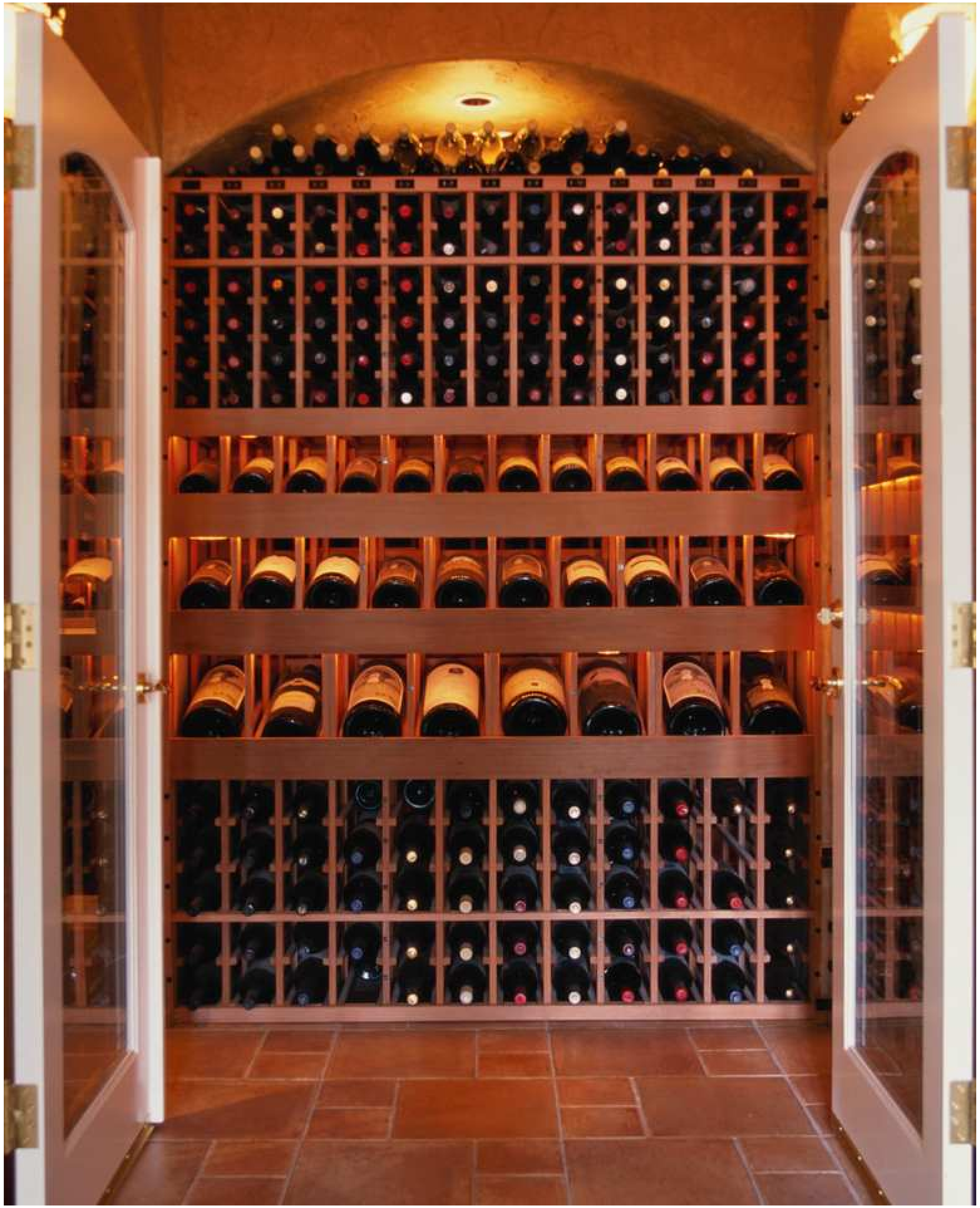
Principle Component Analysis

김형욱(hyounguk1112@gmail.com)

PCA



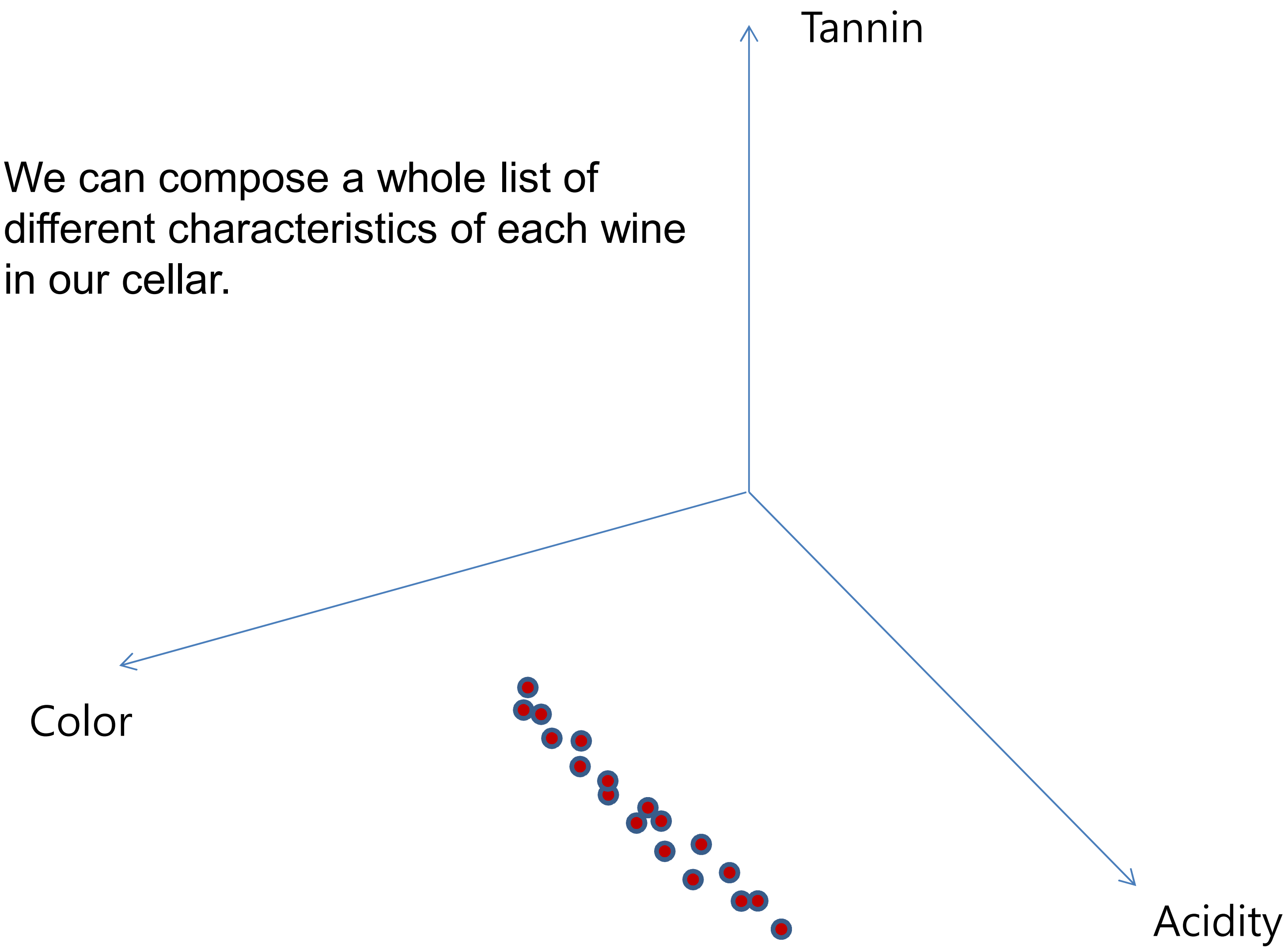
What characteristics can be used to describe a wine?


