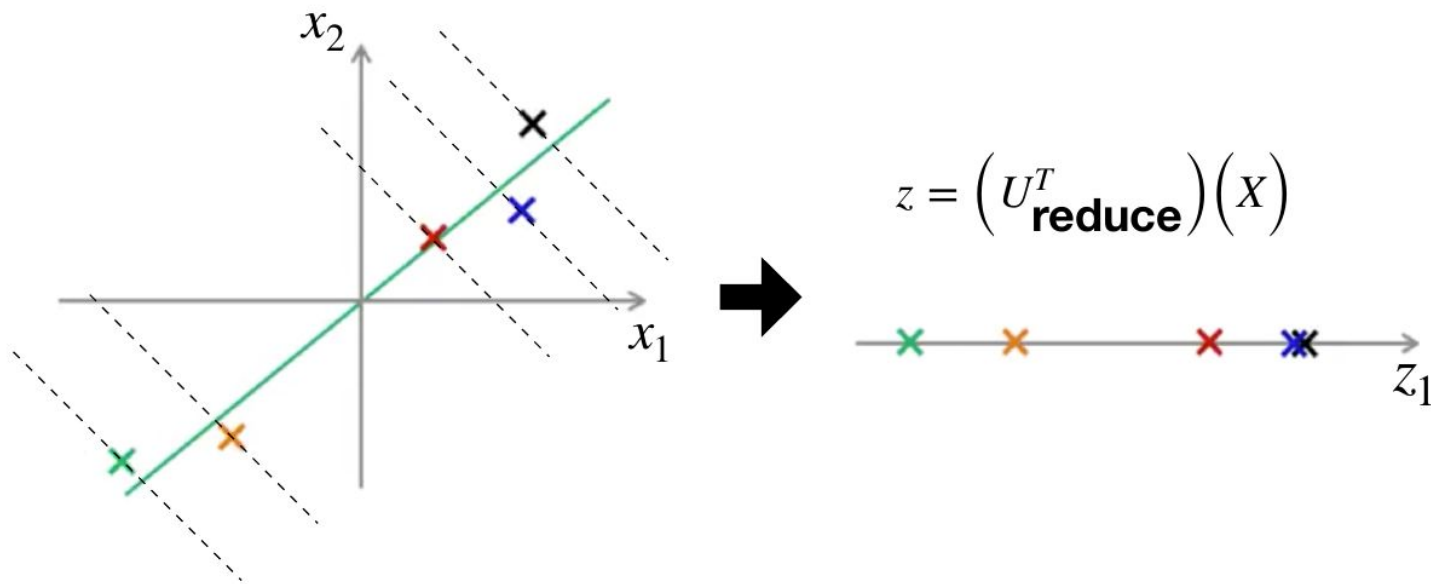
(การฟื้นฟู/สร้างข้อมูล จากตัวแทนที่ถูกบีบอัด)

Krittameth Teachasrisaksakul

ความเข้าใจพื้นฐาน

คำถาม: ถ้า PCA เป็น compression algorithm (algorithm บีบอัดข้อมูล)

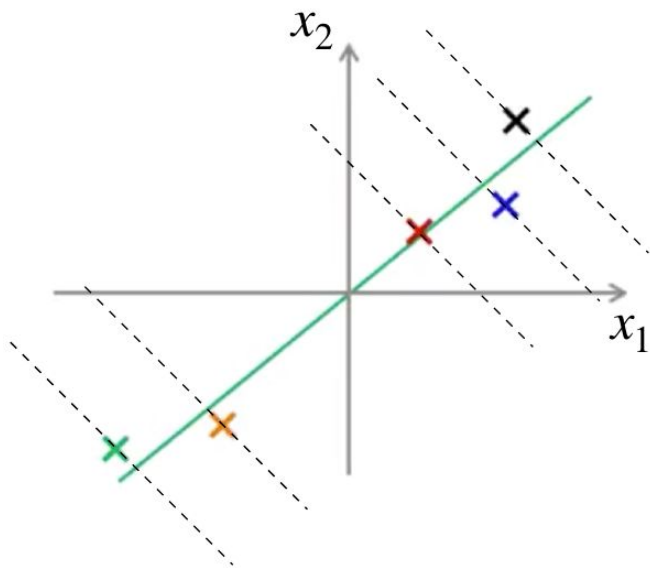
ควรมีทาง de-compress (ย้อนคืนการบีบอัด) representation กลับเป็นข้อมูลดั้งเดิม (โดยประมาณ) ?



