W1

15 May 2025 17:14

L 1.3 - Representation learning

Goal: Given a set of "data points" "understand" something "useful" about it

Datapoints:- vectors in $\mathbb{R}^d$

$$\begin{bmatrix} \text{height} \\ \text{weight} \\ \text{age} \end{bmatrix} \in \mathbb{R}^3$$

Running Theme:- "Comprehension is compression"

Problem: Input $= \{x_1, x_2, \ldots, x_n\}$

$x_i \in \mathbb{R}^d$ $d \leftarrow$ no. of features

Output = Some compressed representation

Example:-

-7	2.5	0.5	0
-14	5	1	0

→ 8 data points

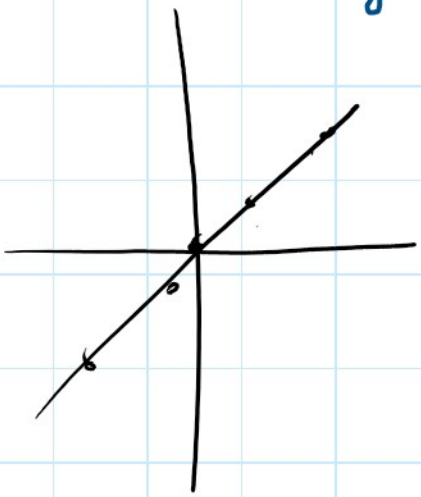
→ As relationship

$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$; $\{-7, 2.5, 0.5 0\}$

[2] , [illegible]

→ 6 points.

→ larger savings in scale



Rep - I

$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$

Rep - II

$\begin{bmatrix} 1/\sqrt{5} \\ 2/\sqrt{5} \end{bmatrix}$ $\{-7\sqrt{5}, 2.5\sqrt{5}, 0.5\sqrt{5}, 0\}$

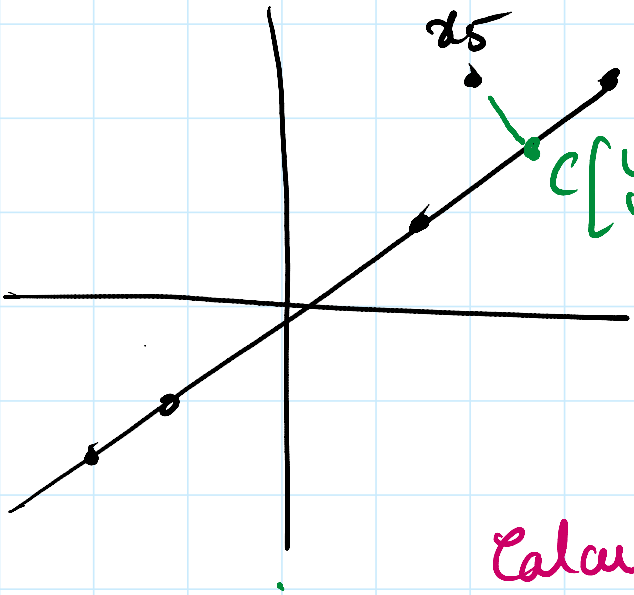
★ Any vector along line can be chosen except $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Compression:

real number = $d \times n$

compressed = $d + n$

★ The above for perfectly fit. what if there are other points



→ Something on the line act as proxy on line

→ Projection i.e dot product.

Calculating projection

→ length of error vector

$$\begin{bmatrix} x_1 - cw_1 \\ x_2 - cw_2 \end{bmatrix}$$

$$\min_c \quad (x_1 - cw_1)^2 + (x_2 - cw_2)^2$$

$$c^* = \left(\frac{x_1 w_1 + x_2 w_2}{w_1^2 + w_2^2} \right) \quad \text{(Scalar)}$$

dot product.

$$c\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \left(\frac{x^T w}{\|w\|} \right) \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

