Chapter 13
Entangled Models

supplementary slides to
Machine Learning Fundamentals

published by Cambridge University Press

August 2020

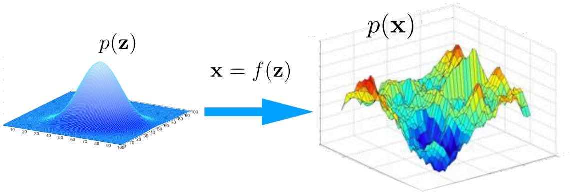
CAMBRIDGE
UNIVERSITY PRESS

Outline

Entangled Models (I)

Borel isomorphism theorem

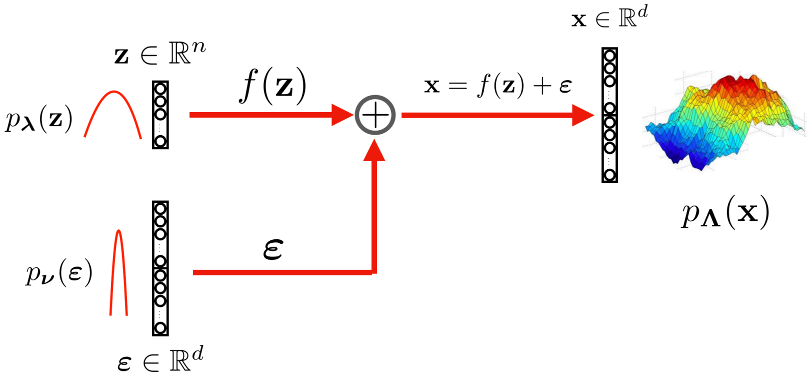
Given a normally distributed random variable $z \sim \mathcal{N}(0, 1)$, for any smooth probability distribution $p(\mathbf{x})$ ($\mathbf{x} \in \mathbb{R}^d$), there exist L^p functions: $f_1(z), \cdots, f_d(z)$ to convert z into a vector $f(z) = \left[f_1(z) \ \cdots \ f_d(z)\right]^\intercal$ so that $f(z)$ follows this distribution, i.e. $f(z) \sim p(\mathbf{x})$.



Entangled Models (II)

- **entangled models**: combine simple models with a complex transformation to derive complex models
- linear vs. non-linear transformations
- disentangled representation learning
  - entangling assumption: independent features $\mathbf{z}$, called *factors*, are entangled by a mixing function $\mathbf{x} = f(\mathbf{z})$ to observations $\mathbf{x}$
  - add residual noises to compute likelihood
  - disentangling: $\mathbf{x} \longmapsto \mathbf{z}$
- three subgroups of entangled models:
  - linear Gaussian models: Gaussian + linear transformation
  - non-Gaussian models: non-Gaussian + linear transformation
  - deep generative models: Gaussian + neural networks

Entangled Models (III)



Entangled Models (IV)

entangled models	factor $\mathbf{z} \sim p(\mathbf{z})$	residual $\boldsymbol{\varepsilon} \sim p(\boldsymbol{\varepsilon})$	mixing $f(\mathbf{z})$
probabilistic PCA	$\mathcal{N}(\mathbf{z}\vert\mathbf{0},\mathbf{I})$	$\mathcal{N}(\boldsymbol{\varepsilon}\vert\mathbf{0},\sigma^2\mathbf{I})$	$\mathbf{Wz}$ linear
factor analysis	$\mathcal{N}(\mathbf{z}\vert\mathbf{0},\mathbf{I})$	$\mathcal{N}(\boldsymbol{\varepsilon}\vert\mathbf{0},\mathbf{D})$ $\mathbf{D}$: diagonal	$\mathbf{Wz}$ linear
ICA	$\prod_i p_i(z_i)$ non-Gaussian	–	$\mathbf{Wz}$ linear
IFA	$\prod_i p_i(z_i)$ factorial GMM	$\mathcal{N}(\boldsymbol{\varepsilon}\vert\mathbf{0},\boldsymbol{\Lambda})$	$\mathbf{Wz}$ linear
HOPE	mixture model (movMF/GMM)	$\mathcal{N}(\boldsymbol{\varepsilon}\vert\mathbf{0},\sigma^2\mathbf{I})$	$\mathbf{Wz}$ $\mathbf{W}$: orthogonal
VAE	$\mathcal{N}(\mathbf{z}\vert\mathbf{0},\mathbf{I})$	$\mathcal{N}(\boldsymbol{\varepsilon}\vert\mathbf{0},\sigma^2\mathbf{I})$	$f(\cdot) \in L^p$ neural nets
GAN	$\mathcal{N}(\mathbf{z}\vert\mathbf{0},\mathbf{I})$	–	$f(\cdot) \in L^p$ neural nets