ความเข้าใจพื้นฐาน

คำถาม: ถ้า PCA เป็น compression algorithm (algorithm บีบอัดข้อมูล)

ควรมีทาง de-compress (ย้อนคืนการบีบอัด) representation กลับเป็นข้อมูลดั้งเดิม (โดยประมาณ) ?



$$z = \left(U_{\text{reduce}}^T \right) (X)$$



เราอยากให้ $z \in \mathbb{R}^k \mapsto X \in \mathbb{R}^n$

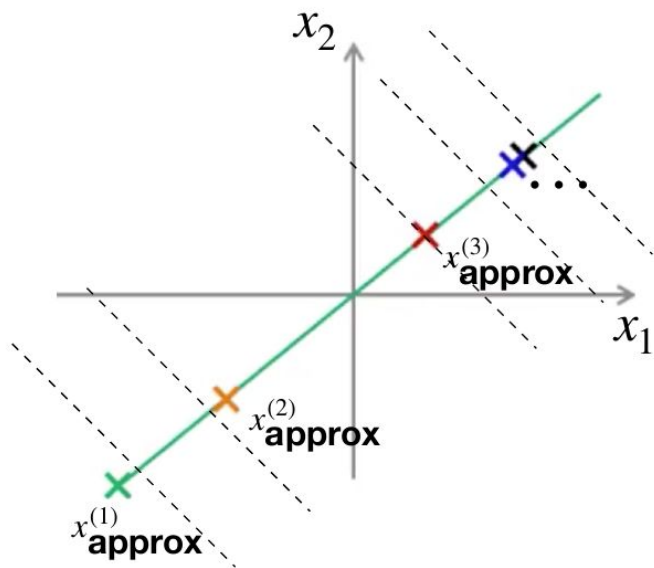
$$X_{\text{approx}} = \underbrace{\left(U_{\text{reduce}} \right)}_{n \times k} \underbrace{(z)}_{k \times 1}$$

$$\therefore X_{\text{approx}} \in \mathbb{R}^n$$

ความเข้าใจพื้นฐาน

คำถาม: ถ้า PCA เป็น compression algorithm (algorithm บีบอัดข้อมูล)

ควรมีทาง de-compress (ย้อนคืนการบีบอัด) representation กลับเป็นข้อมูลดั้งเดิม (โดยประมาณ) ?



$$z = (U^T \text{reduce})(X)$$

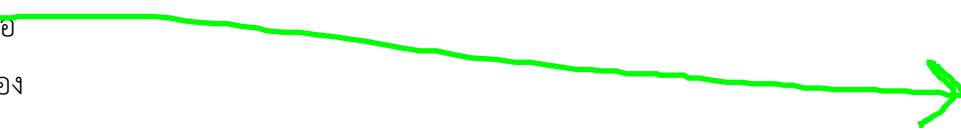


เราอยากให้ $z \in \mathbb{R}^k \mapsto X \in \mathbb{R}^n$

$$X_{\text{approx}} = \underbrace{(U \text{reduce})}_{n \times k} \underbrace{(z)}_{k \times 1}$$

$$\therefore X_{\text{approx}} \in \mathbb{R}^n$$

Question

สมมติ run PCA ด้วย $k = n$ เพื่อให้ dimension ของข้อมูลไม่ลดลงเลย (นี่ไม่มีประโยชน์ในทางปฏิบัติ แต่เป็นแบบฝึกหัดที่ดี) ทบทวน: percent หรือ สัดส่วนของ variance ที่ถูกรักษาไว้ คือ  $\frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}}$ ข้อใดต่อไปนี้เป็นจริง? วงทุกข้อที่ถูกต้อง

- (i) U_{reduce} จะเป็น matrix ขนาด $n \times n$
- (ii) $X_{\text{approx}} = X$ สำหรับค่า X ทุกค่า
- (iii) percentage ของ variance ที่ถูกรักษาไว้ จะเป็น 100%
- (iv) จะได้ว่า $\frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}} > 1$

Question

สมมติ run PCA ด้วย $k = n$ เพื่อให้ dimension ของข้อมูลไม่ลดลงเลย (นี่ไม่มีประโยชน์ในทางปฏิบัติ แต่เป็นแบบฝึกหัดที่ดี) ทบทวน: percent หรือ สัดส่วนของ variance ที่ถูกรักษาไว้ คือ
ข้อใดต่อไปนี้เป็นจริง? วงทุกข้อที่ถูกต้อง