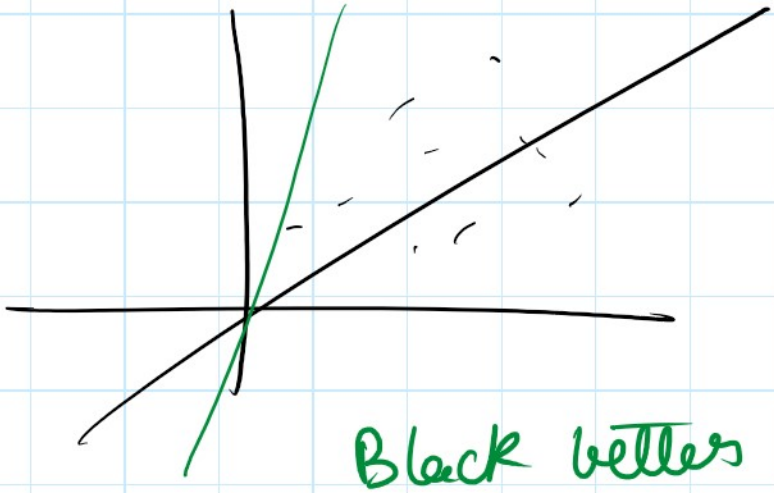
→ If choosing w_1, w_2, where $\|w\| = 1$,

$$c\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = x^T w \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

L 1.4 – Representation learning p2

→ Real life lines not given, random dataset is given.

Goal: Develop way to find a "compressed" representation of data when datapoints not necessarily on line



Black better cuz less reconstruction error

dataset: $\{x_1, x_2 \dots x_n\}$ $x_i \in \mathbb{R}^d$

ERROR (line, dataset) $\|\omega\| = 1$

$$= \sum_{i=1}^{n} \text{length}^2 (x - (x^T\omega)\omega)$$

$$f(\omega) = \frac{1}{n} \sum_{i=1}^{n} \|x - (x^T\omega)\cdot\omega\|^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left[x_i - (x_i^T \omega) \cdot \omega \right] \left[x_i - (x_i^T \omega) \cdot \omega \right]$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left[x_i^T x_i - (x_i^T \omega)^2 = \cancel{(x_i^T \omega)^2 + (x_i^T \omega)^2} \right]$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left[x^T x_i - (x_i^T \omega)^2 \right]$$

↑ constant, irrelevant

$$g(x) = \frac{1}{n} \sum_{i=1}^{n} - (x_i^T \omega)^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} (\omega^T x_i)(x_i^T \omega)$$

$d \times 1 \quad 1 \times d \quad d \times 1 \quad 1 \times d$

$$= \frac{1}{n} \sum_{i=1}^{n} \omega^T \left[x_i x_i^T \right] \omega$$

$$= \omega^T \left[\frac{1}{n} \sum_{i=1}^{n} (x_i x_i^T) \right] \omega$$

↓ co-variance matrix

$= w^T C w$

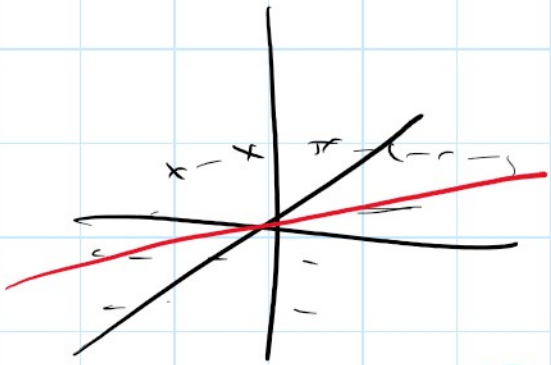
* w is eigenvector corresponding to the max eigen value of C
* $C \rightarrow$ covariance matrix

→ The above eigen vector is an optimization problem.

L 1.5 - Representation learning p3

→ Is this compression satisfactory in all cases.

Eg:- 3-D where points on a plane.



line lies on the plane

- In 2-D cases, we are taking this as error
- In 3-D when points are on a plane, error also has some information

→ algorithm gives line but we also want to retrieve that it is in 2-D.

to capture that it is in 2-D.