Learning of Entangled Models

1. Jacobian method: if the mixing function is invertible and differentiable
   - entangled models $\Longrightarrow$ maximum likelihood estimation

$$p_{\boldsymbol{\Lambda}}(\mathbf{x}) = \left| \, \mathbf{J} \, \right| p_{\boldsymbol{\lambda}}\big(f_1^{-1}(\mathbf{x})\big) \, p_{\boldsymbol{\nu}}\big(f_2^{-1}(\mathbf{x})\big)$$

   - disentangling: $\mathbf{z} = f_1^{-1}(\mathbf{x})$
2. marginalization method
   - entangled models $\Longrightarrow$ maximum likelihood estimation

$$p_{\boldsymbol{\Lambda}}(\mathbf{x}) = \int_{\mathbf{z}} p_{\boldsymbol{\lambda}}(\mathbf{z}) p_{\boldsymbol{\nu}}\big(\mathbf{x} - f(\mathbf{z}; \mathbf{W})\big) \, d\mathbf{z}$$

   - disentangling:

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z}, \mathbf{x})}{p(\mathbf{x})} = \frac{p_{\boldsymbol{\lambda}}(\mathbf{z}) p_{\boldsymbol{\nu}}\big(\mathbf{x} - f(\mathbf{z}; \mathbf{W})\big)}{\int_{\mathbf{z}} p_{\boldsymbol{\lambda}}(\mathbf{z}) p_{\boldsymbol{\nu}}\big(\mathbf{x} - f(\mathbf{z}; \mathbf{W})\big) \, d\mathbf{z}}$$

Linear Gaussian Models

- linear Gaussian models are a group of simple entangled models
- factors $\mathbf{z}$ follow a zero-mean multivariate Gaussian:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \,|\, \mathbf{0}, \boldsymbol{\Sigma}_1)$$

- residual $\boldsymbol{\varepsilon}$ follows another multivariate Gaussian

$$p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}_2)$$

- the mixing function is linear:

$$f(\mathbf{z}; \mathbf{W}) = \mathbf{W}\mathbf{z}$$

- linear Gaussian models:

$$p_{\boldsymbol{\Lambda}}(\mathbf{x}) = \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, \mathbf{W}\boldsymbol{\Sigma}_1\mathbf{W}^\intercal + \boldsymbol{\Sigma}_2)$$

Probabilistic PCA

- probabilistic PCA: a special case of liner Gaussian models
- factor distribution: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \,|\, \mathbf{0}, \mathbf{I})$
- residual distribution: an isotropic covariance Gaussian $p_\sigma(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon} \,|\, \boldsymbol{\mu}, \sigma^2\mathbf{I})$
- probabilistic PCA models: $p_{\boldsymbol{\Lambda}}(\mathbf{x}) = \mathcal{N}\big(\mathbf{x} \,|\, \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\intercal + \sigma^2\mathbf{I}\big)$
- the log-likelihood function

$$l(\mathbf{W}, \boldsymbol{\mu}, \sigma^2) = C - \frac{N}{2} \ln \left|\mathbf{W}\mathbf{W}^\intercal + \sigma^2\mathbf{I}\right| - \frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})^\intercal \left(\mathbf{W}\mathbf{W}^\intercal + \sigma^2\mathbf{I}\right)^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$$

- maximum likelihood estimation $\mathbf{W}_{\mathsf{MLE}} \implies$ a rotation of principle components in regular PCA
- disentangling: $p(\mathbf{z} \,|\, \mathbf{x}) = \mathcal{N}\Big(\mathbf{z} \,|\, \mathbf{M}^{-1}\mathbf{W}^\intercal(\mathbf{x} - \boldsymbol{\mu}), \sigma^{-2}\mathbf{M}\Big)$ where $\mathbf{M} = \mathbf{W}^\intercal\mathbf{W} + \sigma^2\mathbf{I}$

Factor Analysis (I)

- probabilistic PCA: a special case of liner Gaussian models
- factor distribution: $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \,|\, \mathbf{0}, \mathbf{I})$
- residual distribution: a diagonal covariance Gaussian

$$p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon} \,|\, \boldsymbol{\mu}, \mathbf{D})$$

  where $\mathbf{D} \in \mathbb{R}^{d \times d}$ is a diagonal covariance matrix
- factor analysis models:

$$p_{\boldsymbol{\Lambda}}(\mathbf{x}) = \mathcal{N}\left(\mathbf{x} \,\middle|\, \boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\intercal + \mathbf{D}\right)$$