$n \times k$

$$\frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}}$$

(i) U_{reduce} จะเป็น matrix ขนาด $n \times n$

(ii) $X_{\text{approx}} = X$ สำหรับค่า X ทุกค่า

(iii) percentage ของ variance ที่ถูกรักษาไว้ จะเป็น 100%

(iv) จะได้ว่า $\frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}} > 1$

Dimensionality Reduction

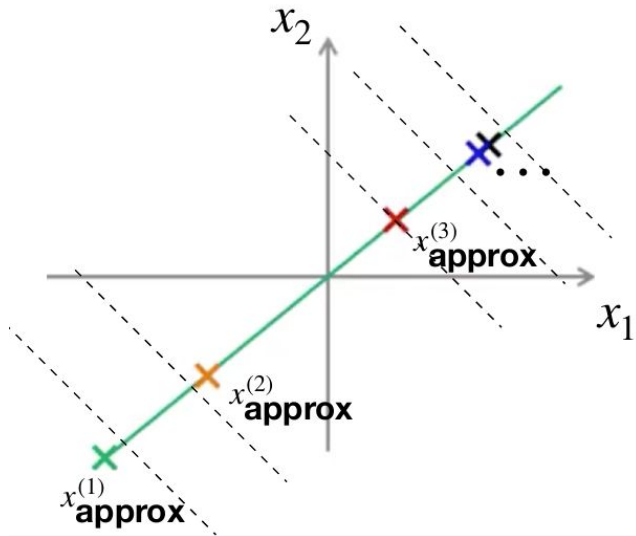
การเลือกจำนวน Principal Components

Krittameth Teachasrisaksakul

เลือกค่า k อย่างไร ?

เกณฑ์การเลือก k (ก็คือ จำนวน principal components)

- Average squared projection error:
- Total variation in the data:



$$\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x^{\text{approx}}\|^2$$

$$\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2$$

โดยทั่วไป เลือก k เป็นค่าที่น้อยที่สุด เพื่อให้

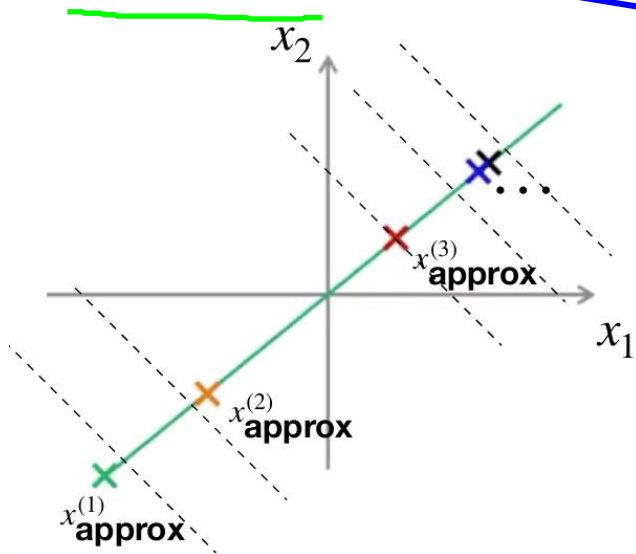
$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x^{\text{approx}}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01$$

ก็คือ 99% ของ variance ยังถูกรักษาไว้

เลือกค่า k อย่างไร ?

เกณฑ์การเลือก k (ก็คือ จำนวน principal components)

- Average squared projection error:
- Total variation in the data:



$$\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x^{(i) \text{ approx}}\|^2$$

$$\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2$$

โดยทั่วไป เลือก k เป็นค่าที่น้อยที่สุด เพื่อให้

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x^{(i) \text{ approx}}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01$$

ก็คือ 99% ของ variance ยังถูกรักษาไว้

Algorithm: เลือก k อย่างไร

Input: ไม่มี

Output: k

ตั้งค่า $k = 1$

ทำซ้ำ จนกระทั่ง เงื่อนไขเป็นจริง (satisfied)

คำนวณ U_{reduce} , $z^{(1)}, z^{(2)}, \dots, z^{(m)}$, $x_{\text{approx}}^{(1)}, \dots, x_{\text{approx}}^{(m)}$