→ If we see the error vectors, they lie along a single direction

Procedure

$x \in \mathbb{R}^d$

↓ Find ω

$(x^T\omega)\cdot\omega$

↓ error

$x - (x^T\omega)\cdot\omega$

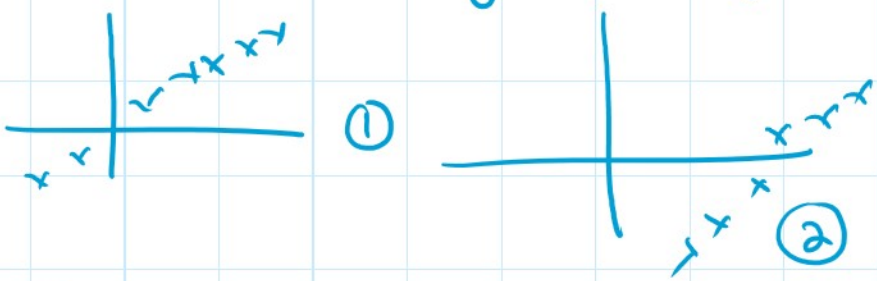
↓

has some information within

Possible algorithm:

- Input: $\{x_1 \dots x_n\}$ $x_i \in \mathbb{R}^d$
  - → $u = \frac{1}{n}\sum_{i=1}^{n} x_i$; $x_i = x_i - u$
  - → Find "best" line $\omega_1 \in \mathbb{R}^d$
  - → $x_i = x_i - (x_i^T\omega)\cdot\omega$
  - → Repeat to obtain ω_2
  - → ! Issue.
    - → The above system works for data about origin i.e $y = mx$.

data about origin i.e $y = mx$.



→ ② doesn't work.

→ so solution for that is to calculate mean & remove it

Questions for the above systems :-

→ How to solve $\max \omega^T C \omega$?

→ How many times to repeat?

→ Where exactly is compression happening?

→ What representations are we learning?

7.6- PCA - I

$$D = \{x_1 \dots x_n\} \quad x_i \in \mathbb{R}^d$$

$$\omega_1 = \underset{\|\omega\|}{\operatorname{argmax}}\ \omega^T C \omega \qquad \omega_2 = \underset{\|\omega_2\|^2 = 1}{\operatorname{argmax}}\ \omega^T C' \omega$$

$$C = \frac{1}{n} \sum_{i=1}^{n} x_i x^T \qquad C' = \frac{1}{n} \sum_{i=1}^{n} x_i' x_i'^T$$

$$x_i' = x_i - (x_i^T \omega_1)\, \omega_1$$

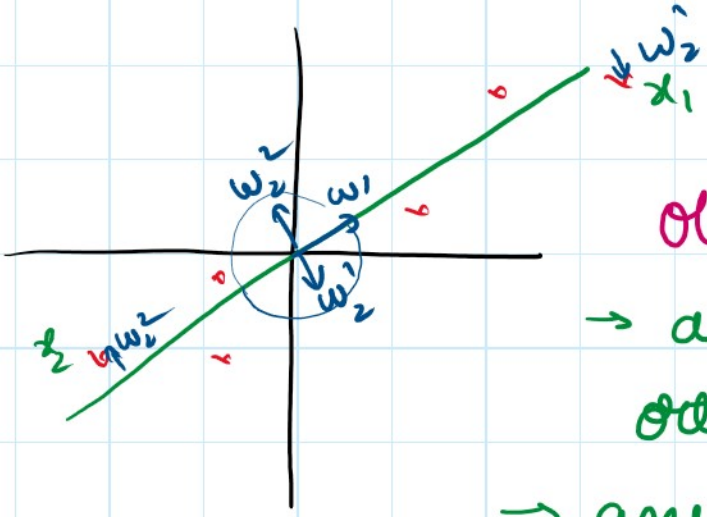
→ Best fit line direction

→ Error line direction (ω_2)

→ Best fit line direction w_1

→ Error line direction w_2

Question: what can we say about w_1 & w_2



$w_2 \perp w_1$

Observation

→ all residues are orthogonal to w_1

→ any line which minimizes sum of errors w.r.t. residues must be orthogonal to w_1

$$w_2^T w_1 = 0$$

→ going into higher dimensions, we can get residuals of residuals & so on.

By continuing,

$\{w_1 \dots w_d\}$ → orthonormal vector.

