Chapter 13
Entangled Models

supplementary slides to
Machine Learning Fundamentals
© Hui Jiang 2020
published by Cambridge University Press

August 2020
Outline

1. Formulation of Entangled Models
2. Linear Gaussian Models
3. Non-Gaussian Models
4. Deep Generative Models
Borel isomorphism theorem

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

\[ p(z) \quad \text{x = f(z)} \quad p(x) \]
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 $z$, called *factors*, are entangled by a mixing function $x = f(z)$ to observations $x$
  - add residual noises to compute likelihood
  - disentangling: $x \mapsto z$

- 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)

\[ p_{\lambda}(z) \quad z \in \mathbb{R}^n \]

\[ f(z) \]

\[ x = f(z) + \varepsilon \]

\[ x \in \mathbb{R}^d \]

\[ p_{\nu}(\varepsilon) \quad \varepsilon \in \mathbb{R}^d \]

\[ \varepsilon \]

\[ p_{\Lambda}(x) \]
### Entangled Models (IV)

<table>
<thead>
<tr>
<th>Entangled Models</th>
<th>Factor $z \sim p(z)$</th>
<th>Residual $\varepsilon \sim p(\varepsilon)$</th>
<th>Mixing $f(z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probabilistic PCA</td>
<td>$\mathcal{N}(z</td>
<td>0, I)$</td>
<td>$\mathcal{N}(\varepsilon</td>
</tr>
<tr>
<td>Factor analysis</td>
<td>$\mathcal{N}(z</td>
<td>0, I)$</td>
<td>$\mathcal{N}(\varepsilon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$D$: diagonal</td>
<td></td>
</tr>
<tr>
<td>ICA</td>
<td>$\prod_i p_i(z_i)$</td>
<td>non-Gaussian</td>
<td>$Wz$ linear</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IFA</td>
<td>$\prod_i p_i(z_i)$</td>
<td>$\mathcal{N}(\varepsilon</td>
<td>0, \Lambda)$</td>
</tr>
<tr>
<td></td>
<td>factorial GMM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HOPE</td>
<td>mixture model (movMF/GMM)</td>
<td>$\mathcal{N}(\varepsilon</td>
<td>0, \sigma^2 I)$</td>
</tr>
<tr>
<td>VAE</td>
<td>$\mathcal{N}(z</td>
<td>0, I)$</td>
<td>$\mathcal{N}(\varepsilon</td>
</tr>
<tr>
<td>GAN</td>
<td>$\mathcal{N}(z</td>
<td>0, I)$</td>
<td>$-$</td>
</tr>
</tbody>
</table>
Learning of Entangled Models

1. Jacobian method: if the mixing function is invertible and differentiable
   - entangled models $\Rightarrow$ maximum likelihood estimation
     $$p_\Lambda(x) = |J| \ p_\lambda(f_1^{-1}(x)) \ p_\nu(f_2^{-1}(x))$$
   - disentangling: $z = f_1^{-1}(x)$

2. Marginalization method
   - entangled models $\Rightarrow$ maximum likelihood estimation
     $$p_\Lambda(x) = \int_z p_\lambda(z) p_\nu(x - f(z; W)) \ dz$$
   - disentangling:
     $$p(z|x) = \frac{p(z, x)}{p(x)} = \frac{p_\lambda(z) p_\nu(x - f(z; W))}{\int_z p_\lambda(z) p_\nu(x - f(z; W)) \ dz}$$
Linear Gaussian Models

- Linear Gaussian models are a group of simple entangled models.
- Factors $z$ follow a zero-mean multivariate Gaussian:
  \[ p(z) = \mathcal{N}(z \mid 0, \Sigma_1) \]
- Residual $\varepsilon$ follows another multivariate Gaussian:
  \[ p(\varepsilon) = \mathcal{N}(\varepsilon \mid \mu, \Sigma_2) \]
- The mixing function is linear:
  \[ f(z; W) = Wz \]
- Linear Gaussian models:
  \[ p_\Lambda(x) = \mathcal{N}(x \mid \mu, W\Sigma_1W^T + \Sigma_2) \]
Probabilistic PCA

- probabilistic PCA: a special case of linear Gaussian models
- factor distribution: \( p(z) = \mathcal{N}(z \mid 0, I) \)
- residual distribution: an isotropic covariance Gaussian
  \( p_\sigma(\varepsilon) = \mathcal{N}(\varepsilon \mid \mu, \sigma^2 I) \)
- probabilistic PCA models: \( p_\Lambda(x) = \mathcal{N}(x \mid \mu, WW^\top + \sigma^2 I) \)
- the log-likelihood function
  \[
  l(W, \mu, \sigma^2) = C - \frac{N}{2} \ln |WW^\top + \sigma^2 I| - \frac{1}{2} \sum_{i=1}^{N} (x_i - \mu)^\top (WW^\top + \sigma^2 I)^{-1} (x_i - \mu)
  \]
- maximum likelihood estimation \( W_{\text{MLE}} \) \( \Longrightarrow \) a rotation of principle components in regular PCA
- disentangling: \( p(z \mid x) = \mathcal{N}\left(z \mid M^{-1}W^\top(x - \mu), \sigma^{-2}M\right) \)
  where \( M = WW^\top + \sigma^2 I \)
Factor Analysis (I)