ตรวจสอบว่า เงื่อนไขเป็นจริงหรือไม่ ก็คือ

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{\text{approx}}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01?$$

ถ้ายัง increment k

Algorithm: เลือก k อย่างไร

Input: ไม่มี

Output: k

ตั้งค่า $k = 1$

ทำซ้ำ จนกระทั่ง เงื่อนไขเป็นจริง (satisfied)

คำนวณ $U_{\text{reduce}}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x_{\text{approx}}^{(1)}, \dots, x_{\text{approx}}^{(m)}$

ตรวจสอบว่า เงื่อนไขเป็นจริงหรือไม่ ก็คือ

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{\text{approx}}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01?$$

ถ้าไม่เป็นจริง : increment k

$$(U(S)V) = \text{svd}(\Sigma)$$

$$S = \begin{bmatrix} S_{11} & 0 & 0 & 0 & 0 \\ 0 & S_{22} & 0 & 0 & 0 \\ 0 & 0 & S_{33} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & S_{nn} \end{bmatrix}$$

สำหรับค่า k ที่มี

$$= 1 - \frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}}$$

Algorithm: เลือก k อย่างไร

Input: ไม่มี

Output: k

ตั้งค่า $k = 1$

ทำซ้ำ จนกระทั่ง เงื่อนไขเป็นจริง (satisfied)

คำนวณ $U_{\text{reduce}}, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x_{\text{approx}}^{(1)}, \dots, x_{\text{approx}}^{(m)}$

ตรวจสอบว่า เงื่อนไขเป็นจริงหรือไม่ ก็คือ

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{\text{approx}}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01?$$

ถ้าไม่เป็นจริง : increment k

$$(U, S, V) = \text{svd}(\Sigma)$$

$$S = \begin{bmatrix} S_{11} & 0 & 0 & 0 & 0 \\ 0 & S_{22} & 0 & 0 & 0 \\ 0 & 0 & S_{33} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & S_{nn} \end{bmatrix}$$

สำหรับค่า k ที่สี่

$$= 1 - \frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}}$$

ถ้าทำแบบนี้ เราไม่จำเป็นต้อง run PCA ใหม่ตั้งแต่ต้น
ซ้ำแล้วซ้ำอีก

Question

ก่อนหน้านี้ เรากล่าวว่า PCA เลือกทิศทาง $u^{(1)}$ (หรือทิศทาง k ทิศทาง $u^{(1)}, \dots, u^{(k)}$) ที่ project ข้อมูลลงไปได้ เพื่อให้ the (squared) projection error น้อยที่สุด อีกวิธีที่จะพูดแบบเดียวกัน ก็คือ PCA พยายาม ทำให้ function ใด น้อยที่สุด

$$(i) \quad \frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2$$

$$(ii) \quad \frac{1}{m} \sum_{i=1}^m \|x^{(i)}_{\text{approx}}\|^2$$

$$(iii) \quad \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x^{(i)}_{\text{approx}}\|^2$$

$$(iv) \quad \frac{1}{m} \sum_{i=1}^m \|x^{(i)} + x^{(i)}_{\text{approx}}\|^2$$

Question

ก่อนหน้านี้ เรากล่าวว่า PCA เลือกทิศทาง $u^{(1)}$ (หรือทิศทาง k ทิศทาง $u^{(1)}, \dots, u^{(k)}$) ที่ project ข้อมูลลงไปได้ เพื่อให้ the (squared) projection error น้อยที่สุด อีกวิธีที่จะพูดแบบเดียวกัน ก็คือ PCA พยายาม ทำให้ function ใด น้อยที่สุด

$$(i) \quad \frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2$$

$$(ii) \quad \frac{1}{m} \sum_{i=1}^m \|x^{(i)}_{\text{approx}}\|^2$$

$$(iii) \quad \frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x^{(i)}_{\text{approx}}\|^2$$

$$(iv) \quad \frac{1}{m} \sum_{i=1}^m \|x^{(i)} + x^{(i)}_{\text{approx}}\|^2$$

Dimensionality Reduction

คำแนะนำเกี่ยวกับ การใช้ PCA

Krittameth Teachasrisaksakul

PCA สำหรับ Speed-Up Learning

ชุดข้อมูลสำหรับ supervised learning:

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)}), \text{ where } x^{(i)} \in \mathbb{R}^{30,000}$$

1. ดึง input ออกมา:

ชุดข้อมูลที่ไม่มี label :
(unlabeled dataset)

$$x^{(1)}, x^{(2)}, \dots, x^{(m)} \in \mathbb{R}^{30,000}$$

PCA

$$z^{(1)}, z^{(2)}, \dots, z^{(m)} \in \mathbb{R}^{3,000}$$

2. training dataset ใหม่:

$$(z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \dots, (z^{(m)}, y^{(m)})$$

100 px

100 px



PCA สำหรับ Speed-Up Learning

ชุดข้อมูลสำหรับ supervised learning:

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)}), \text{ where } x^{(i)} \in \mathbb{R}^{30,000}$$

1. ดึง input ออกมา:

ชุดข้อมูลที่ไม่มี label :
(unlabeled dataset)

$$x^{(1)}, x^{(2)}, \dots, x^{(m)} \in \mathbb{R}^{30,000}$$

PCA

$$z^{(1)}, z^{(2)}, \dots, z^{(m)} \in \mathbb{R}^{3,000}$$

2. training dataset ใหม่:

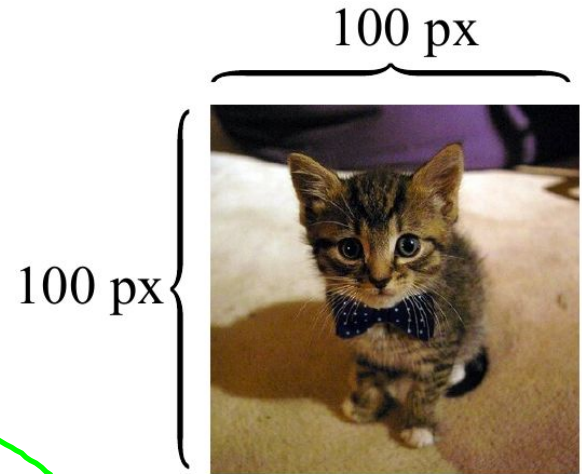
$$(z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \dots, (z^{(m)}, y^{(m)})$$

ถ้ามี X ตัวใหม่ เราจะทำ:

(assume ว่า เราใช้ logistic regression
model)

$$x \rightarrow z \rightarrow h_{\theta}(z) = \frac{1}{1 + e^{-\theta^T z}}$$

ก็คือ run PCA กับ training set !



การประยุกต์ใช้ PCA

- Compression / การบีบอัดข้อมูล
 - ลดขนาด memory / disk ที่ต้องใช้เก็บข้อมูล
 - Speed up learning algorithm
- Visualization / การนำเสนอข้อมูลเป็นภาพ
 - เพราะเราสามารถ plot ได้เพียงข้อมูล 2D หรือ 3D
 - บ่อยครั้ง เราจึงตั้งค่า $k = 2$ หรือ $k = 3$



เลือก k โดยอิงกับ percentage of variance ที่ถูก
รักษาไว้ (variance retained)

การใช้ PCA แบบไม่ถูกต้อง

1. ใช้ PCA เพื่อป้องกัน การเกิด overfitting

ใช้ $z^{(i)}$ แทน $x^{(i)}$ เพื่อลดจำนวน features เป็น $k < n$

เพราะจำนวน features ถูกลดลง แล้วมันจะมีแนวโน้มน้อยลงที่จะ overfit !

นี้อาจทำงานได้ OK แต่ไม่ใช่วิธีที่ดีที่จะแก้ overfitting

ใช้ regularization แทน ก็คือ

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

ทำไมวิธีนี้ ไม่ใช่ วิธีที่ดีสำหรับแก้ overfitting .

(ข้อมูลบางอย่างจะหายไป !)

การใช้ PCA แบบไม่ถูกต้อง

2. ใช้ PCA โดยไม่มีเหตุผลสนับสนุน

ต่อไปนี้เป็น design ของระบบ machine learning

- หา training dataset $\{ (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)}) \}$
- Run PCA เพื่อลด dimension ของ $x^{(i)}$ และหา $z^{(i)}$
- Train logistic regression โดยใช้ $\{ (z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \dots, (z^{(m)}, y^{(m)}) \}$
- Test โดยใช้ test set ก็คือ map (เชื่อมโยง) $x_{\text{test}}^{(i)}$ to $z_{\text{test}}^{(i)}$
- Run $h_{\theta}(z)$ กับ $\{ (z_{\text{test}}^{(1)}, y^{(1)}), (z_{\text{test}}^{(2)}, y^{(2)}), \dots, (z_{\text{test}}^{(m)}, y^{(m)}) \}$

แล้วถ้าทำทั้งหมดนี้ โดยไม่ใช้ PCA จะเป็นอย่างไร?