Properties:

- $\|w_k\|^2 = 1 \quad \forall \; k$. [for all k]
- $w_i^T w_j = 0 \quad \forall \; i \neq j$

- $\omega_i^T \omega_j = 0 \quad \forall \; i \neq j$

Residue after Round - I

$$\{ (x_1 - (x_1^T \omega_1)\omega_1), \ldots \quad x_i \in \mathbb{R}^d$$
$$(x_n - (x_n^T \omega_1)\omega_1) \}$$

- $\omega_2 \rightarrow$ Best fit line
- $\omega_1^T \omega_2 = 0$

Residues after Round 2.

$$\left[x_i - (x_i^T \omega_1)\omega_1 - \left[(x_i - (x_i^T \omega_1)\omega_1)\omega_2 \right] \omega_2 \right]$$

$$\Downarrow$$

$$\left[x_i - (x_i^T \omega_1)\omega_1 - \left(x_i^T \omega_2 - (x_i^T \omega_1)\omega_1^T \omega_2 \right) \omega_2 \right.$$

$\omega_1^T \omega_2 = 0 \quad \omega_1 \perp \omega_2$

$$\Rightarrow \left[x_i - (x_i^T \omega_1)\omega_1 - (x_i^T \omega_2)\omega_2 \right]$$

Residues after d rounds.

$$\forall i \quad x_i - \left((x_i^T \omega_1)\omega_1 + (x_i^T \omega_2)\omega_2 + \ldots \right.$$
$$\left. + (x_i^T \omega_d)\omega_d \right)$$

$$= x_i - \sum_{j=1}^{d} (x_i^T \omega_j)\omega_j$$

$$x_i - \sum_{j=1}^{d} (x_i^T w_j) w_j$$

If you run for d rounds, the residual will be 0 because the points will align

$$x_i - \left(\sum_{j=1}^{d} (x_i^T w_j) w_j\right) = 0 \quad \in \mathbb{R}^d$$

$$\Rightarrow x_i = \sum_{j=1}^{d} (x_i^T w_j) w_j$$

What have we gained?

- If data lives in a lower dimensional subspace, residues become 0 much earlier than d rounds.

Example:- 100d but residuals 0 after 3 rounds

$$D = \{x_1 \cdots x_n\} \in \mathbb{R}^{100}$$

$$\forall i \quad x_i = (x_i^T w_1) w_1 + (x_i^T w_2) w_2 + (x_i^T w_3) w_3$$

Rep

$\{w_1, w_2, w_3\}$

for full dataset

Co-efficients

$x_i \to [x_i^T w_1 \;\; x_i^T w_2 \;\; x_i^T w_3]$

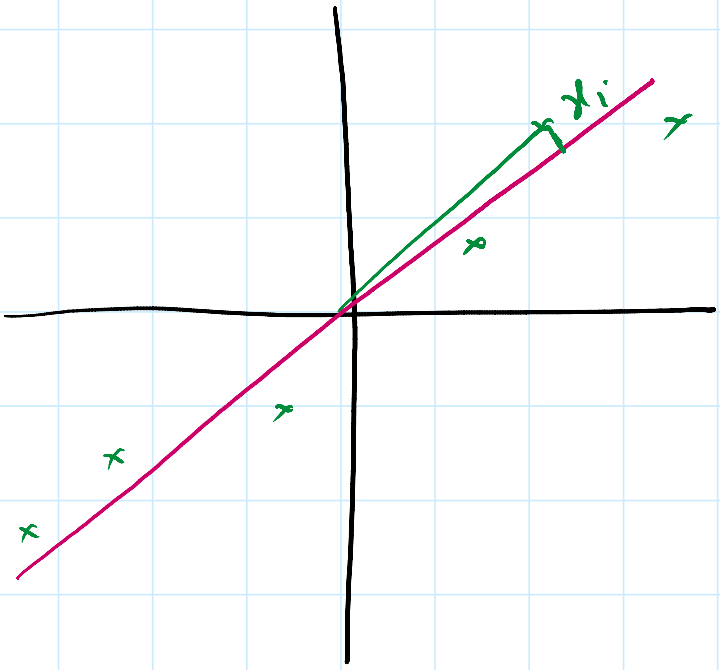
for each point

original :- $100 \times n$ after :- $3 \times 100 + 3n$.

$\to d \times n \quad \Rightarrow d \times k + k \times n$.

What if data "approximately" fits in a low-dimensional space?

→ use a tolerance.



