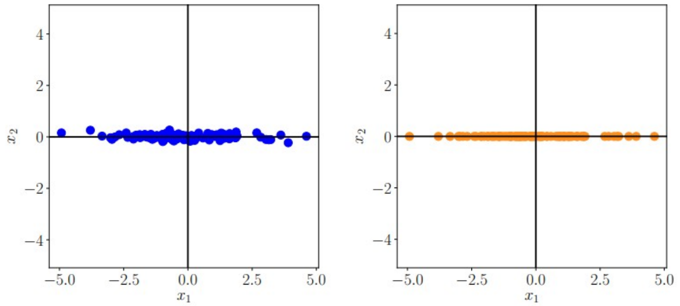
Chapter 10: Dimensionality Reduction with Principal Component Analysis

Mathematics for Machine Learning

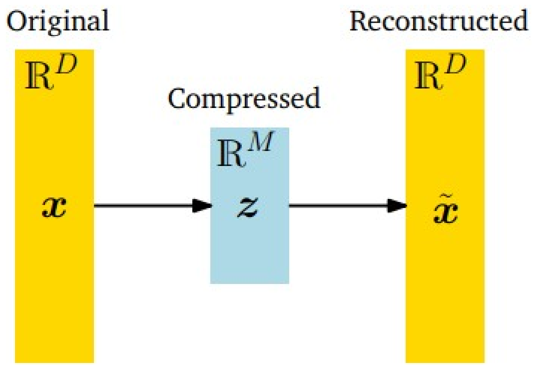
San Diego Machine Learning
Liam Barstad

Principal Component Analysis (PCA)

Linear method for **dimensionality reduction**



(a) Dataset with x_1 and x_2 coordinates. (b) Compressed dataset where only the x_1 coordinate is relevant.



$$\boldsymbol{z}_n = \boldsymbol{B}^\top \boldsymbol{x}_n \in \mathbb{R}^M$$

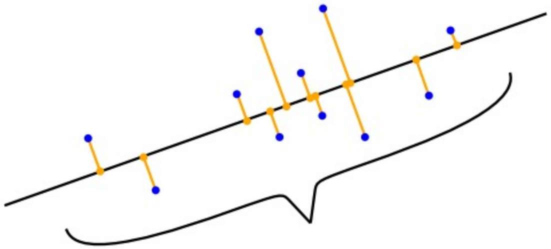
B = orthonormal basis (ONB) for the projection of *x* to *z*, defining the **principal subspace** *U*

How do we minimize compression loss?

- Maximize variance in *z*
- Minimize reconstruction loss

Maximizing Variance Perspective

Finding the values for *B* (the principal subspace ONB) and *z* (the coordinates of the projection) that maximize variance retains the most information about *X*



Variance for first z coordinate: $V_1 := \mathbb{V}[z_1] = \frac{1}{N}\sum_{n=1}^{N} z_{1n}^2$

Substitute for covariance matrix:

$$V_1 = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{b}_1^\top \boldsymbol{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{b}_1^\top \boldsymbol{x}_n \boldsymbol{x}_n^\top \boldsymbol{b}_1$$

$$= \boldsymbol{b}_1^\top\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\right)\boldsymbol{b}_1 = \boldsymbol{b}_1^\top \boldsymbol{S}\boldsymbol{b}_1\,,$$

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\,.$$

Re-write as optimization problem:

$$\max_{\boldsymbol{b}_1} \boldsymbol{b}_1^\top \boldsymbol{S}\boldsymbol{b}_1$$

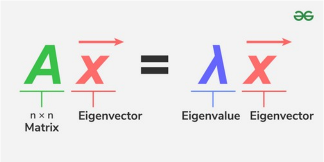
$$\text{subject to } \|\boldsymbol{b}_1\|^2 = 1\,.$$

$$\mathfrak{L}(\boldsymbol{b}_1, \lambda) = \boldsymbol{b}_1^\top \boldsymbol{S}\boldsymbol{b}_1 + \lambda_1(1 - \boldsymbol{b}_1^\top \boldsymbol{b}_1)$$

