Reading Note for

A comprehensive survey and analysis of generative models in machine learning

This paper mainly discussed the algos and implemenatations of different categories of generative models(GM).

- Gaussian Mixture Models (GMM)
- Hidden Markov Models (HMM)
- Latent Dirichlet Allocation (LDA)
- Boltzmann Machines
    - Restricted Boltzmann Machines (RBM)
    - Deep Belief Networks (DBN)
    - Deep Boltzmann Machines (DBM)
- Variational Autoencoders (VAE)
- Generative Adversarial Networks (GANs)

$\Rightarrow$ Discreminative models

- draw the decision boundary in a data space;
- estimate P(Y|X) -- Y is the target.

$\Rightarrow$ Generative models

- learn the overall distribution of the data;
- estimate the distribution given by P(X|Y) and P(Y) -- X is the target

GMM

Different Gaussians are represented by different $\{\pi_i, \mu_i, \rho_i^2\}$ tuples, with parameters meaning weight, mean, and variance. GMM is a mixture of them. Each Gaussian can be seen as a cluster, with:

$$\sum_{i=1}^{n} \pi_i = 1$$

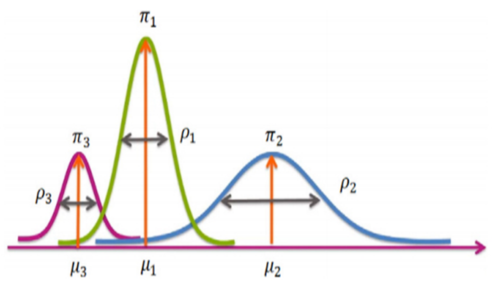


Fig. 3. Mixture of three Gaussians where π signifies the weight associated to the Gaussian and hence also the probability of the data belonging to the *i*th cluster or Gaussian, μ specifies the position of the Gaussian with the mean, ρ signifies the 'spread' of the Gaussian over the overall distribution by the variance.

If we are given that z_k(a latent that sampled from a Gaussian) is from a cluster i, the likelihood of observing x_k is given as,

$$P(x_k|z_k = i, \mu_i, \rho_i) = N(x_k|\mu_i, \rho_i)$$

Thus, for a N component Gaussian mixture model:

$$P(x|\{\pi_k, \mu_k, \rho_k^2\}) = \sum_{k=1}^{N} w_i b(x|u_k, p_k)$$

Which indicates the probability of the generated x is sampled from each single cluster.

HMM

A probabilistic representation of a system it models.
It is used for modeling linear problems involving time series or sequences.

The probability of reaching the next state is dependent on the transition function of the current state. Each state emits residues or symbols based on their symbol-emission probabilities. The Markov chain(state sequence) that led to these emissions are not observable.

VAE

- **Autoencoder**: unsupervised training with unlabelled data

  $$\mathbf{Z} = f(\mathbf{W}\mathbf{X} + b)$$

  $$L = ||x - \hat{x}||^2$$

  $\mathbf{Z}$ is a compressed important feature

- **Variational Autoencoders**: generating data $\mathbf{x}$ from a unknown latent distribution z.

  Assume the data $\{x^i\}_{i=1}^N$ is generated by a true prior latent distribution z(a Gaussian) given by $p_\theta(z)$ where θ are the parameters of the model.

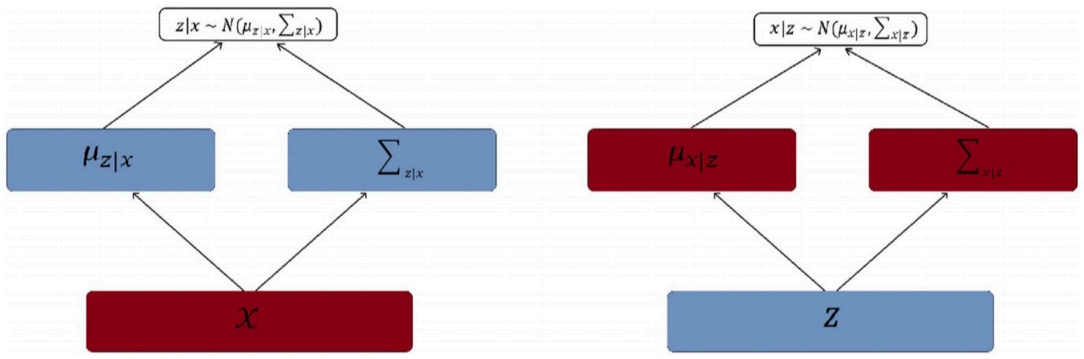
The true condition is $p_\theta(\mathbf{x}|z^i)$ but intractable.

