Variational Autoencoder

11-785 Introduction to Deep Learning

- lecture 21 -

TAVE Research DL001
Heeji Won

Contents

Contents

01. Generative model

a model that can generate data with a distribution similar to the given data x

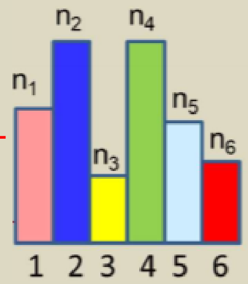
Learning a generative model

"Estimate the $\boldsymbol{\theta}$ such that $\boldsymbol{P}(\boldsymbol{x};\boldsymbol{\theta})$ best 'fits' the observations $\boldsymbol{X} = \{\boldsymbol{x}\}$"

$$\underset{\theta}{\operatorname{argmax}}\, P(X;\theta) = \underset{\theta}{\operatorname{argmax}} \log(P(X;\theta))$$

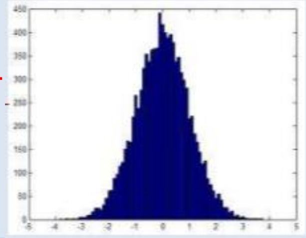
$$\underset{\{p_1,p_2,p_3,p_4,p_5,p_6\}}{\operatorname{argmax}} \sum_i n_i \log(p_i)$$

$p_i = \frac{n_i}{N}$ (N is the total number of observations)



$$\underset{\mu,\sigma^2}{\operatorname{argmax}} \sum_{x\in X} \log Gaussian(x;\mu,\sigma^2)$$

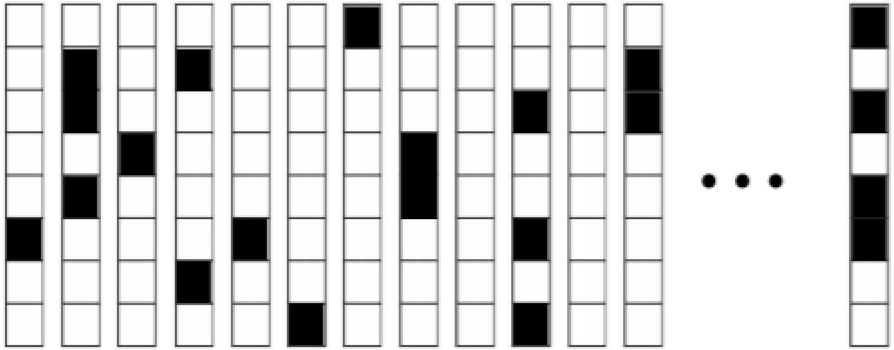
$\mu = \frac{1}{N}\sum_{x\in X} x$; $\sigma^2 = \frac{1}{N}\sum_{x\in X}(x-\mu)^2$



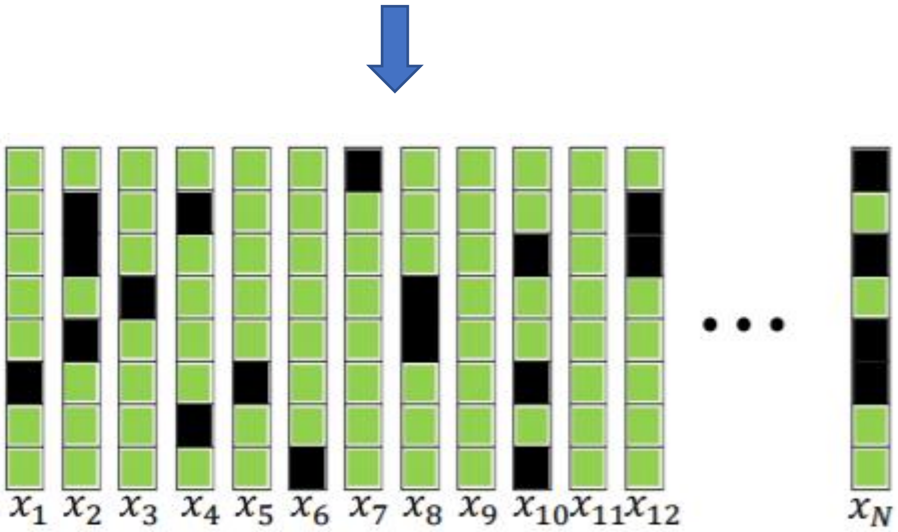
Contents

02. How to deal with incomplete data

if the data have missing components



Blacked-out components are missing from data



- Complete data includes the observed & missing components

$$X = \{x_1, \ldots, x_N\}, \qquad \mathrm{x}_i = (o_i, m_i)$$

- Original problem :

$$\underset{\mu,\sigma^2}{\operatorname{argmax}} \sum_{x \in X} \log P(x)$$

where X is the entire data!