$$\frac{\partial \mathfrak{L}}{\partial \boldsymbol{b}_1} = 2\boldsymbol{b}_1^\top \boldsymbol{S} - 2\lambda_1 \boldsymbol{b}_1^\top\,, \qquad \frac{\partial \mathfrak{L}}{\partial \lambda_1} = 1 - \boldsymbol{b}_1^\top \boldsymbol{b}_1\,,$$

$$\boldsymbol{S}\boldsymbol{b}_1 = \lambda_1 \boldsymbol{b}_1\,,$$

$$\boldsymbol{b}_1^\top \boldsymbol{b}_1 = 1\,.$$



Projection Perspective

Can also be thought of as minimizing reconstruction error, distance from points to projection

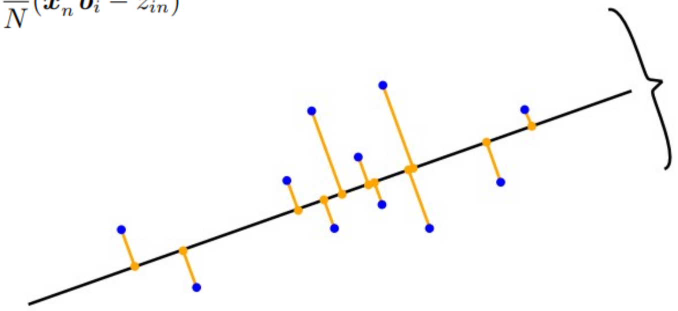
$$J_M := \frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 ,$$

$$\frac{\partial J_M}{\partial z_{in}} = \frac{\partial J_M}{\partial \tilde{\boldsymbol{x}}_n}\frac{\partial \tilde{\boldsymbol{x}}_n}{\partial z_{in}} ,$$

$$\frac{\partial J_M}{\partial \tilde{\boldsymbol{x}}_n} = -\frac{2}{N}(\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n)^\top \in \mathbb{R}^{1\times D}$$

$$\frac{\partial \tilde{\boldsymbol{x}}_n}{\partial z_{in}} \overset{(10.28)}{=} \frac{\partial}{\partial z_{in}}\left(\sum_{m=1}^{M} z_{mn}\boldsymbol{b}_m\right) = \boldsymbol{b}_i$$

$$\overset{\text{ONB}}{=} -\frac{2}{N}(\boldsymbol{x}_n^\top \boldsymbol{b}_i - z_{in}\boldsymbol{b}_i^\top \boldsymbol{b}_i) = -\frac{2}{N}(\boldsymbol{x}_n^\top \boldsymbol{b}_i - z_{in})$$



Coordinates : $z_{in} = \boldsymbol{x}_n^\top \boldsymbol{b}_i = \boldsymbol{b}_i^\top \boldsymbol{x}_n$

Optimal projection is orthonormal:

$$\tilde{\boldsymbol{x}}_n = \sum_{m=1}^{M} z_{mn}\boldsymbol{b}_m \overset{(10.32)}{=} \sum_{m=1}^{M}(\boldsymbol{x}_n^\top \boldsymbol{b}_m)\boldsymbol{b}_m .$$

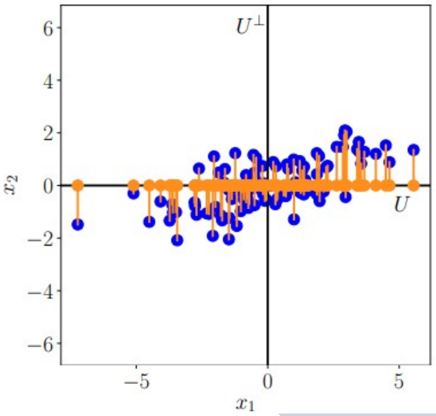
$$\boldsymbol{x}_n = \sum_{d=1}^{D} z_{dn}\boldsymbol{b}_d \overset{(10.32)}{=} \sum_{d=1}^{D}(\boldsymbol{x}_n^\top \boldsymbol{b}_d)\boldsymbol{b}_d = \left(\sum_{d=1}^{D}\boldsymbol{b}_d\boldsymbol{b}_d^\top\right)\boldsymbol{x}_n$$