- the log-likelihood function:

$$l(\mathbf{W}, \mathbf{D}) = C - \frac{N}{2}\left[\ln\left|\mathbf{W}\mathbf{W}^\intercal + \mathbf{D}\right| + \mathrm{tr}\Big(\big(\mathbf{W}\mathbf{W}^\intercal + \mathbf{D}\big)^{-1}\mathbf{S}\Big)\right]$$

- no closed-form solution for maximum likelihood estimation

Factor Analysis (II)

alternating MLE method

Input: the sample covariance matrix $\mathbf{S}$
Output: $\mathbf{W}$ and $\mathbf{D}$

randomly initialize $\mathbf{D}_0$; set $t = 1$
while not converged **do**

1. construct $\mathbf{P}_t$ using the n leading eigenvectors of $\mathbf{D}_{t-1}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}_{t-1}^{-\frac{1}{2}}$
2. $\mathbf{W}_t = \mathbf{D}_t^{\frac{1}{2}}\mathbf{P}_t$
3. $\mathbf{D}_t = \text{diag}\Big(\mathbf{S} - \mathbf{W}_t\mathbf{W}_t^\intercal\Big)$
4. $t = t + 1$

end while

Non-Gaussian Entangled Models

use a non-Gaussian factor distribution and a linear mixing function

- Independent Component Analysis (ICA)

$$p(\mathbf{z}) = \prod_{j=1}^{n} p(z_j) = \prod_{j=1}^{n} \frac{4}{\pi(e^{z_j} + e^{-z_j})}$$

- Independent Factor Analysis (IFA)

$$p(z_j) = \sum_{m=1}^{M} w_{jm} \mathcal{N}(z_j \,|\, \mu_{jm}, \sigma_{jm}^2)$$

- Hybrid Orthogonal Projection and Estimation (HOPE)
  - $p(\mathbf{z})$ is a mixture model in $\mathbb{R}^n$
  - $p(\boldsymbol{\varepsilon})$ is a zero-mean isotropic covariance Gaussian in $\mathbb{R}^{d-n}$: $p(\boldsymbol{\varepsilon}) = \mathcal{N}(\boldsymbol{\varepsilon} \,|\, \mathbf{0}, \sigma^2 \mathbf{I})$
  - $\mathbf{W}$ is an orthogonal matrix

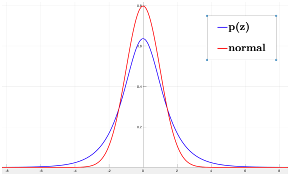


Figure: the normal distribution vs. the heavy-tail distribution $p(z)$ in ICA

Independent Component Analysis (ICA)

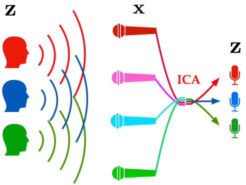
- ICA is used for blind source separation
- use a linear transformation to recover $\mathbf{z}$ from $\mathbf{x}$

$$\mathbf{z} = \mathbf{W}^{-1}\mathbf{x}$$

- the log-likelihood function:

$$l(\mathbf{W}^{-1}) = \sum_{i=1}^{N}\sum_{j=1}^{n} \ln p(\mathbf{w}_j^\intercal \mathbf{x}_i) + N \ln \left|\mathbf{W}^{-1}\right|$$

- maximum likelihood estimation: use any gradient descent methods to maximize $l(\mathbf{W}^{-1})$ with respect to $\mathbf{W}^{-1}$



blind source separation

Deep Generative Models

- combine a simple Gaussian with a complex non-linear mixing function
- use a deep neural network $\mathbb{W}$ for the mixing function:

$$\mathbf{x} = f(\mathbf{z}; \mathbb{W})$$

- cannot explicitly evaluate the likelihood function
- no direct way to disentangle: $p(\mathbf{z}|\mathbf{x})$
- use some tricks to bypass these difficulties:
  1. variational autoencoders (VAE)
  2. generative adversarial nets (GAN)
  3. normalizing flows