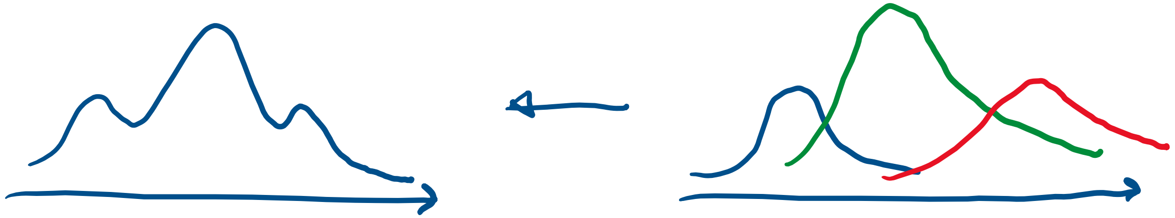
But, there are missing values

$$\underset{\mu,\sigma^2}{\operatorname{argmax}} \log(P(O)) = \underset{\mu,\sigma^2}{\operatorname{argmax}} \sum_{o \in O} \log P(o)$$

$$= \underset{\mu,\sigma^2}{\operatorname{argmax}} \sum_{o \in O} \log \int_{-\infty}^{\infty} P(o, m)\, dm$$

02. How to deal with incomplete data

cf) The Gaussian Mixture model



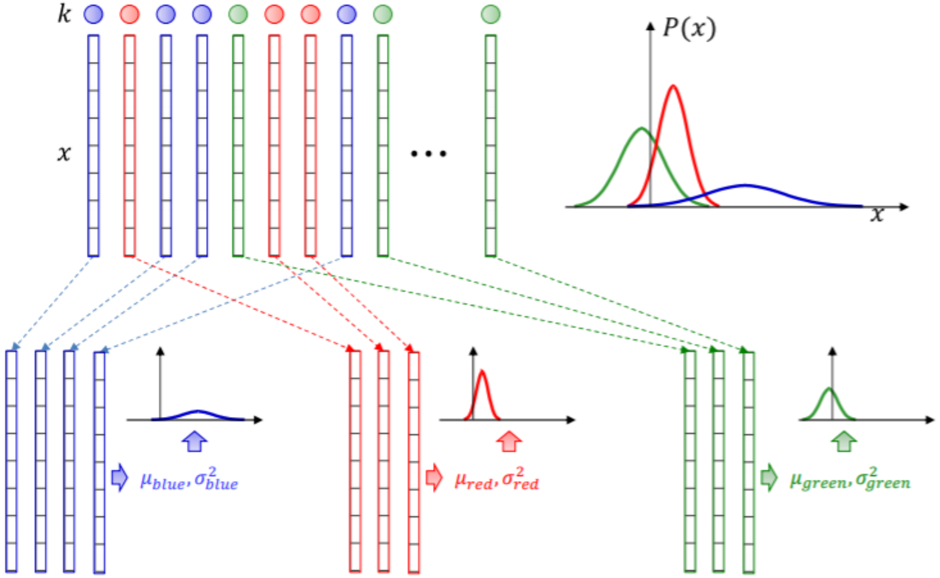
$\sim N(\mu_1, \sigma_1^2)$

$\sim N(\mu_2, \sigma_2^2)$

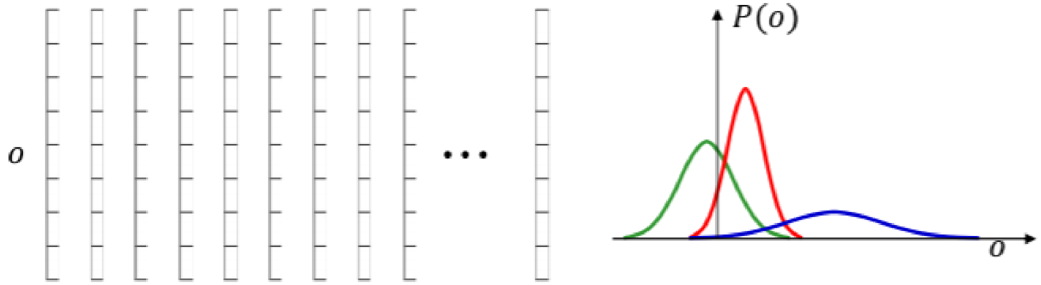
$\sim N(\mu_3, \sigma_3^2)$

The structure of the network

- if learning a GMM with 'complete' data



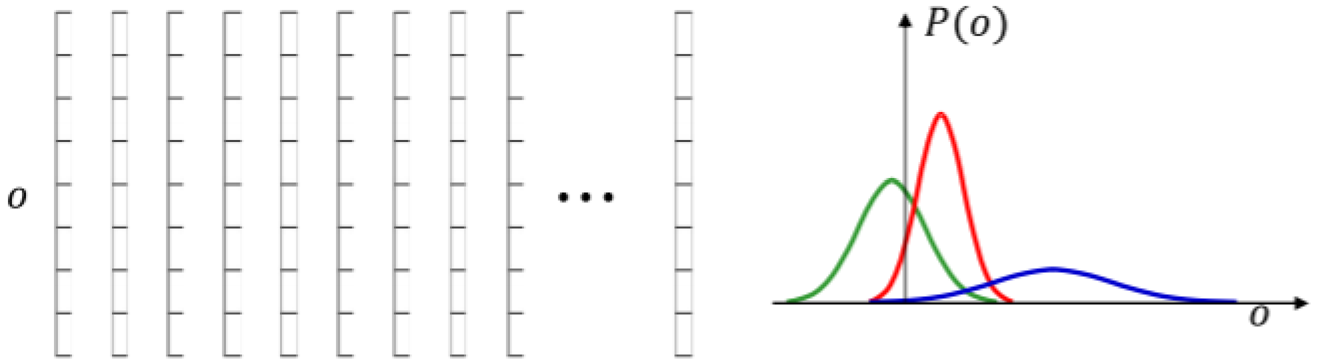
✓ But, we are not given the actual Gaussian for each o_i



- What we want : $(o_1, k_1), (o_2, k_2), \dots$
- What we have : $o_1, o_2, \dots$