THE COLOR OF WINE
wide watery rim
transparent
[light-bodied]
Light-bodied red wines tend to have low tannin and high acidity.
e.g. Pinot Noir, Gamay
medium rim width
semi-transparent opaque core
[medium-bodied]
Medium-bodied red wines tend to have moderate tannin and medium acidity.
e.g. Tempranillo, Merlot and Sangiovese
compact rim width
opaque
[full-bodied]
Full-bodied red wines tend to have high tannin and low acidity.
e.g. Syrah, Malbec and Cabernet Sauvignon
compact rim width
rich color more opaque violet - red
young wine
A young wine is at its peak level of tannin, acidity and fruit aroma.
high color variation
wide rim
dull color transparent red - orange
old wine
Wine loses acidity and tannin over time but gains bottle-aged aromas of spice.
pale yellow-green silver glow
[light-bodied]
Light bodied white wines tend to have high acidity and are best enjoyed ice- cold.
e.g. Pinot Grigio, Albariño, Muscadet
pale gold platinum glow
[medium-bodied]
Medium bodied white wines tend to have moderate acidity. Most white wines fall into this category.
e.g. Sauvignon Blanc, Trebbiano, Chenin Blanc
rich yellow-ochre copper glow
[full-bodied]
Full bodied white wines have lower acidity and rich creamy flavors.
e.g. Chardonnay, Viognier, Semillon
saturated color bright glow yellow to green
young wine
Most white wines are meant to be enjoyed young with higher acidity and fresh flavors.
dull color more faded yellow to brown
old wine
Aging is best suited for full-bodied and sweet wines. It lowers acidity but adds tertiary nutty aromas.
WINE FOLLY
Designed by winefolly.com

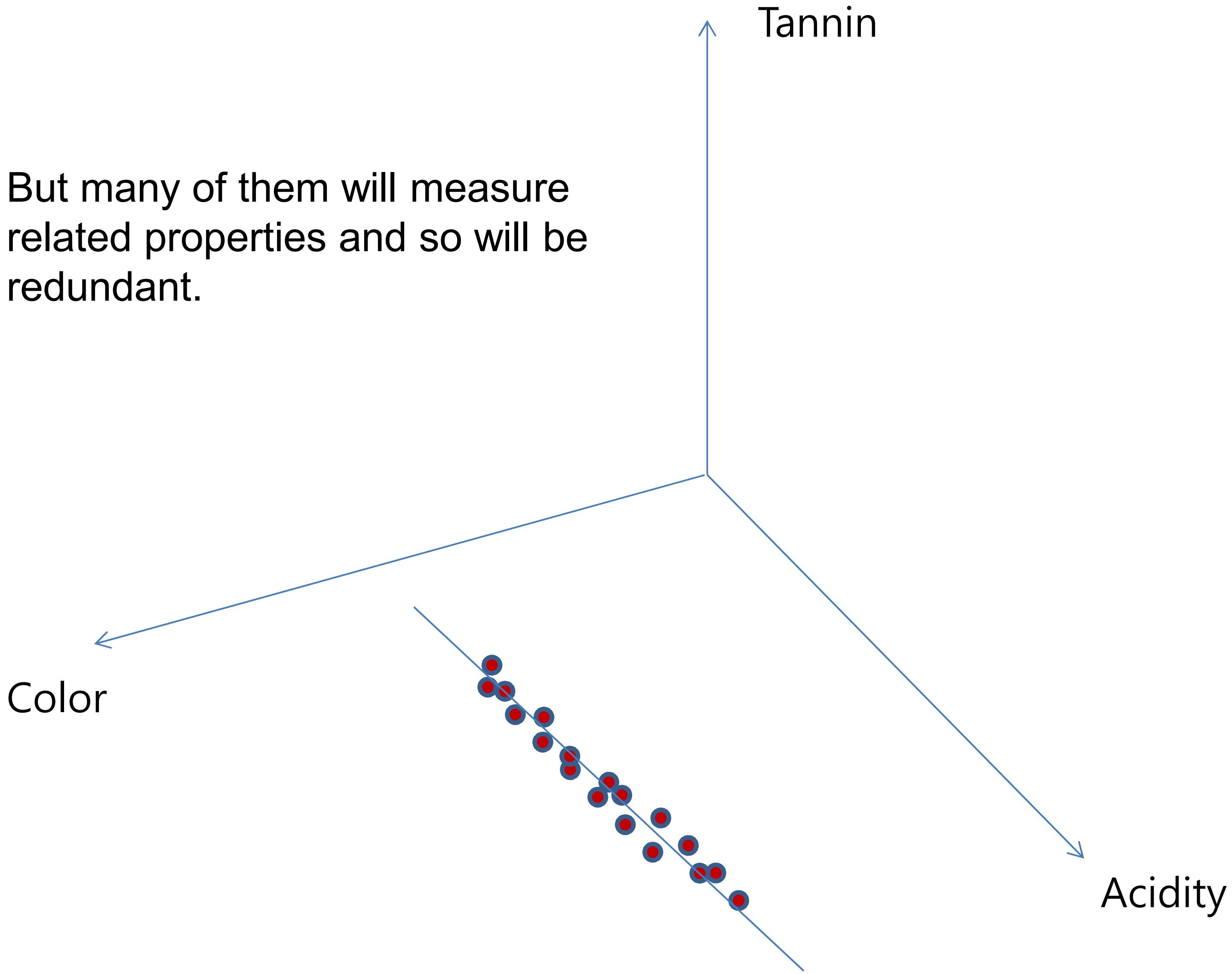
Like this, you can say

- How strong color It is.
- How old It is
- Acid level
- Amount of tannin
- And so on.