$$= \left(\sum_{m=1}^{M}\boldsymbol{b}_m\boldsymbol{b}_m^\top\right)\boldsymbol{x}_n + \left(\sum_{j=M+1}^{D}\boldsymbol{b}_j\boldsymbol{b}_j^\top\right)\boldsymbol{x}_n ,$$

$$\tilde{\boldsymbol{x}} = \boldsymbol{B}(\underbrace{\boldsymbol{B}^\top\boldsymbol{B}}_{=\boldsymbol{I}})^{-1}\boldsymbol{B}^\top\boldsymbol{x} = \boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{x} ,$$

$$\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n = \left(\sum_{j=M+1}^{D}\boldsymbol{b}_j\boldsymbol{b}_j^\top\right)\boldsymbol{x}_n$$

$$= \sum_{j=M+1}^{D}(\boldsymbol{x}_n^\top\boldsymbol{b}_j)\boldsymbol{b}_j .$$



Projection Perspective – Basis Vectors

We get projection matrix: $$\sum_{m=1}^{M} \boldsymbol{b}_m \boldsymbol{b}_m^\top = \boldsymbol{B}\boldsymbol{B}^\top .$$

In order to minimize, need to find the best rank-*M* approximation of $BB^T = I$

$$\frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 = \frac{1}{N}\sum_{n=1}^{N}\left\|\boldsymbol{x}_n - \boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{x}_n\right\|^2$$
$$= \frac{1}{N}\sum_{n=1}^{N}\left\|(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{B}^\top)\boldsymbol{x}_n\right\|^2 .$$

$$J_M = \frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2 \overset{(10.38b)}{=} \frac{1}{N}\sum_{n=1}^{N}\left\|\sum_{j=M+1}^{D}(\boldsymbol{b}_j^\top\boldsymbol{x}_n)\boldsymbol{b}_j\right\|^2$$

$$J_M = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=M+1}^{D}(\boldsymbol{b}_j^\top\boldsymbol{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=M+1}^{D}\boldsymbol{b}_j^\top\boldsymbol{x}_n\boldsymbol{b}_j^\top\boldsymbol{x}_n$$
$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{j=M+1}^{D}\boldsymbol{b}_j^\top\boldsymbol{x}_n\boldsymbol{x}_n^\top\boldsymbol{b}_j ,$$

$$J_M = \sum_{j=M+1}^{D}\boldsymbol{b}_j^\top\underbrace{\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^\top\right)}_{=:\boldsymbol{S}}\boldsymbol{b}_j = \sum_{j=M+1}^{D}\boldsymbol{b}_j^\top\boldsymbol{S}\boldsymbol{b}_j \qquad (10.43a)$$
$$= \sum_{j=M+1}^{D}\mathrm{tr}(\boldsymbol{b}_j^\top\boldsymbol{S}\boldsymbol{b}_j) = \sum_{j=M+1}^{D}\mathrm{tr}(\boldsymbol{S}\boldsymbol{b}_j\boldsymbol{b}_j^\top) = \mathrm{tr}\Big(\underbrace{\Big(\sum_{j=M+1}^{D}\boldsymbol{b}_j\boldsymbol{b}_j^\top\Big)}_{\text{projection matrix}}\boldsymbol{S}\Big),$$

$$J_M = \sum_{j=M+1}^{D}\lambda_j ,$$

The distance between the data’s subspace and the projection’s subspace is proportional to the covariance matrix S, because the optimal projection is orthogonal

From here, the derivations can continue from the maximum variance calculations

Therefore, minimizing the projection is equivalent to maximizing the variance

Summary So Far

- **Principal components** – eigenvectors of the correlation matrix S

- PCA uses the first M principal components to generate the basis for a portion the underlying variance of the data

- A new subspace, the principal subspace, is created from the principal components, and data points are projected from the original set using the linear transformation $z = B^T x$

- This projection both minimizes the euclidean distance between the data and its projection and maximizes the variance of its encoding

Methods of Computing PCA

- Can perform eigendecomposition since S is square
- Can use SVD – Columns of U are eigenvectors of XX^T, or S

$$\underbrace{\boldsymbol{X}}_{D\times N} = \underbrace{\boldsymbol{U}}_{D\times D}\underbrace{\boldsymbol{\Sigma}}_{D\times N}\underbrace{\boldsymbol{V}^\top}_{N\times N},$$

This relationship between the eigenvalues of $\boldsymbol{S}$ and the singular values of $\boldsymbol{X}$ provides the connection between the maximum variance view (Section 10.2) and the singular value decomposition.