02. How to deal with incomplete data

The structure of the network



we are not given the actual Gaussian for each o_i

– MLE with only observed data

$$\underset{\{(\mu_k,\sigma_k^2),\forall k\}}{\operatorname{argmax}} \log(P(O)) = \underset{\{(\mu_k,\sigma_k^2),\forall k\}}{\operatorname{argmax}} \sum_{o \in O} \log P(o), \qquad P(o) = \sum_k P(k) N(o; \mu_k, \sigma_k^2)$$

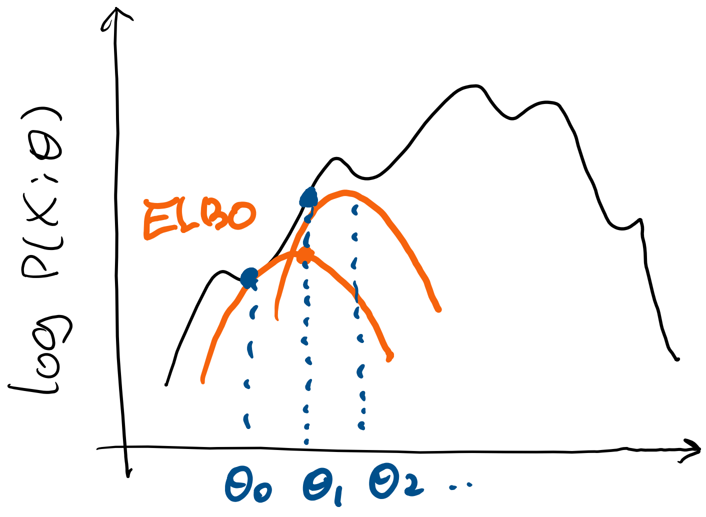
$$= \underset{\{(\mu_k,\sigma_k^2),\forall k\}}{\operatorname{argmax}} \sum_{o \in O} \log \sum_k P(k) N(o; \mu_k, \sigma_k^2)$$

challenging! => **EM algorithm**

- ✓ no closed form solutions
- ✓ need efficient iterative algorithms

Contents

03. Expectation Maximization



- initialize θ^0
- k = 0
- iterate (over k) until $\sum_{o \in O} P(o;\theta)$ converges:
  - Expectation Step:

    Compute $P(h|o;\theta)$ for all $o \in O$ for all k

  - Maximization Step:

    $\theta^{k+1} \leftarrow armax_\theta \underbrace{\sum_{o \in O} P(h|o;\theta^k) logP(h,o;\theta)}_{\text{ELBO}}$

* h : missing components

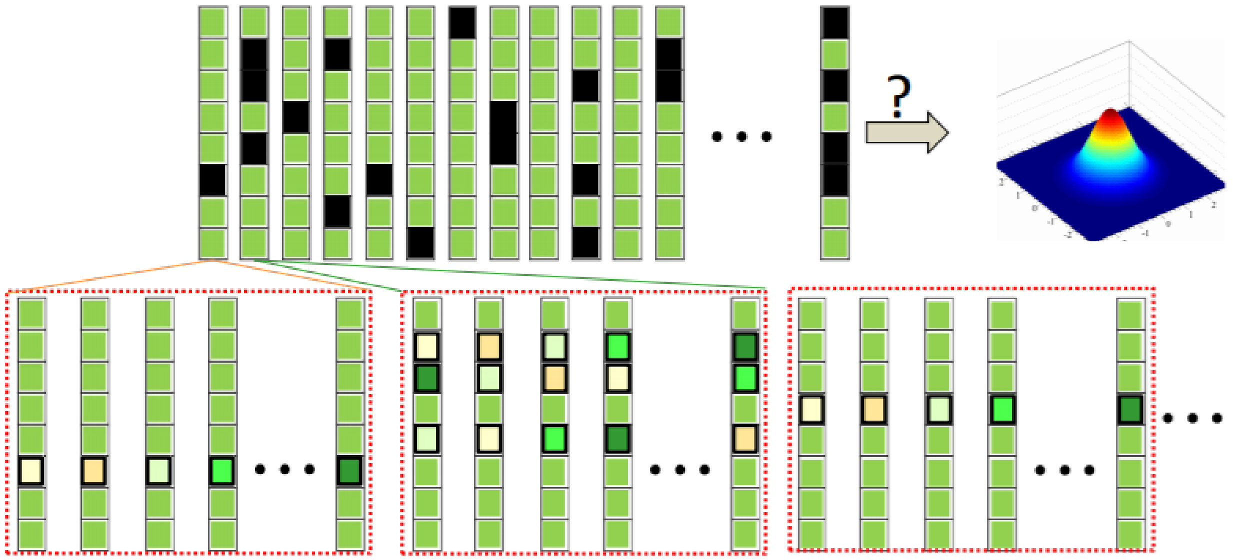
<Construct an ELBO(empirical lower bound function) $J(\theta, \theta^k)$>

$$J(\theta,\theta^k) = \boxed{\sum_{o \in O}\sum_{h} P(h|o;\theta^k)\log P(h,o;\theta)} - \underbrace{\sum_{o \in O}\sum_{h} P(h|o;\theta^k)\log P(h|o;\theta^k)}_{\text{not a function of } \theta}$$

03. Expectation Maximization

if the data have missing components

- Completing incomplete vector



- Expand every incomplete vector out into all possibilities
- in proportion: $P(m|o)$