We can compose a whole list of different characteristics of each wine in our cellar.



But many of them will measure related properties and so will be redundant.

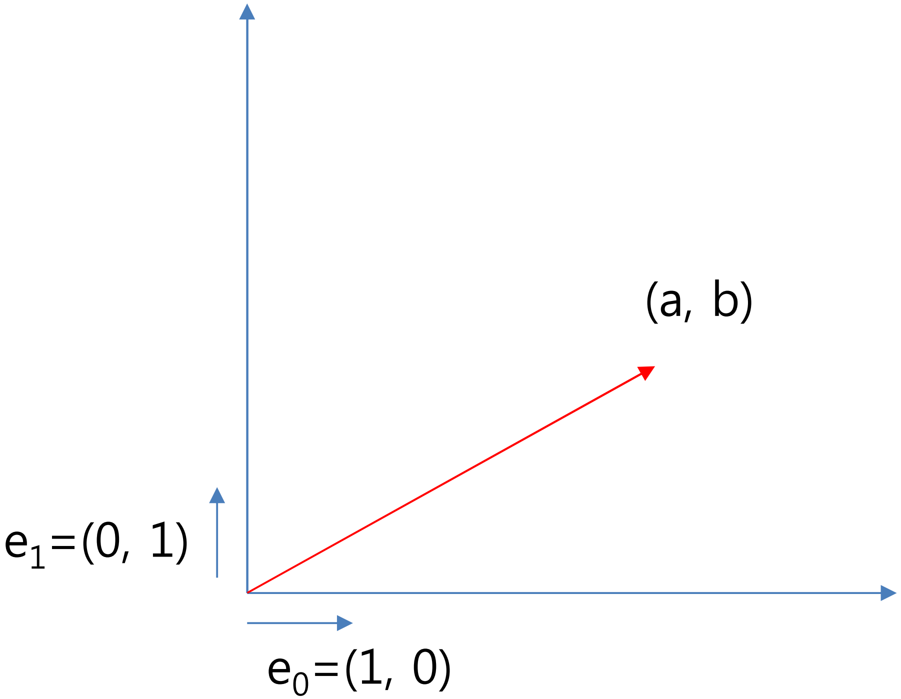


we should be able to **summarize each wine with less characteristics!**
This is what PCA does.

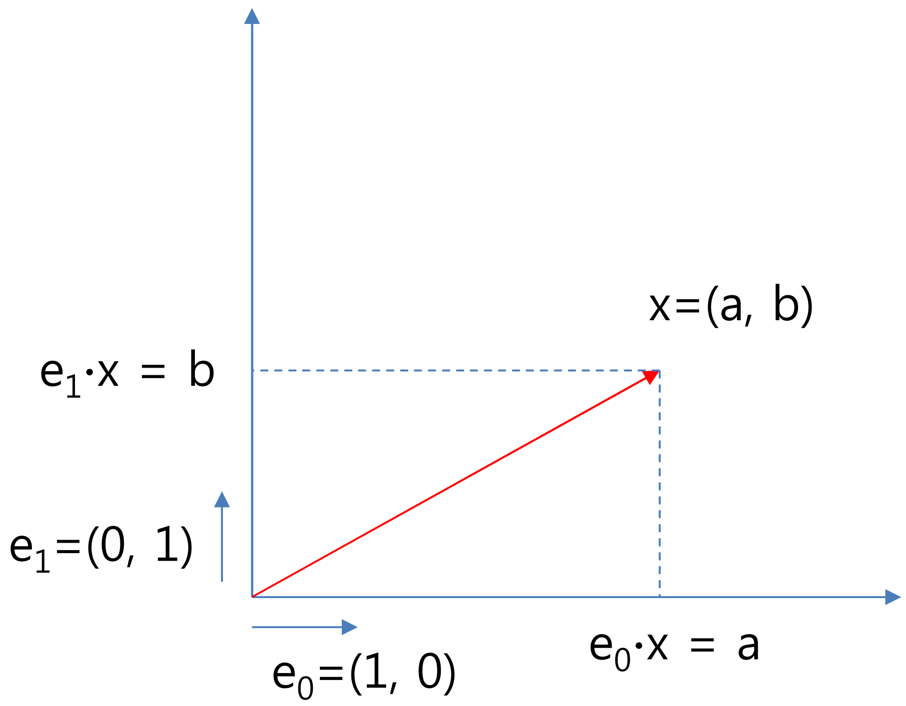
So this PCA thing checks what characteristics are redundant and discards them?

Excellent question! **But, no.**
PCA finds the best possible new characteristics, the ones that summarize the list of wines as well as only possible

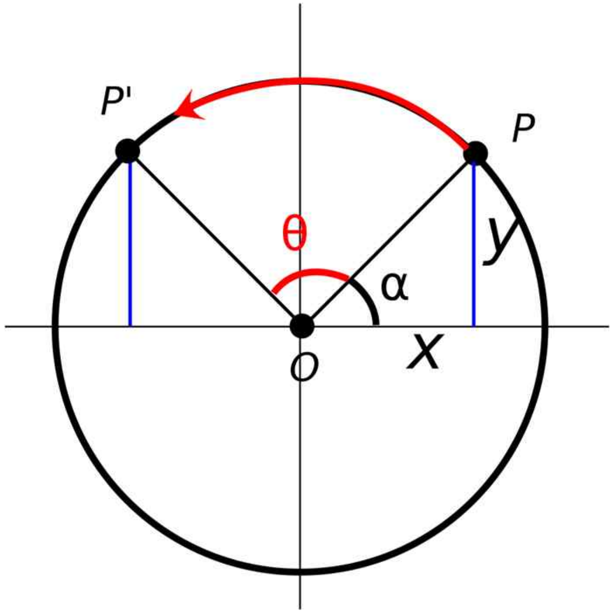
Let's go to the world of Linear Algebra for a second.



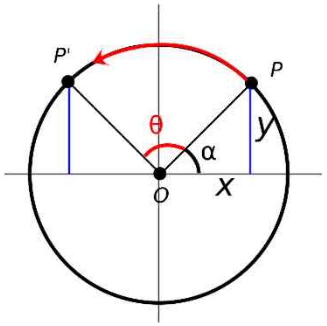
e_1 and e_2 are basis.
The inner product between a vector and a base indicates the size(scalar) of vector on the axis of the base.