Learning algorithm run เข้าเกินไป หรือ

ต้องใช้ memory และ ขนาด disk (disk space) มากเกินไป

Question

ข้อใดต่อไปนี้เป็น การประยุกต์ใช้ PCA ที่ดี (ที่แนะนำ)?

วงทุกข้อที่ถูกต้อง

- (i) เพื่อบีบอัดข้อมูล เพื่อให้ใช้ computer memory / disk space น้อยลง
- (ii) เพื่อลดมิติของ input data เพื่อให้ learning algorithm ทำงานเร็วขึ้น
- (iii) แทนที่จะใช้ regularization ใช้ PCA เพื่อลดจำนวน features เพื่อลด overfitting
- (iv) เพื่อ visualize (นำเสนอด้วยภาพ) ข้อมูลที่มีมิติสูง (high-dimensional data) (โดยเลือก $k = 2$ หรือ $k = 3$)

Question

ข้อใดต่อไปนี้เป็น การประยุกต์ใช้ PCA ที่ดี (ที่แนะนำ)?

วงทุกข้อที่ถูกต้อง

- (i) เพื่อบีบอัดข้อมูล เพื่อให้ใช้ computer memory / disk space น้อยลง
- (ii) เพื่อลดมิติของ input data เพื่อให้ learning algorithm ทำงานเร็วขึ้น
- (iii) แทนที่จะใช้ regularization ใช้ PCA เพื่อลดจำนวน features เพื่อลด overfitting
- (iv) เพื่อ visualize (นำเสนอด้วยภาพ) ข้อมูลที่มีมิติสูง (high-dimensional data) (โดยเลือก $k = 2$ หรือ $k = 3$)

PCA vs. LDA

c

Dimensionality Reduction

Unsupervised Learning Algorithm อื่นๆ

Krittameth Teachasrisaksakul

Learning Algorithms อื่นๆ

ในช่วงก่อนหน้านี้ เราได้เรียนรู้ (โดยย่อๆ) เกี่ยวกับการใช้ K-means ทำ clustering และการใช้ PCA สำหรับทำ dimensionality reduction (การลดมิติของข้อมูล)

มี learning algorithm อื่นๆ อะไรบ้าง ?

Unsupervised learning algorithm ส่วนมาก มีเป้าหมายหนึ่งข้อ หรือมากกว่าหนึ่งข้อ ดังนี้

1. **Clustering** เพื่อ discretizing (นำเสนอหรือแทนข้อมูลด้วยค่า/ปริมาณที่ไม่ต่อเนื่อง / discrete) หรือ ตรวจจับ anomaly (anomaly detection)
2. **Probability density estimation** เพื่อตรวจจับ positive (positive detection), ตรวจจับ anomaly, สังเคราะห์ข้อมูลใหม่ที่มีการกระจายตัวคล้ายกับชุดข้อมูล training set
3. **Latent space discovery** เพื่อทำ dimensionality reduction, ตรวจจับ positive, สังเคราะห์ข้อมูลใหม่ที่มีการกระจายตัวคล้ายกับชุดข้อมูล training set

Clustering

เราได้เห็นไปแล้วว่า K-means ทำ clustering กับข้อมูล อย่างไร

การทำ clustering ในลักษณะนี้ บางครั้ง เรียกว่า '**vector quantization**' → เป้าหมายของมัน คือ การ coding (เข้ารหัส/แปลง) inputs เป็นสมาชิกของ discrete set (เซตที่มีค่าไม่ต่อเนื่อง) โดยอ้างอิงจาก **spatial locality** (ความใกล้เคียงพื้นที่/ตำแหน่ง)

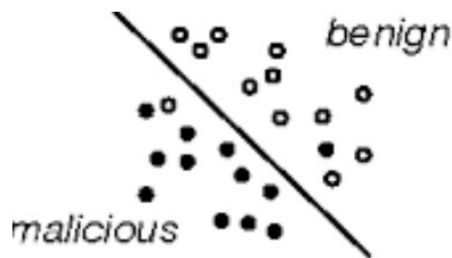
clustering algorithm อีกประเภทหนึ่ง คือ วิธี **pairwise clustering** ที่ใช้ **similarity distance** เพื่อจัดกลุ่ม inputs มี 2 ประเภท คือ