from a previous estimate of the model

- Let $x_i(m)$ be the 'completed' version of the observation o_i

$$x_i(m) = (m, o_i)$$

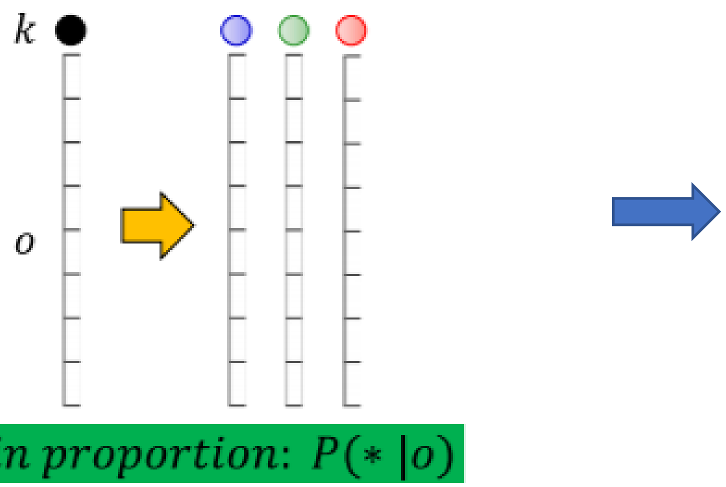
- Estimate from the expanded data

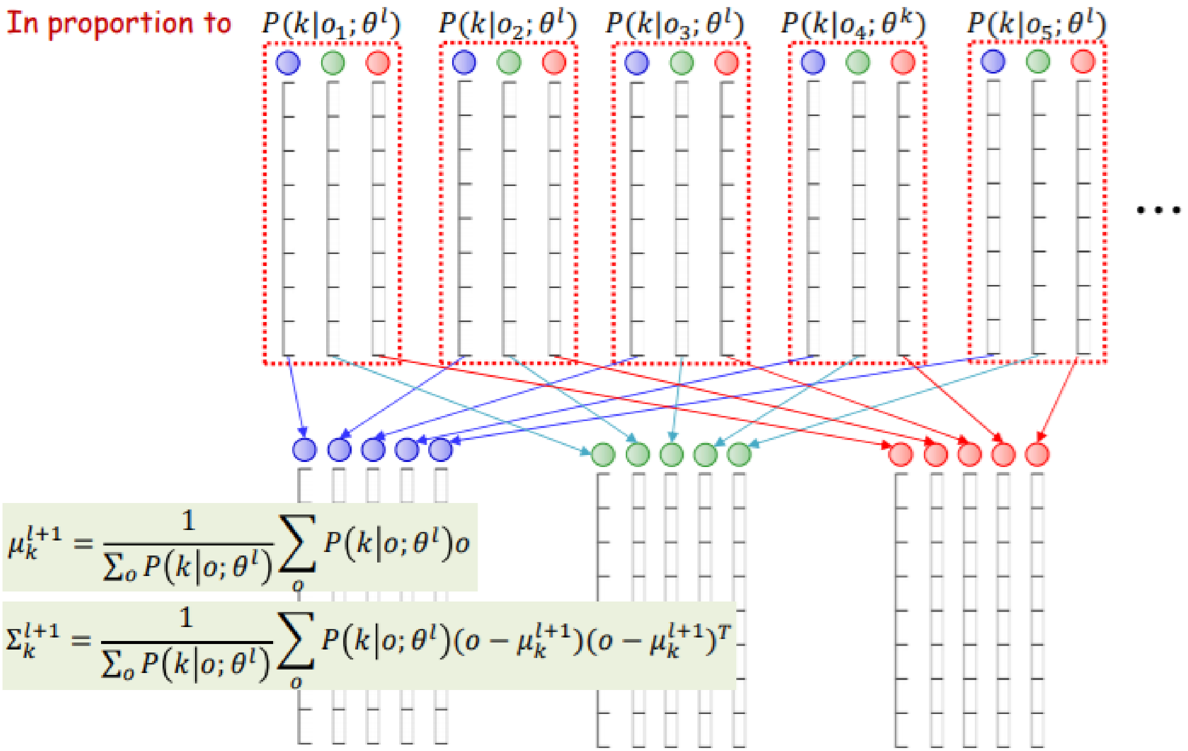
$$\mu^{k+1} = \frac{1}{N}\sum_{o \in O} \int_{-\infty}^{\infty} P(m|o;\theta^k)x_i(m)dm$$

$$\Sigma^{k+1} = \frac{1}{N}\sum_{o \in O} \int_{-\infty}^{\infty} P(m|o;\theta^k)(x_i(m)-\mu^{k+1})(x_i(m)-\mu^{k+1})^T dm$$

03. Expectation Maximization

The structure of network





$$\mu_k^{l+1} = \frac{1}{\sum_o P(k|o;\theta^l)} \sum_o P(k|o;\theta^l) o$$

$$\Sigma_k^{l+1} = \frac{1}{\sum_o P(k|o;\theta^l)} \sum_o P(k|o;\theta^l)(o-\mu_k^{l+1})(o-\mu_k^{l+1})^T$$

Proportion to $P(k|o)$ which can be computed if we know $P(k)$ and $P(o|k)$

from previous estimate of model

iterate!