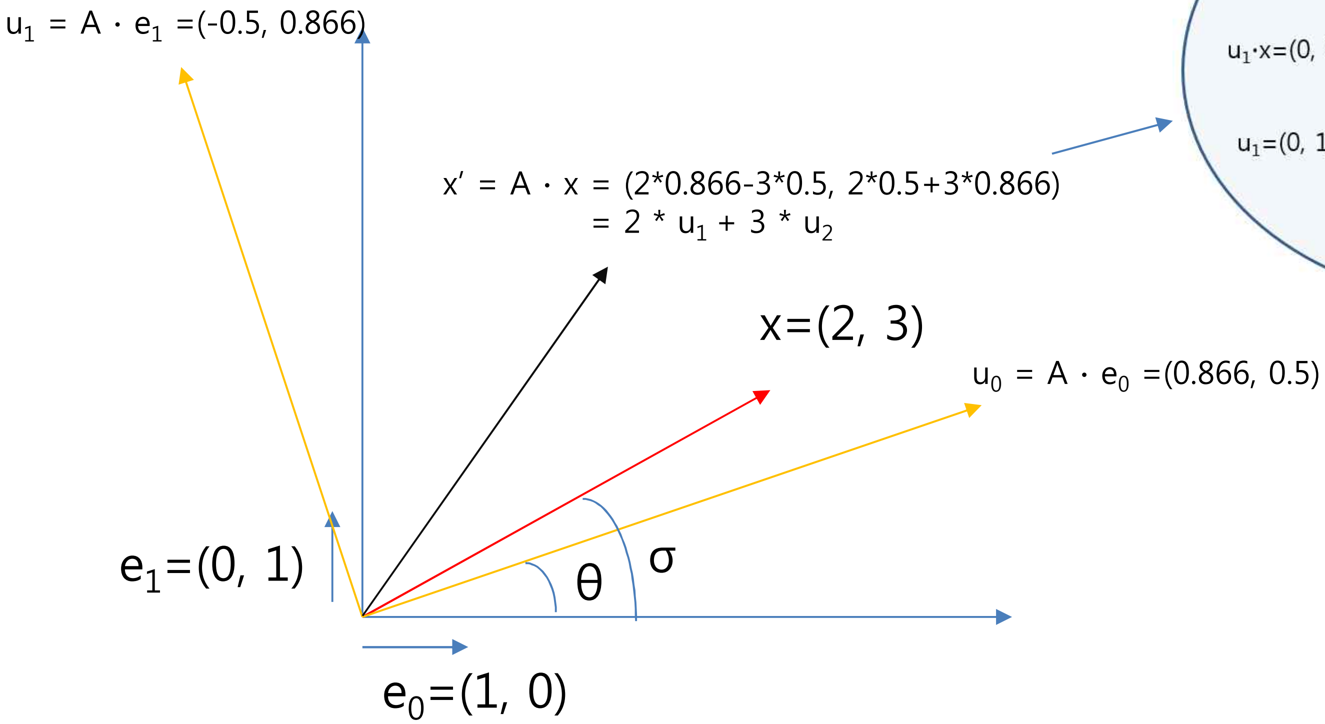
The transformation of rotation

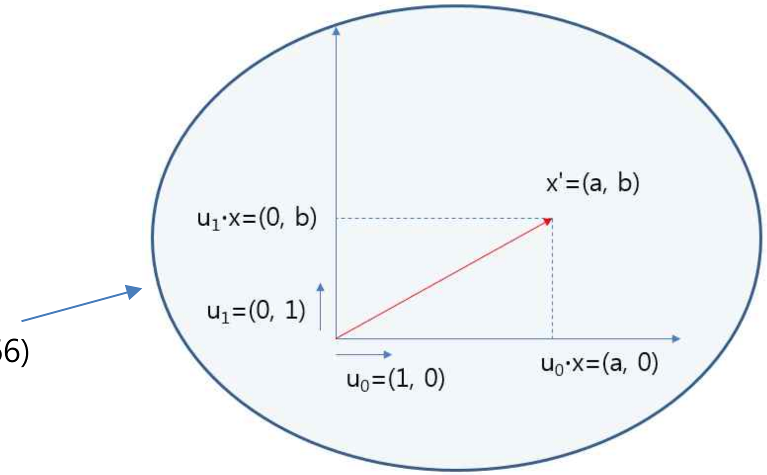


$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix}$$

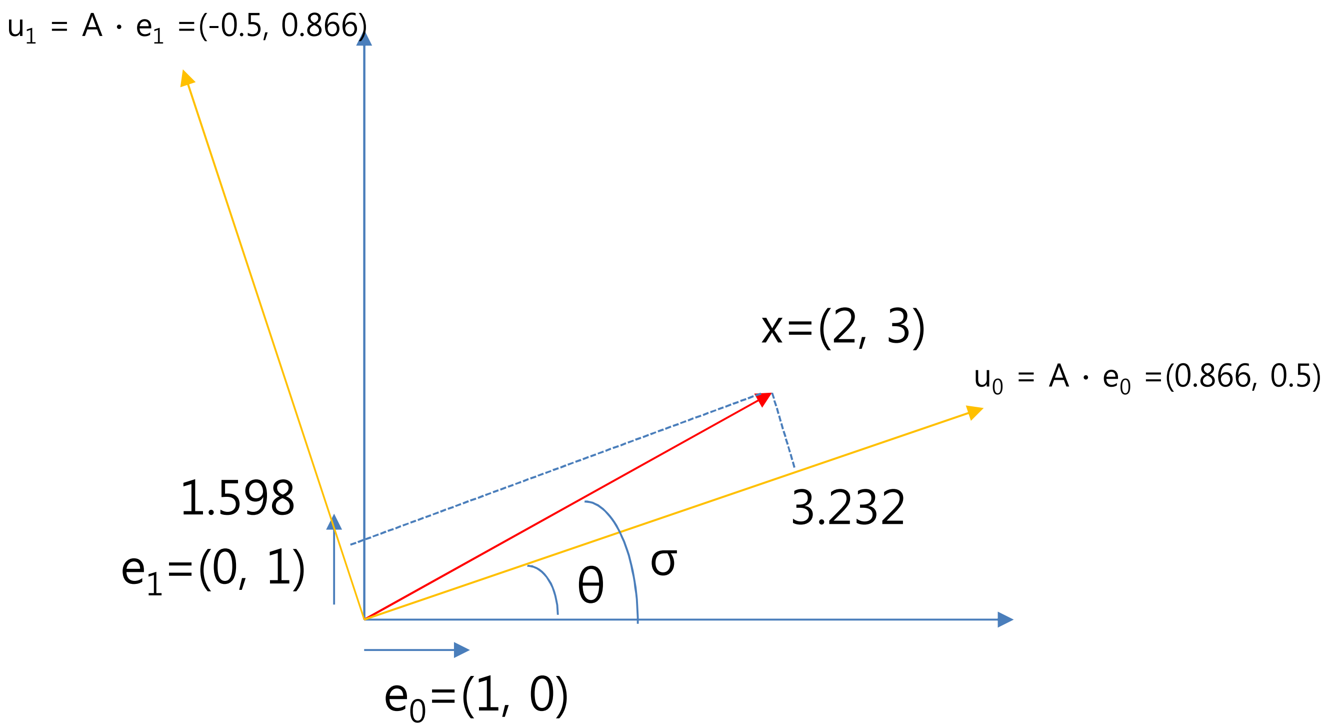


$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x' \\ y' \end{pmatrix} \Rightarrow \Theta = pi/6,\ A = \begin{pmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{pmatrix}$$





만약에 새로 찾은 축에 대해서 기존의 벡터를 나타내 보면 어떨까?