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})dz$$

Because we don't know the $p_\theta(\mathbf{x})$, the posterior density is also intractable.

$$p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{x})}$$

However, we can **approximate** $p_\theta(\mathbf{z}|\mathbf{x})$ through an inference network $q_\phi(\mathbf{z}|\mathbf{x})$ which allows us to **derive a tractable lower bound** which can be maximized by proper optimization.



$z|x \sim N(\mu_{z|x}, \Sigma_{z|x})$
$\mu_{z|x}$
$\Sigma_{z|x}$
x
$x|z \sim N(\mu_{x|z}, \Sigma_{x|z})$
$\mu_{x|z}$
$\Sigma_{x|z}$
z

Fig. 14. (**Left**): The inference network represented by $q_\varphi(\boldsymbol{z}|\boldsymbol{x})$ which outputs mean and diagonal covariance vectors of $\boldsymbol{z}|\boldsymbol{x}$ from which we sample $\boldsymbol{z}|\boldsymbol{x}$. (**Right**): The generator network (or decoder) represented by $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ which also output the mean and diagonal covariance vectors of $\boldsymbol{x}|\boldsymbol{z}$ from which the distribution $\boldsymbol{x}|\boldsymbol{z}$ can be sampled.

$(\mathbf{Left})$ the inference network $q_\phi(\mathbf{z}|\mathbf{x})$ | $(\mathbf{Right})$ the generator network $p_\theta(\mathbf{x}|\mathbf{z})$.

Their outputs are the means and the diagonal covariances.

With $p_\theta(x^i)$ independent of z, the logarithm of the data likelihood can then be expressed as:

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|x^i)}[\log p_\theta(x^i)] &= \mathbb{E}\mathbf{z}\left[\log \frac{p_\theta(x^i|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{z}|x^i)}\right] \\
&= \mathbb{E}\mathbf{z}\left[\log \frac{p_\theta(x^i|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{z}|x^i)}\frac{q_\phi(\mathbf{z}|x^i)}{q_\phi(\mathbf{z}|x^i)}\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\log p_\theta(x^i|\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}}\left[\log \frac{q_\phi(\mathbf{z}|x^i)}{p_\theta(\mathbf{z})}\right] + \mathbb{E}_{\mathbf{z}}\left[\log \frac{q_\phi(\mathbf{z}|x^i)}{p_\theta(\mathbf{z}|x^i)}\right] \\
&= \mathbb{E}_{\mathbf{z}}\left[\log p_\theta(x^i|\mathbf{z})\right] - KL(q_\phi(\mathbf{z}|x^i)||p_\theta(\mathbf{z})) + KL(q_\phi(\mathbf{z}|x^i)||p_\theta(\mathbf{z}|x^i))
\end{aligned}$$

Concept: KL divergence

Then using **term I** and **term II** to get a tractable lower bound $\epsilon(x^i, \theta, \phi) \leq \log p_\theta(x^i)$ with its gradient available.

- term I $\Rightarrow$ uses reparameterization trick to make it differentiable & describes the reconstruction of input data.
- term II $\Rightarrow$ the posterior distribution must be as similar as possible to the prior.
- term III ≥ 0

Therefore, while training, we attempt to estimate the parameters θ' and ϕ' by maximizing $\epsilon(x^i, \theta, \phi)$ as

$$\theta', \phi' = \mathrm{argmax}_{\theta,\phi} \sum_{i=1}^{N} \epsilon(x^i, \theta, \phi)$$

GAN

- VAE: not the most accurate; blurriness in the generated images
- GAN: w/o the usual procedure of maximizing a log likelihood (**term II** in VAE); doesn't require Markov chains.
    - Deep Convolutional GANs (DCGANs),
    - Fully Connected and Convolutional GANs (FCC-GANs)
    - Conditional GANs (CGANs)
    - Stack GANs (SGAN)

Important Concepts

The Kullback-Leibler (KL) Divergence

is used to measure the similarity between two density functions $p(x)$ and $q(x)$, which is always ≥ 0 and is 0 iff $p = q$. It is defined by:

$$D(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$