For any $W \in R^d$ s.t $\|w\|^2 = 1$

For all

$\|x_i\|^2$

$= \|x_i - (x_i^T w) w\|^2$

$+ \|(x_i^T w) w\|^2$

$$\Rightarrow \underbrace{\frac{1}{n}\sum_{i=1}^{n} \|x_i\|^2}_{\downarrow} = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \|x_i - (x_i^T w) w\|^2}_{\downarrow} + \frac{1}{n}\sum_{i=1}^{n} \|(x_i^T w) w\|^2$$

for any w avg length of dataset

avg error, minimize

$$+\frac{1}{n}\sum_{i=1}^{n}\|(x_i^T w)w\|^2$$

maximize ①

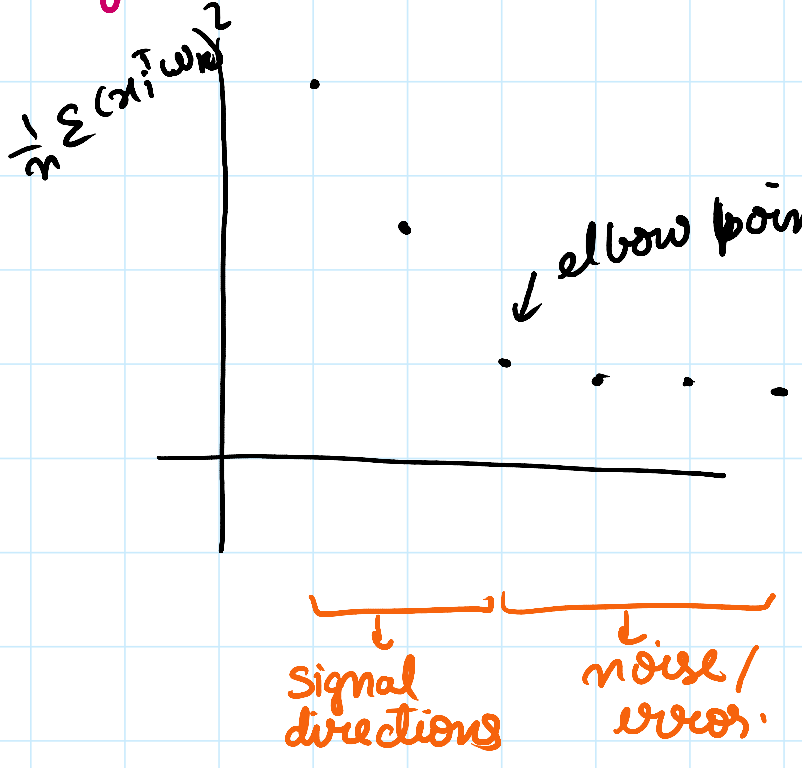
$$① \rightarrow \frac{1}{n}\sum_{i=1}^{n}\|(x_i^T w)w\|^2$$

$$= \|(x_i^T w)^2 w^2\|^1$$

$$= \frac{1}{n}\sum_{i=1}^{n}\|(x_i^T w)\|^2$$

★ Larger the value of $\frac{1}{n}\sum_{i=1}^{n}(x_i^T w)^2$ the better the fit

graphing it: –



- We choose threshold.
  - Say 95% represented by k, we take only k

11.7 - PCA - II

Referring back to optimization problem 11.4

$$\underset{\omega \,|\omega|^2=1}{\text{argmax}}\ \omega^T C \omega \qquad C = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$$

C → covariance matrix

Soln:

- ω_1 → eigenvectors corresponding to largest eigen value C

  [Hilbert min-max problem.

- $\{\omega_1, \ldots \omega_d\}$ The eigenvectors of C forms orthonormal basis
- ω_k → best line one can obtain in round k.

What do eigen values mean?

- we know

$$C\omega_1 = \lambda_1 \omega_1$$

$$\omega_1^T C \omega_1 = \omega_1^T (\lambda \omega_1)$$

$$\omega_1^T C \omega_1 = \omega_1^T (\lambda \omega_1)$$
$$\omega_1^T C \omega_1 = \lambda_1$$

$$\Rightarrow \lambda_1 = \omega_1^T C \omega_1 = \omega_1^T \left(\frac{1}{n}\sum_{i=1}^{n} x_i x_i^T\right)\omega_1$$

$$= \frac{1}{n}\sum_{i=1}^{n} (x_i^T \omega_i)^2$$

$$\boxed{\lambda_1 = \frac{1}{n}\sum_{i=1}^{n} (x_i^T \omega_1)^2}$$ ← will always +ve.