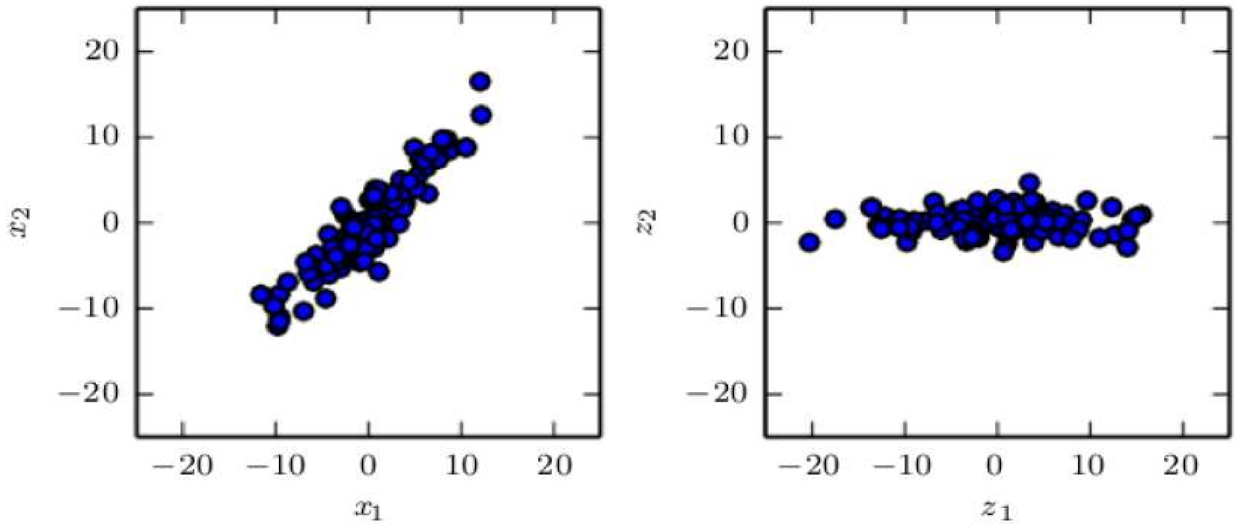
$x = (2, 3) = 3.232 * u_0 + 1.598 * u_1$ / 회전된 좌표계에서는 (3.232, 1.598)

무슨 뜻?
새로운 vector space에서 data x는 (3.232, 1.598)로 나타낼 수 있다.
x뿐만 아니라 **어떤 데이터 분포 X에 포함된 모든 데이터를 새로운 좌표계에서 나타낼 수 있으며**, 원래 차원으로 복원할 수도 있다.
만약 새로운 좌표축이 분석하기 좋거나, 데이터 압축에 도움이 된다면?

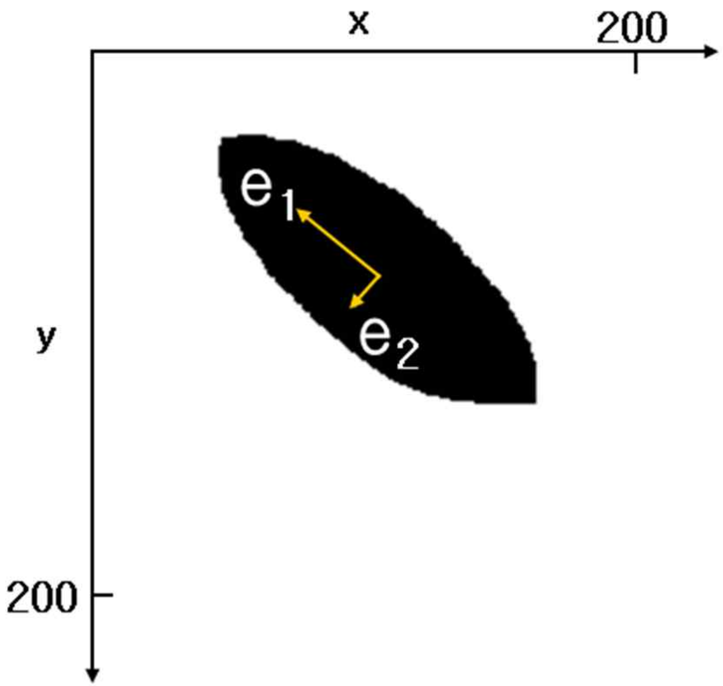
Principal Component Analysis

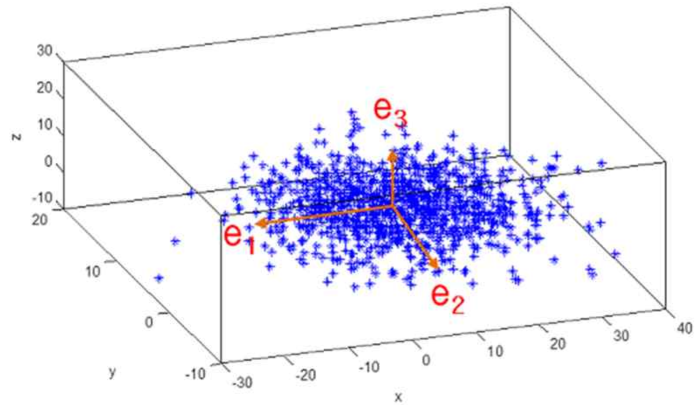
- PCA learns a linear projection that aligns the direction of greatest variance with the axes of the new space. It also learns a representation whose elements have no linear correlation with each other(independent).