- probabilistic PCA: a special case of linear Gaussian models
- factor distribution: \( p(z) = \mathcal{N}(z \mid 0, I) \)
- residual distribution: a diagonal covariance Gaussian
  \[
  p(\varepsilon) = \mathcal{N}(\varepsilon \mid \mu, D)
  \]
  where \( D \in \mathbb{R}^{d \times d} \) is a diagonal covariance matrix
- factor analysis models:
  \[
  p_\Lambda(x) = \mathcal{N}(x \mid \mu, WW^T + D)
  \]
- the log-likelihood function:
  \[
  l(W, D) = C - \frac{N}{2} \left[ \ln | WW^T + D| + \text{tr}\left((WW^T + D)^{-1} S\right) \right]
  \]
  - no closed-form solution for maximum likelihood estimation
Factor Analysis (II)

alternating MLE method

**Input:** the sample covariance matrix $S$

**Output:** $W$ and $D$

randomly initialize $D_0$; set $t = 1$

**while** not converged **do**

1. construct $P_t$ using the $n$ leading eigenvectors of $D_{t-1}^{-\frac{1}{2}}SD_{t-1}^{-\frac{1}{2}}$

2. $W_t = D_t^{\frac{1}{2}}P_t$

3. $D_t = \text{diag}\left(S - W_tW_t^T\right)$

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

- Independent Factor Analysis (IFA)
  \[ p(z_j) = \sum_{m=1}^{M} w_{jm} \mathcal{N}(z_j \mid \mu_{jm}, \sigma_{jm}^2) \]

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

Figure: the normal distribution vs. the heavy-tail distribution \( p(z) \) in ICA
ICA is used for blind source separation

\[
z = W^{-1}x
\]

the log-likelihood function:

\[
l(W^{-1}) = \sum_{i=1}^{N} \sum_{j=1}^{n} \ln p(w_j^T x_i) + N \ln |W^{-1}|
\]

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

- combine a simple Gaussian with a complex non-linear mixing function
- use a deep neural network $W$ for the mixing function:
  \[ x = f(z; W) \]
- cannot explicitly evaluate the likelihood function
- no direct way to disentangle: $p(z|x)$
- 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 $q(z|x)$ to approximate the intractable distribution $p(z|x)$ as:
  $$p(z|x) \approx q(z|x)$$

- $q(z|x) \triangleq \mathcal{N}(z | \mu_x, \Sigma_x)$ has $x$-dependent mean and covariance

- Introduce another deep neural network $V$, called encoder:
  $$[\mu_x, \Sigma_x] = h(x; V)$$

- Derive a lower bound for the intractable log-likelihood:
  $$\ln p(x) = KL( q(z|x) \| p(z|x) ) + \left\{ \mathbb{E}_{q(z|x)} [ \ln p(x|z) ] - KL( q(z|x) \| p(z) ) \right\}$$
Variational Autoencoders (VAE): Optimization

- VAE aims to learn both \(W\) and \(V\):

\[
\arg \max_{W,V,\sigma} \sum_{i=1}^{N} L(W, V, \sigma|x_i)
\]

\[
\implies \arg \max_{W,V,\sigma} \sum_{i=1}^{N} \left\{ \mathbb{E}_{q(z|x_i)} \left[ \ln p(x_i|z) \right] - KL \left( q(z|x_i) || p(z) \right) \right\}
\]

- use sampling for expectation: \(z_j \sim q(z|x_i) = \mathcal{N}(z | \mu_{x_i}, \Sigma_{x_i})\), then

\[
\mathbb{E}_{q(z|x_i)} \left[ \ln p(x_i|z) \right] \approx \frac{1}{G} \sum_{j=1}^{G} \ln p(x_i|z_j)
\]

- a reparameterization trick: \(\epsilon_j \sim \mathcal{N}(\epsilon | 0, I)\), then

\[
\mathbb{E}_{q(z|x_i)} \left[ \ln p(x_i|z) \right] \approx \frac{1}{G} \sum_{j=1}^{G} \ln p(x_i | \Sigma_{x_i}^{1/2} \epsilon_j + \mu_{x_i})
\]
Variational Autoencoders (VAE)
Generative Adversarial Nets (GAN)

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