- Using low rank approximations $\tilde{\boldsymbol{X}}_M := \operatorname{argmin}_{\operatorname{rk}(\boldsymbol{A})\leqslant M} \|\boldsymbol{X} - \boldsymbol{A}\|_2 \in \mathbb{R}^{D\times N}$

- Power Iteration (for first eigenvector) $\boldsymbol{x}_{k+1} = \frac{\boldsymbol{S}\boldsymbol{x}_k}{\|\boldsymbol{S}\boldsymbol{x}_k\|}, \quad k = 0, 1, \dots .$

- For high dimensional data (e.g. images) can turn *DxD* into *NxN* matrix

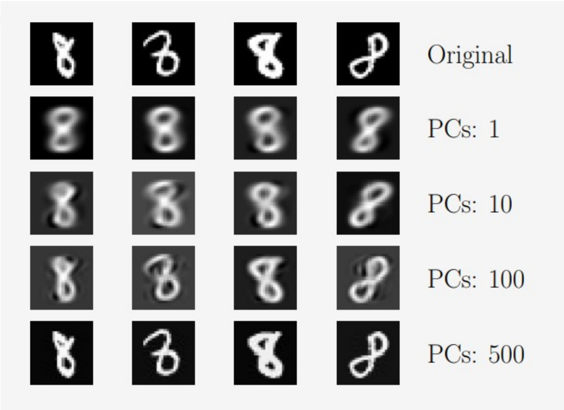
$$\boldsymbol{S}\boldsymbol{b}_m = \lambda_m \boldsymbol{b}_m, \quad m = 1, \dots, M,$$

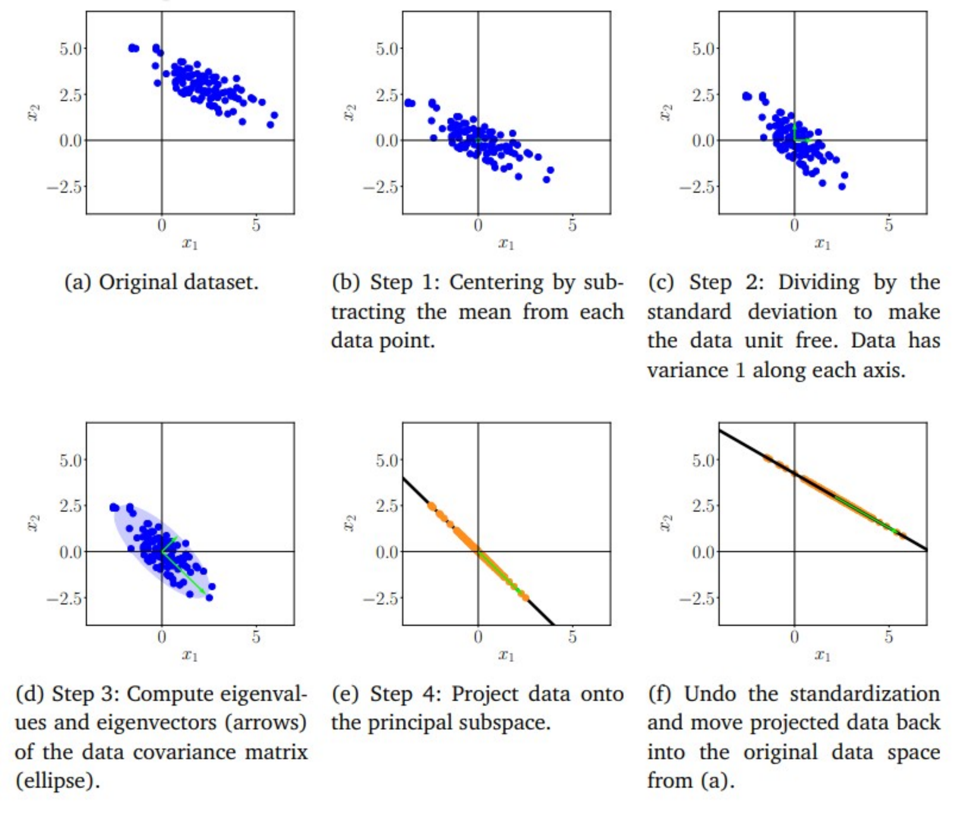
$$\frac{1}{N}\underbrace{\boldsymbol{X}^\top\boldsymbol{X}}_{N\times N}\underbrace{\boldsymbol{X}^\top\boldsymbol{b}_m}_{=:\boldsymbol{c}_m} = \lambda_m \boldsymbol{X}^\top\boldsymbol{b}_m \iff \frac{1}{N}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{c}_m = \lambda_m\boldsymbol{c}_m,$$