- The important thing is preserving as much of the information in the data as possible

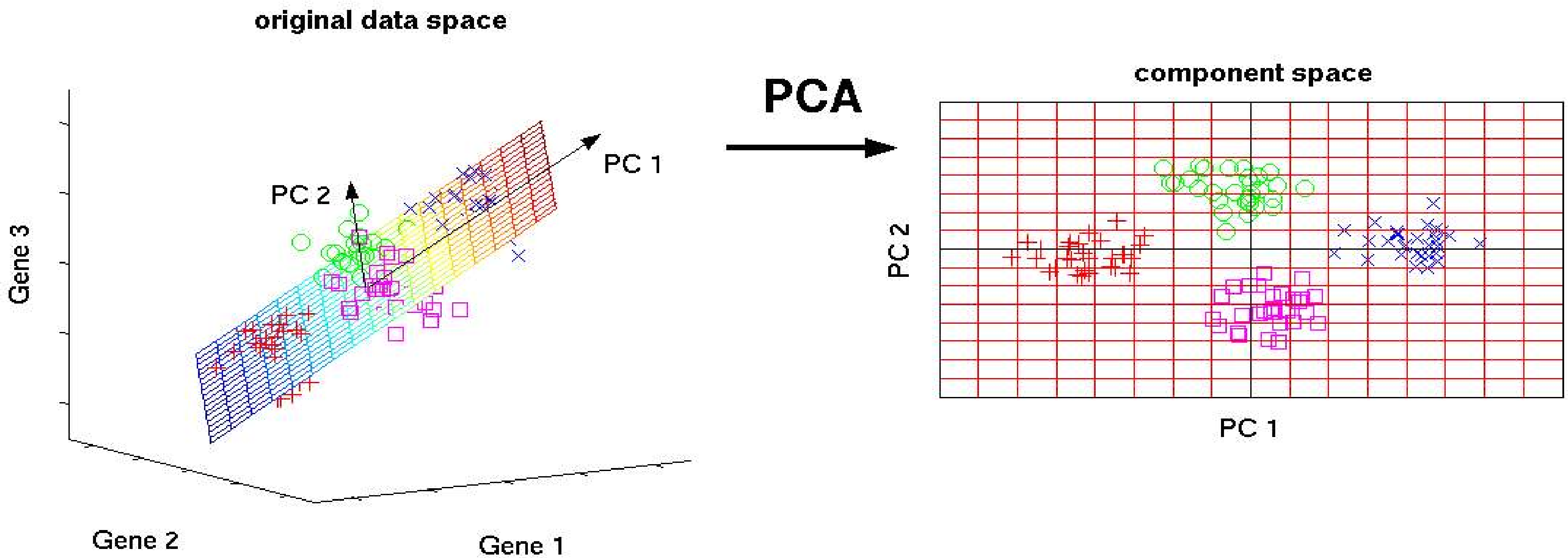


How to find principal components?

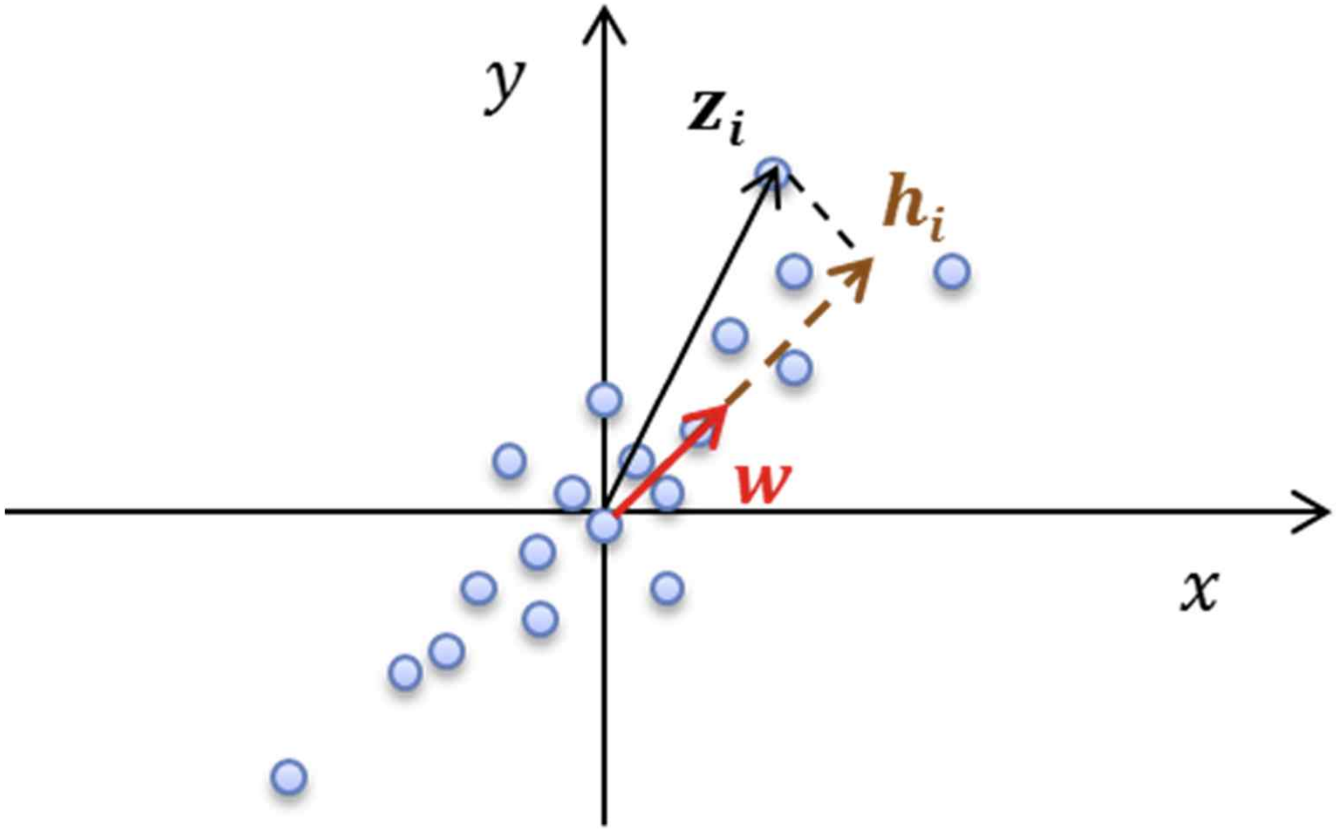




First principal component = the vector which has direction of the largest variance
Next principal component = orthogonal to PC1 +Largest variance


original data space
PCA
component space
PC 1
PC 2
Gene 3
Gene 2
Gene 1
PC 2
PC 1

Finding a vector to maximize variance of data distribution



Finding a vector to maximize variance of data distribution

$$\sigma_w^2 = \frac{1}{n}\sum_i (z_i \cdot w)^2 - \left(\frac{1}{n}\sum_i (z_i \cdot w)\right)^2$$