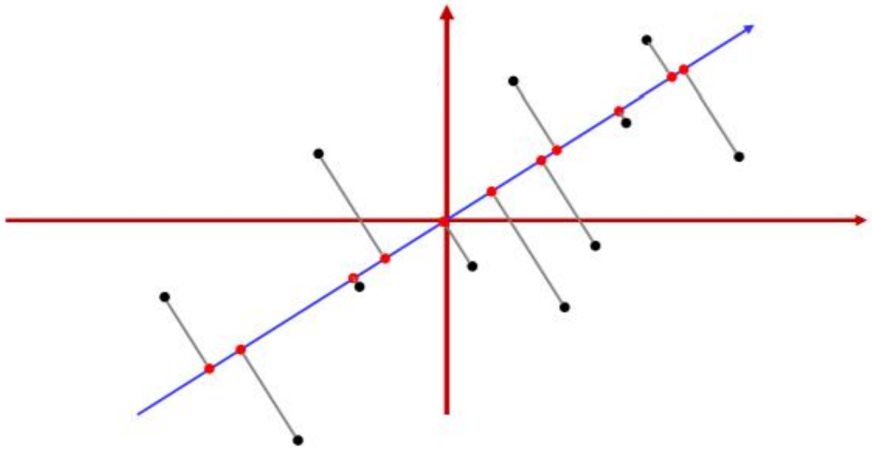
< EM principle >

or sampling

- ✓ 'Complete' the data by considering every possible value for missing data in proportion to **posterior prob.**
- ✓ **Re-estimate** parameters

Contents

04. PCA

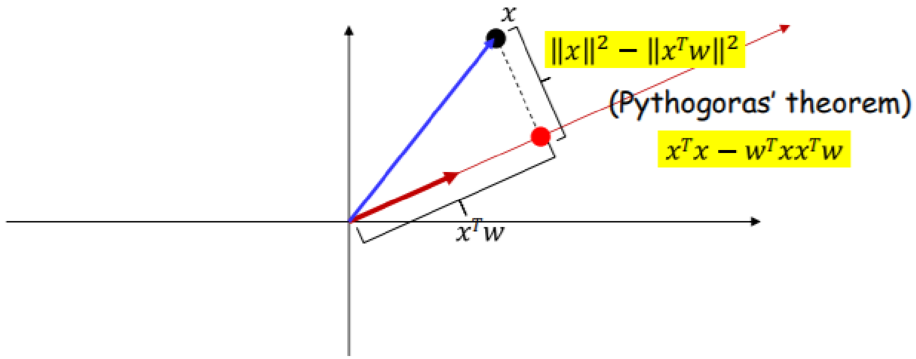


- Find the principal subspace that can be projected
- Minimize the sum of the squared lengths
- There are several method to find

1) Search method

" search through all subspaces with minimum projection error"

2) Close form



minimizing L_2 error : $L = \sum_x x^T x - wTxx^T w$

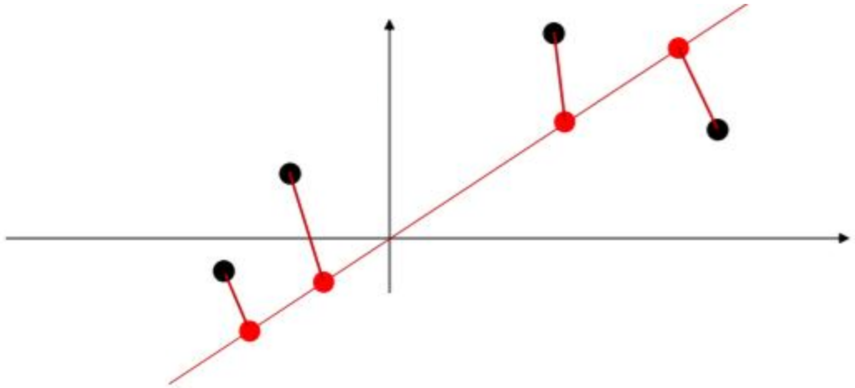
minimizing L

$$\left(\sum_x x^T x\right) w = \lambda w$$

eigenvalue equation

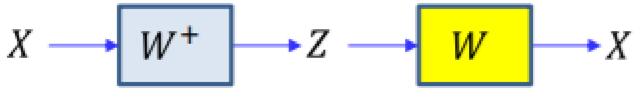
04. PCA

3) The iterative algorithm



"Let W rotate and stretch/shrink, keeping the arrangement of Z location fixed"