- **Hierarchical clustering** เป็นแบบ **top-down**
- **Agglomerative clustering** เป็นแบบ **bottom-up**

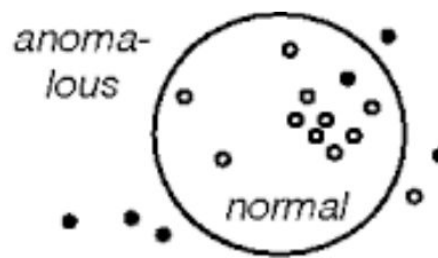
Density Estimation : การประมาณค่าความหนาแน่น

ตัวประมาณค่าความหนาแน่น (Density estimators) สามารถใช้ทำ **การตรวจจับ / detection** (input ใดๆ ที่มี probability density มากกว่าค่าบางค่า จะถูกแยกประเภทเป็น positive)

Density estimators สามารถใช้ทำ **anomaly detection** (input ใดๆ ที่มี probability density ต่ำกว่าค่าบางค่า จะถูกแยกประเภทเป็น anomalous = ผิดปกติ หรือต่างจากเกณฑ์ปกติ)



(a) Classification



(b) Anomaly detection

Latent Space Discovery : การค้นพบปริภูมิแอบแฝง

วิธี **Latent space** เชื่อมโยง (map) input space ไปยัง **representation** ที่มีมิติต่ำกว่า (lower-dimensional), เรียกว่า **latent / semantic representation**

Principle component analysis (PCA) สร้างแบบจำลอง (model) ของ latent space เป็นการกระจายตัวแบบปกติ (Gaussian distribution) ใน **linear subspace** ที่มีมิติเป็น k ของ space ดั้งเดิม ซึ่งข้อมูล input มี variance สูงสุด

Locally-linear embedding (LLE) และวิธี dimensionality reduction ที่ไม่เป็นเชิงเส้น (non-linear) อื่นๆ สร้างแบบจำลองของข้อมูลเป็น เหมือนกับที่สร้างจาก **non-linear submanifold** ของ input space

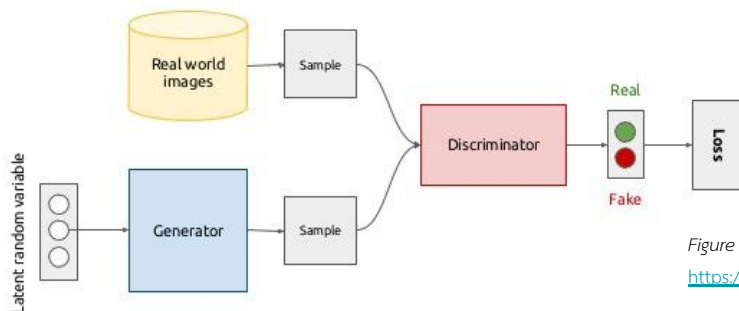
Latent Dirichlet allocation (LDA) เป็น **topic model** ในสาขา NLP ที่ **imposes** a prior on การกระจายตัวของข้อมูล (data distribution) ใน latent space

Latent Space Discovery : การค้นพบปริภูมิแอบแฝง

- วิธี Latent space เชื่อมโยง (map) input space ไปยัง representation ที่มีมิติต่ำกว่า (lower-dimensional), เรียกว่า latent / semantic representation
- Generative adversarial networks (GANs) represent การเชื่อมโยง (mapping) จาก latent space ไปยัง data space ด้วย neural network ซึ่งส่วนมากเป็น deconvolutional neural network
- Generative model นี้ถูก train (ฝึก) alongside discriminative adversary



Generative adversarial networks (conceptual)



Figure

source:

<https://medium.com/archieai/a-dozen-times-artificial-intelligence-startled-the-world-eae5005153db>

สรุป

Unsupervised learning เป็น area ที่ rich อย่างมาก และเป็นหัวข้อที่สามารถสอนแยกเป็นคอร์สอีกคอร์สได้ !

References

1. Andrew Ng, Machine Learning, Coursera.
2. Teeradaj Racharak, AI Practical Development Bootcamp.
3. What is Machine Learning?, <https://www.digitalskill.org/contents/5>