Mean zero

Correlation and covariance of Each feature of data are the same by mean shift(To zero)

$$= \frac{1}{n}\sum_i (z_i \cdot w)^2$$

$$= \frac{1}{n}(Zw)^T(Zw)$$

$$= \frac{1}{n} w^T Z^T Z w$$

$$= w^T \frac{Z^T Z}{n} w$$

$$= w^T C w$$

lagrange multiplier

$$u = w^T C w - \lambda(w^T w - 1)$$

$$\frac{\partial u}{\partial w} = 2Cw - 2\lambda w = 0$$

$$Cw = \lambda w$$

So PCA algorithm is the problem to find eigen-vectors of Covariance matrix

A calculation of PCA

- covariance
  - $\mathrm{cov}(x,y) = E[(x-m_x)(y-m_y)]$
- covariance matrix
  - $x=[x_1,\ldots,x_n]^T$: sample data, n차원 열벡터
  - $C = E[(x-m_X)(x-m_X)^T]$: $n \times n$ 행렬
  - $\langle C \rangle_{ij} = E[(x_i-m_{xi})(x_j-m_{xj})^T]$: i번째 성분과 j번째 성분의 공분산
  - C is real and symmetric $C = \begin{pmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{pmatrix}$

A calculation of PCA

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(x,y) & cov(y,y) \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{n}\sum(x_i - m_x)^2 & \frac{1}{n}\sum(x_i - m_x)(y_i - m_y) \\ \frac{1}{n}\sum(x_i - m_x)(y_i - m_y) & \frac{1}{n}\sum(y_i - m_y)^2 \end{pmatrix}$$

A calculation of PCA

PCA

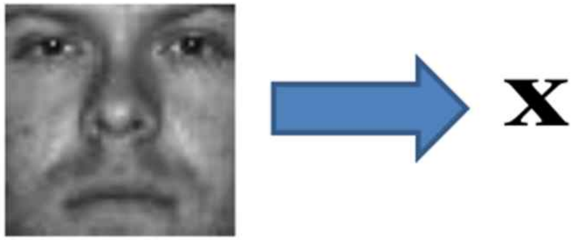
- C : covariance matrix of x
- $C = P\Sigma P^T$ (P: orthogonal, Σ: diagonal)

$$C = \begin{pmatrix} \boxed{e_1} & \cdots & \boxed{e_n} \end{pmatrix} \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} \begin{pmatrix} \boxed{e_1^T} \\ \vdots \\ \boxed{e_n^T} \end{pmatrix}$$

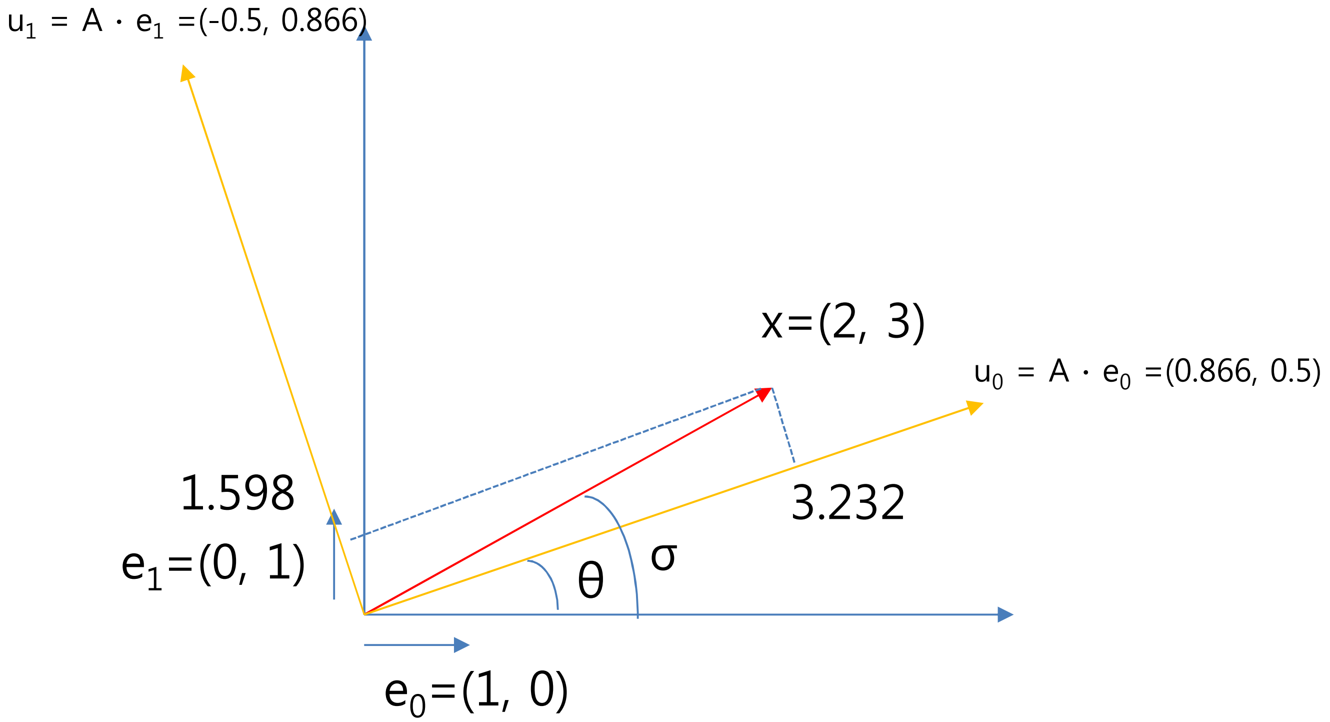
- P : n×n orthogonal matrix
- Σ : n×n diagonal matrix
- $Ce_i = \lambda_i e_i$
  - e_i : eigenvector of C, direction of variance
  - λ_i : eigenvalue, e_i 방향으로의 분산
  - $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$
- e_1: 가장 분산이 큰 방향
- e_2: e_1에 수직이면서 다음으로 가장 분산이 큰 방향
- e_k: $e_1, \ldots, e_{k-1}$에 모두 수직이면서 가장 분산이 큰 방향

Application for face detection

- Simple Idea for Face Detection
- 1. Treat each window in the image like a vector