  - Get eigenvectors of X^TX, then recover Xc_m as eigenvector of S $\underbrace{\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top}_{\boldsymbol{S}}\boldsymbol{X}\boldsymbol{c}_m = \lambda_m\boldsymbol{X}\boldsymbol{c}_m$

Steps to Compute PCA

- Subtract mean from all data points (mean = 0)
- **Standardize** (divide data points by stddev)
- Get eigenvectors of covariance matrix
- Project data onto principal subspace
- Multiply by original stddev and add back mean





(a) Original dataset.

(b) Step 1: Centering by subtracting the mean from each data point.

(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.

(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).

(e) Step 4: Project data onto the principal subspace.

(f) Undo the standardization and move projected data back into the original data space from (a).

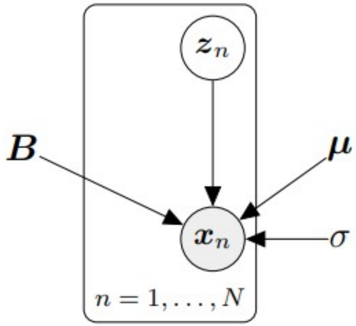
Probabilistic PCA (PPCA)

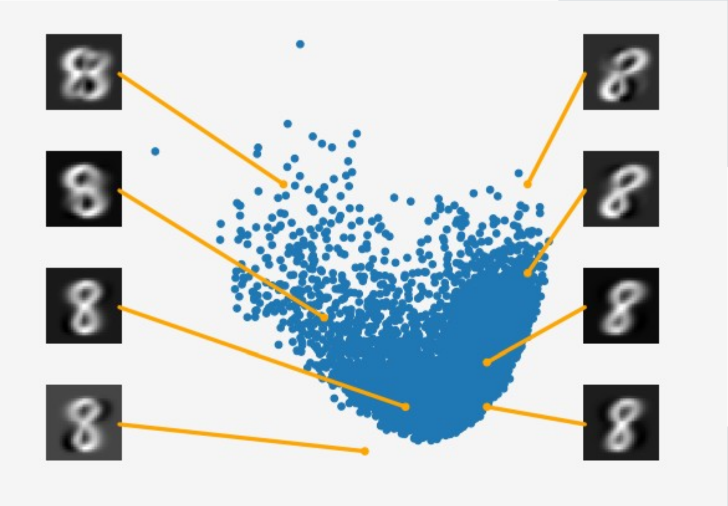
Can also consider *z* to be a latent variable

$$\boldsymbol{x} = \boldsymbol{B}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \in \mathbb{R}^D$$

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{B}, \boldsymbol{\mu}, \sigma^2) = \mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{B}\boldsymbol{z} + \boldsymbol{\mu},\, \sigma^2\boldsymbol{I})$$

$$\boldsymbol{z}_n \sim \mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{0},\, \boldsymbol{I})$$
$$\boldsymbol{x}_n\,|\,\boldsymbol{z}_n \sim \mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{B}\boldsymbol{z}_n + \boldsymbol{\mu},\, \sigma^2\boldsymbol{I})$$