★ $\lambda_k(C)$ graph vs k gives same elbow graph.

★ eigenvalues are non-negative

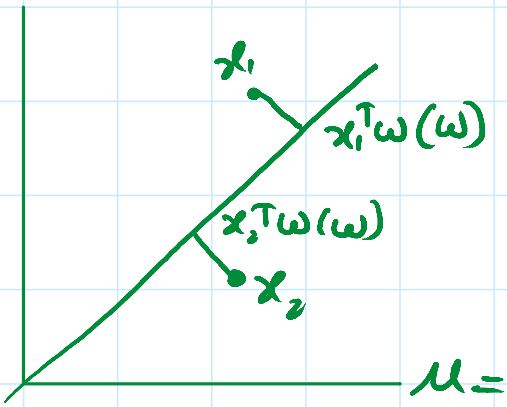
★ Rule of Thumb- (for no. of dimensions)

$$\frac{\sum_{i=1}^{d} \lambda_i(C)}{\sum_{i=1}^{d} \lambda_i(C)} \geq \text{Threshold.}$$

[0.95 in practice]

Taking a collection:-

For a ω, taking collection of points



$\{x_1^T\omega, x_2^T\omega \ldots x_n^T\omega\}$

Avg:-

$$\mu = \frac{1}{n}\sum_{i=1}^{n}(x_i^T\omega)$$

$$\mu = \left(\frac{1}{n}\sum_{i=1}^{n}x_i\right)^T\omega = (0^T\omega)\,\omega$$

$$\Downarrow$$

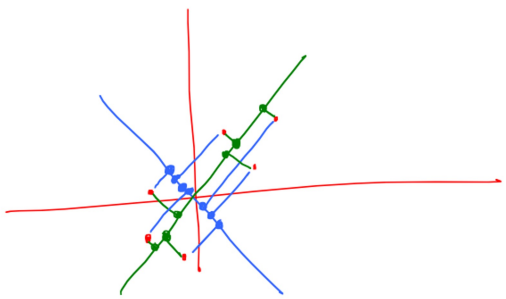
$$0$$

$\mu = 0$ [for centred dataset]

Variance:- $\frac{1}{n}\sum_{i=1}^{n}(x_i^T\omega - \text{mean})^2$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i^T\omega)^2$$

☆ Error minimization on centred dataset → variance maximization



Want directions where Projections don't "crowd up" ↵ i.e., Variance is not small.

Eg:- Refer pdf.

basically that the components are de correlated

PCA → $\{\omega_1 \dots \omega_R\}$

→ Means that

→ only k dimensions used to represent d dimensions

→ PCA → dimentionality reduction "

★ PCA finds combination of features that are de correlated.

→ Loosely speaking independent of each other

Assignment: -

2) x on L

$x = (2, 5)$

$\omega = [1, 1]$

$$\frac{x^T \omega}{\|\omega\|^2} \omega \quad \begin{bmatrix} 2 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\frac{7}{2} = \begin{bmatrix} 3.5 \\ 3.5 \end{bmatrix}$$

$\sqrt{2 \times 12.25} = \sqrt{24.5}$

$\sqrt{4.5}$

$d \times m$ $\quad$ $d \times k + k \times m$

30000 $\quad$ $120 + 4000$

30000 $\quad$ 4120

$\frac{25880}{30000} = 86.26$

$x_i = \begin{bmatrix} x \\ y \end{bmatrix}$

$x_i x_i^T = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} x & y \end{bmatrix}$

$\begin{bmatrix} x^2 & xy \\ xy & y^2 \end{bmatrix}$

$a \quad 4 \quad 1 \quad 4$ $\quad$ $0 \quad -2 \quad -2$

$a \quad 4 \quad 1 \quad 4$ $\quad$ $\cdots$

$$\begin{vmatrix} 0 & -2 & -2 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{vmatrix} \qquad \begin{vmatrix} 0 & 1 & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{vmatrix}$$

$$36 \qquad -4$$
$$-4 \qquad 2$$