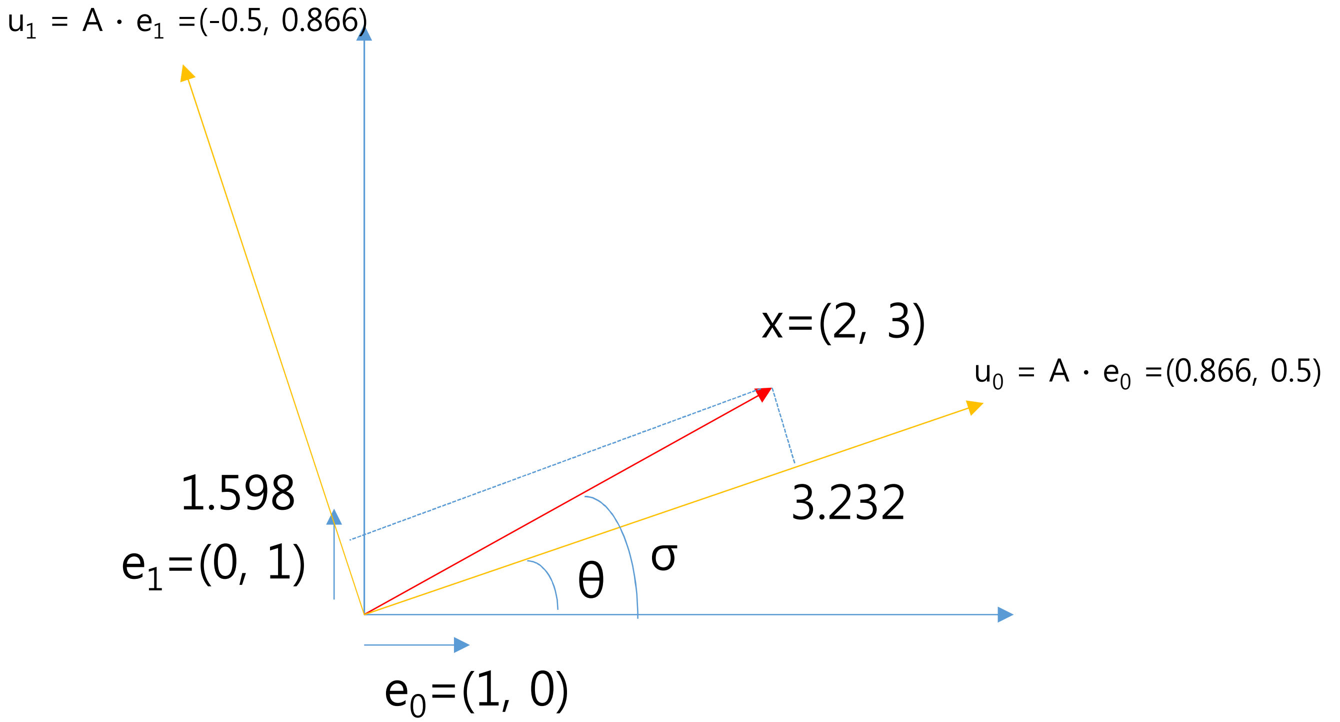


- 2. Eigenface idea: construct a low-dimensional lin ear
- subspace that contains most of the face images
- possible (possibly with small errors)



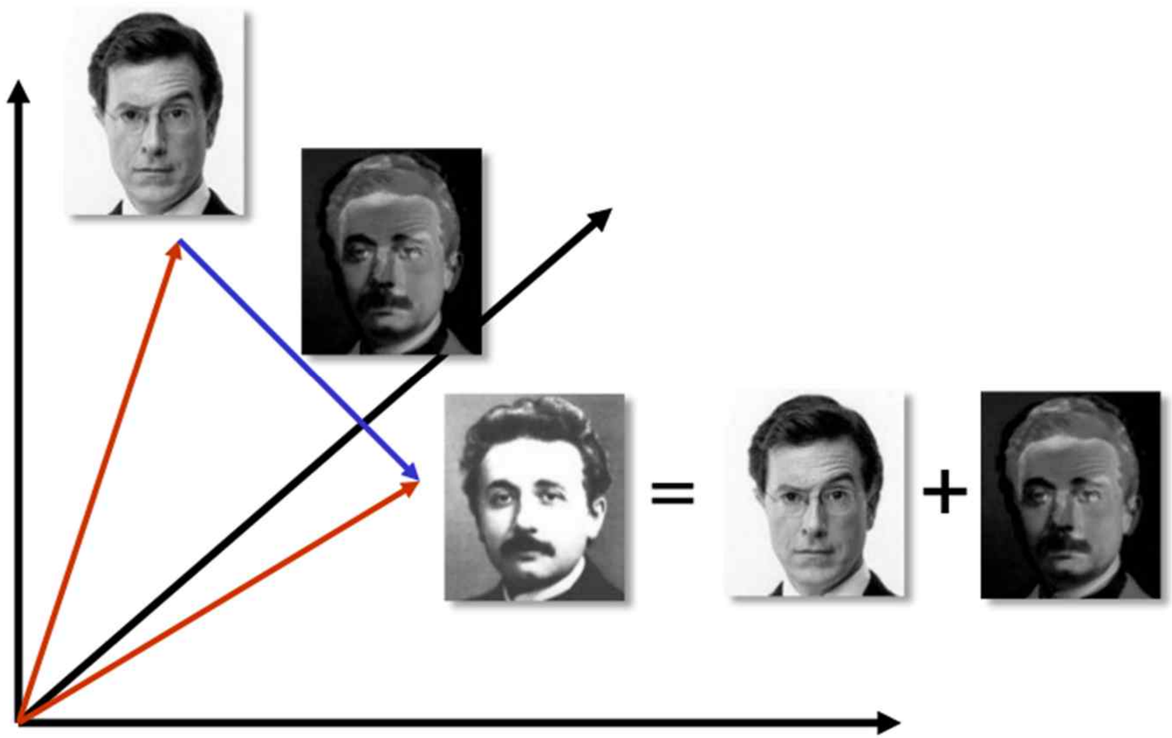
$x = (2, 3) = 3.232 * u_0 + 1.598 * u_1$ / 회전된 좌표계에서는 (3.232, 1.598)

예를 들어 100x100 image는 10000 feature를 가진 data.
마찬가지로 Eigenface들의 linear combination으로
기존의 얼굴 이미지를 표현할 수 있다.



$x = (2, 3) = 3.232 * u_0 + 1.598 * u_1$ / 회전된 좌표계에서는 (3.232, 1.598)

Space of faces



Reconstruction

- For a subspace with the orthonormal basis of size k $\mathrm{V_k} = \{v_0, v_1, v_2, \dots v_k\}$, the best reconstruction of *x* in that subspace is:

$$\hat{x}_k = (x \cdot v_0)v_0 + (x \cdot v_1)v_1 + \cdots + (x \cdot v_k)v_k$$

  – If x is in the span of V_k, this is an exact reconstruction

  – If not, this is the projection of *x* on *V*

  - $\hat{x}_k = (x \cdot v_0)v_0 + (x \cdot v_1)v_1 + \cdots + (x \cdot v_k)v_k$

- Squared reconstruction error: $(\hat{x}_k - x)^2$

Reconstruction

After computing eigenfaces using 400 face images from ORL face database

Thank you