Drawing this differently



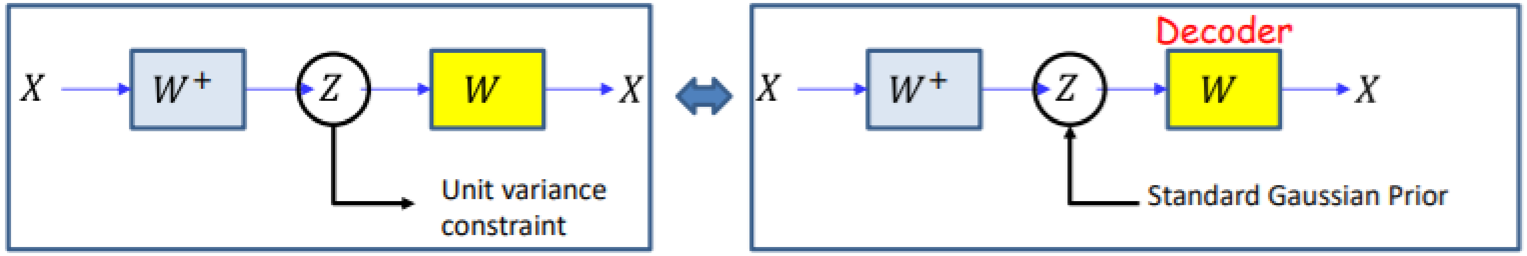
- Autoencoder with linear activation!
- But, the solution is not unique!
  - ✓ Scale invariance
  - ✓ Rotation invariance

04. PCA

Find a unique W

1. Orthogonal & unit eigen vector : standard eigen vector

2. Constrain the variance of Z to be unity

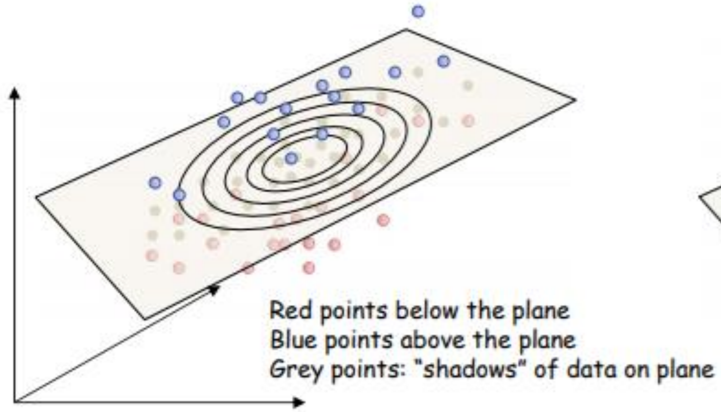


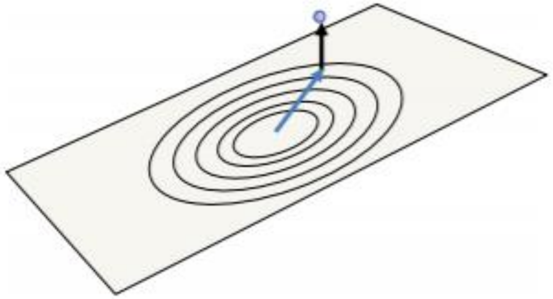
- Imposing the constraint that z must have unit variance is the same as assuming that is **drawn from a standard Gaussian**

- The decoder of AE with the unit-variance constraint on z is in fact a **Generative model**