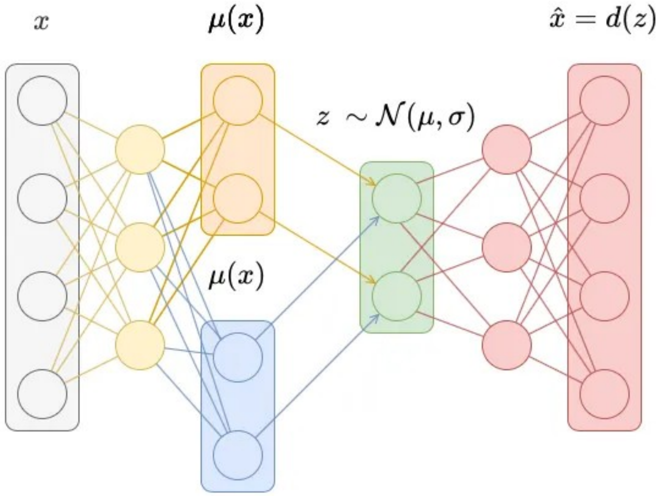


Can use likelihood and posterior for MAP and MLE

$$\begin{aligned} p(\boldsymbol{x}\,|\,\boldsymbol{B}, \boldsymbol{\mu}, \sigma^2) &= \int p(\boldsymbol{x}\,|\,\boldsymbol{z}, \boldsymbol{\mu}, \sigma^2)p(\boldsymbol{z})\mathrm{d}\boldsymbol{z} \\ &= \int \mathcal{N}(\boldsymbol{x}\,|\,\boldsymbol{B}\boldsymbol{z} + \boldsymbol{\mu},\, \sigma^2\boldsymbol{I})\mathcal{N}(\boldsymbol{z}\,|\,\boldsymbol{0},\, \boldsymbol{I})\mathrm{d}\boldsymbol{z} \end{aligned}$$

Example – Variational Autoencoder

Trains statistics for normally distributed latent variable *z*



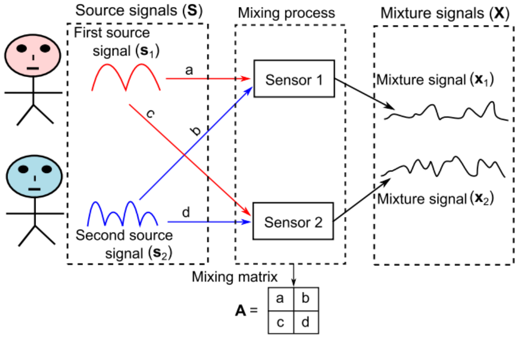
Source

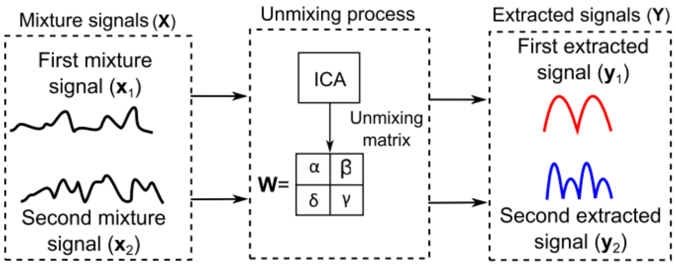
- Optimizes objective function (euclidean distance, but also implicitly variance)
- Lower dimensional representation

https://medium.com/@weidagang/demystifying-neural-networks-variational-autoencoders-6a44e75d0271

Further Reading – Independent Component Analysis

PCA models lower dimensional subspace, ICA models underlying signals
PCA optimizes covariance, ICA optimizes higher-order metrics like kurtosis





ICA extracts I.I.D non-Gaussian signals and measure “Gaussianity”

Steps
- Subtract mean from data points
- “decorrelate” – project into PCA principal subspace
- Scale/normalize
- Find the “unmixing” matrix W, where x = ASW, by maximizing “non-gaussianity”, e.g. kurtosis, negentropy
- Projecting x into W gives signals

https://www.emerald.com/insight/content/doi/10.1016/j.aci.2018.08.006/full/pdf?title=independent-component-analysis-an-introduction