Variational Autoencoders (VAE): Formulation

- use a Gaussian $q(\mathbf{z}|\mathbf{x})$ to approximate the intractable distribution $p(\mathbf{z}|\mathbf{x})$ as:

$$p(\mathbf{z}|\mathbf{x}) \approx q(\mathbf{z}|\mathbf{x})$$

- $q(\mathbf{z}|\mathbf{x}) \triangleq \mathcal{N}(\mathbf{z} \,|\, \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}})$ has $\mathbf{x}$-dependent mean and covariance
- introduce another deep neural network $\mathbb{V}$, called *encoder*:

$$\big[\boldsymbol{\mu}_{\mathbf{x}} \; \boldsymbol{\Sigma}_{\mathbf{x}}\big] = h(\mathbf{x}; \mathbb{V})$$

- derive a lower bound for the intractable log-likelihood:

$$\overbrace{\ln p(\mathbf{x})}^{l(\mathbb{W},\sigma|\mathbf{x})} = \overbrace{\mathsf{KL}\Big(q(\mathbf{z}|\mathbf{x})\Big\|p(\mathbf{z}|\mathbf{x})\Big)}^{\geq 0} + \overbrace{\Big\{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\big[\ln p(\mathbf{x}|\mathbf{z})\big] - \mathsf{KL}\Big(q(\mathbf{z}|\mathbf{x})\Big\|p(\mathbf{z})\Big)\Big\}}^{L(\mathbb{W},\mathbb{V},\sigma|\mathbf{x})}$$

Variational Autoencoders (VAE): Optimization

- VAE aims to learn both $\mathbb{W}$ and $\mathbb{V}$:

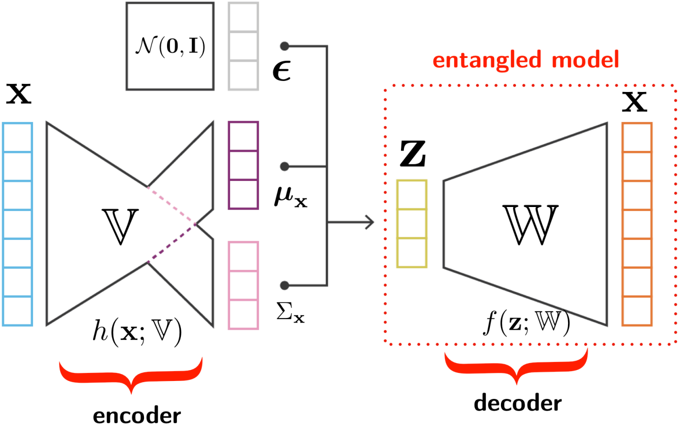
$$\arg\max_{\mathbb{W},\mathbb{V},\sigma} \sum_{i=1}^{N} L(\mathbb{W},\mathbb{V},\sigma|\mathbf{x}_i)$$

$$\Longrightarrow \arg\max_{\mathbb{W},\mathbb{V},\sigma} \sum_{i=1}^{N} \left\{ \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)}\big[\ln p(\mathbf{x}_i|\mathbf{z})\big] - \mathsf{KL}\Big(q(\mathbf{z}|\mathbf{x}_i)\Big\|p(\mathbf{z})\Big) \right\}$$

- use sampling for expectation: $\mathbf{z}_j \sim q(\mathbf{z}|\mathbf{x}_i) = \mathcal{N}(\mathbf{z}\,|\,\boldsymbol{\mu}_{\mathbf{x}_i}, \boldsymbol{\Sigma}_{\mathbf{x}_i})$, then $\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)}\big[\ln p(\mathbf{x}_i|\mathbf{z})\big] \approx \frac{1}{G}\sum_{j=1}^{G} \ln p(\mathbf{x}_i|\mathbf{z_j})$
- a reparameterization trick: $\boldsymbol{\epsilon}_j \sim \mathcal{N}(\boldsymbol{\epsilon}\,|\,\mathbf{0},\mathbf{I})$, then

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)}\big[\ln p(\mathbf{x}_i\,|\,\mathbf{z})\big] \approx \frac{1}{G}\sum_{j=1}^{G} \ln p(\mathbf{x}_i\,|\,\boldsymbol{\Sigma}_{\mathbf{x}_i}^{\frac{1}{2}}\boldsymbol{\epsilon}_j + \boldsymbol{\mu}_{\mathbf{x}_i})$$

Variational Autoencoders (VAE)



Generative Adversarial Nets (GAN)

- introduce a discriminator $\mathbb{V}$ to replace the intractable likelihood function
- use a pure sampling-based training procedure
- *equilibrium*: $\mathbb{V}$ cannot distinguish fake samples from the true samples
- equilibrium $\Longrightarrow$ $\mathbb{W}$ is a good entangled model for the data distribution