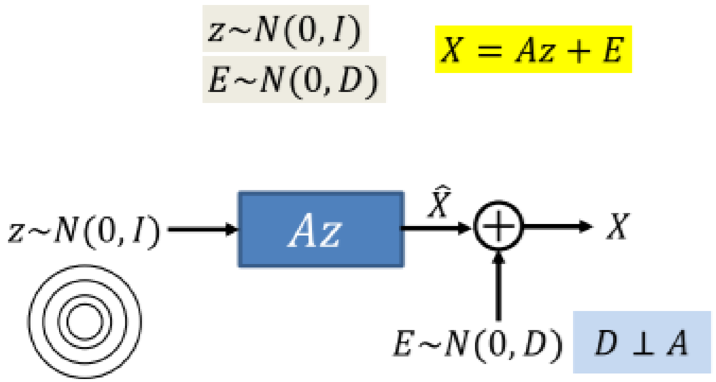
Contents

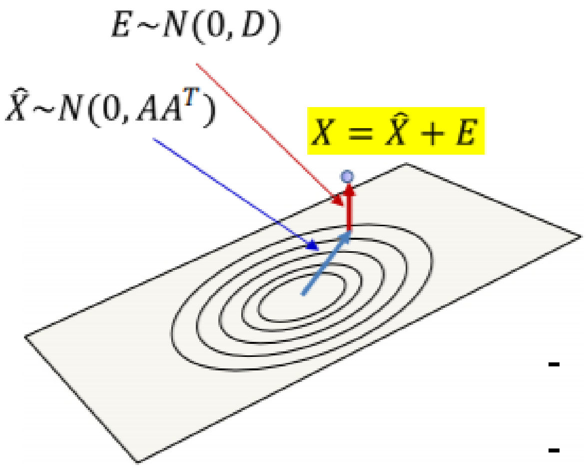
05. VAE





- take a Gaussian step on the principal plane
- take a orthogonal Gaussian step where we land to generate a point

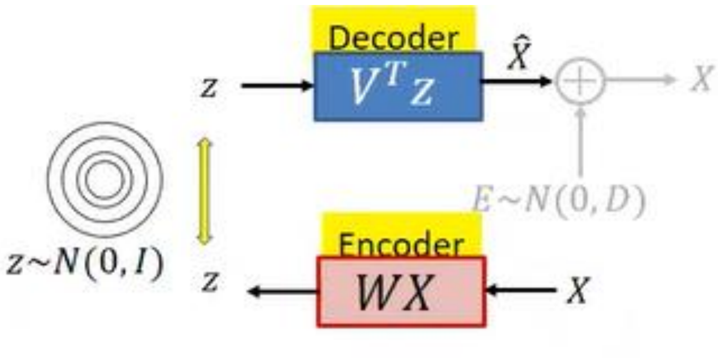




- z is drawn from K-dim isotropic Gaussian
- A is a basis matrix
- E is a Gaussian noise that is orthogonal

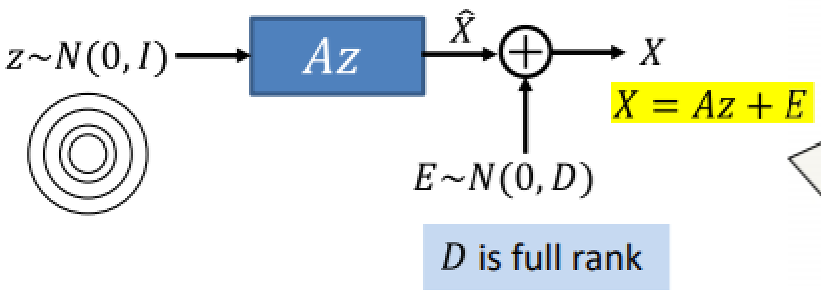
* covariance is low-rank

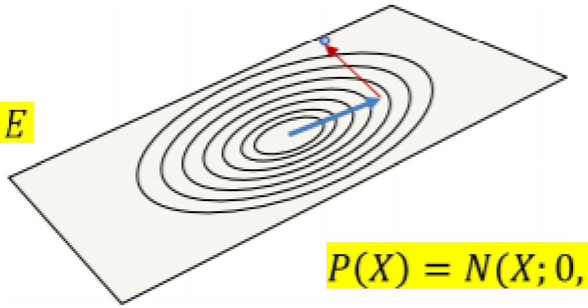
05. VAE



- The decoder weights are just the PCA basis matrix
- Encoder: transforms input X into Gaussian z
- Decoder: transforms Gaussian z into principal subspace reconstruction $\hat{X}$

The Linear Gaussian Model





$P(X) = N(X; 0, AA^T + D)$

- Update the model : The noise can lie in any direction
- Noise is drawn from full-rank uncorrelated Gaussian distribution

⇒ The way to produce any data instance is no longer unique!

also a generative model!

Also known as Factor Analysis

- A is the loading matrix
- z are the factors
- D is diagonal

Thank you