Gaussian Process Prior Variational Autoencoder

Accepted in NeurIPs 2018

Kyeong Ryeol, Go
M.S. Candidate of OSI Lab

Contents

- VAE [1]
- Beta-VAE [2]
- GPP-VAE [3]

[1] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114* (2013).
[2] Higgins, Irina, et al. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework." *ICLR* 2.5 (2017): 6.
[3] Casale, Francesco Paolo, et al. "Gaussian process prior variational autoencoders." *Advances in Neural Information Processing Systems*. 2018.

VAE Intro

- Aim

Performing efficient approximate inference on latent variables that has intractable posterior

- Contribution

Introduce a differentiable unbiased estimator of variational lower bound

 - Stochastic Gradient Variational Bayes(SGVB) estimators

Introduce an algorithm to learn the associated parameters in mini-batch unit

 - Auto-Encoding Variational Bayes(AEVB) algorithm

SGVB / AEVB

- $\log p(x) = \log \int p(x,z)\,dz = \log \int q(z|x)\,\frac{p(x,z)}{q(Z|X)}\,dz$

$$\geq \int q(z|x) \log \frac{p(x,z)}{q(Z|X)}\,dz \qquad (Jensen's\ inequality)$$

$$= \int q(z|x) \log p(x|z) + q(z|x) \log \frac{p(z)}{q(Z|X)}\,dz$$

$$= \mathbb{E}_{z\sim q(z|x)}[\log p(x|z)] - KL\big(q(z|x)\big|\big|p(z)\big) \qquad (ELBO)$$

- $\log p(X) \approx \frac{N}{M}\sum_{i=1}^{M}[\frac{1}{L}\sum_{l=1}^{L}\log p\left(x^i\middle|z^{i,l}\right) + \frac{1}{2}\sum_{j=1}^{J}(1+\log\left({\sigma_j^{(i)}}^2\right) - {\mu_j^{(i)}}^2 - {\sigma_j^{(i)}}^2]$

$$where \quad z^{(i,l)} = \mu^{(i)} + \sigma^{(i)} \odot \varepsilon^{(l)} \qquad \varepsilon^{(l)} \sim N(0,I) \qquad (Reparameterization)$$

$$when \quad q\left(z\middle|x^{(i)}\right) = \mathcal{N}\left(\mu^{(i)}, {\sigma^{(i)}}^2 I\right) \qquad p(z) = N(0,I)$$

N : *total data points*, M (≈ 100): *batch size*, L (≈ 1): *number of samples of* z *on each* x^i, J: $\dim(z)$

Learning a disentangled representation

- Where is it useful?
  - supervised learning
  - reinforcement learning
  - transfer learning
  - zero-shot learning

- Problem : the definition of disentanglement is still open to debate

Bengio et al (2013)
"A representation where a change in one dimension corresponds to a change in one factor of variation, while being relatively invariant to changes in other factors"

Beta-VAE Intro

- Aim

Learning a disentangled factor in a purely unsupervised manner augmenting VAE framework

- Contribution

Introduce a constrained optimization problem

Devise a metric to quantify the degree of disentanglement

Formulation

$$\max_{\theta} \mathbb{E}_{p_\theta(z)}[\log p_\theta(x|z)] \geq \max_{\phi,\theta} \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z))$$

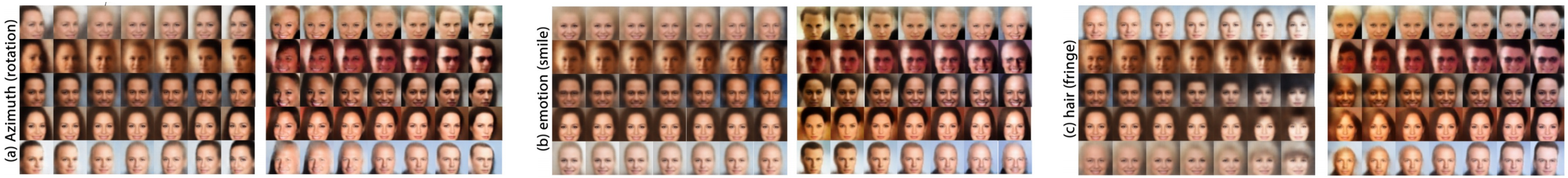
$$\rightarrow \max_{\phi,\theta} \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z))$$
$$s.t. \qquad KL(q_\phi(z|x)||p(z)) < \varepsilon$$

$$\rightarrow \min_{\lambda} max_{\phi,\theta}\, \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - (\lambda + 1) * KL(q_\phi(z|x)||p(z))$$
$$s.t. \qquad \lambda \geq 0$$

Disentangled representations can be captured when the right balance is found between data reconstruction and latent regularization

Analysis

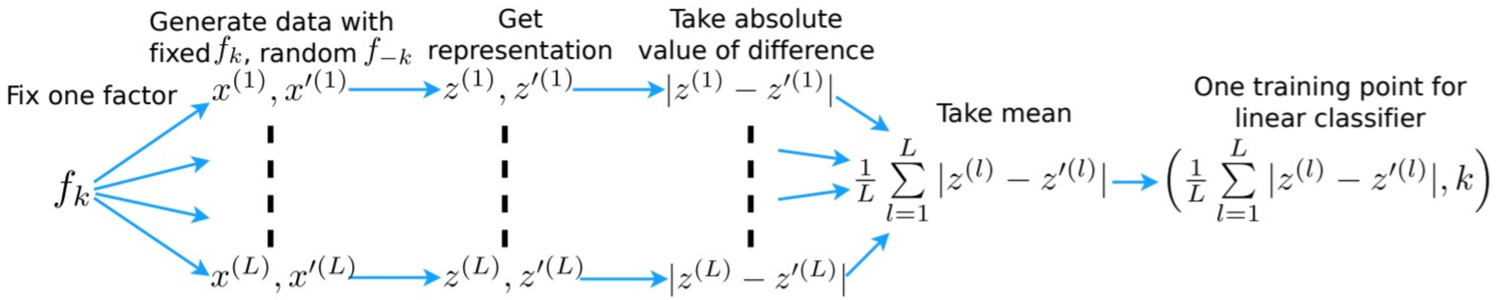
- Qualitative analysis



- Quantitative analysis

Independence : correlation between latent dimensions

Interpretability : robust classification even with simple classifier



Extensions

- Other interpretation

  [4] Burgess, Christopher P., et al. "Understanding disentangling in β-VAE." *arXiv preprint arXiv:1804.03599* (2018).

  [5] Mathieu, Emile, et al. "Disentangling Disentanglement." *arXiv preprint arXiv:1812.02833* (2018).

- Several other variants
  - Beta-TCVAE [6]
  - Factor-VAE [7]
  - DIP-VAE [8]

[6] Chen, Tian Qi, et al. "Isolating sources of disentanglement in variational autoencoders." *Advances in Neural Information Processing Systems*. 2018.

[7] Kim, Hyunjik, and Andriy Mnih. "Disentangling by factorising." *arXiv preprint arXiv:1802.05983* (2018).

[8] Kumar, Abhishek, Prasanna Sattigeri, and Avinash Balakrishnan. "Variational inference of disentangled latent concepts from unlabeled observations." *arXiv preprint arXiv:1711.00848* (2017).

GPP-VAE Intro

- Motivation
  - Prior assumption that latent encodings are i.i.d. across dimensions and samples does not fit to real-world problem
  - Accounting for covariances between samples can yield a better model
- How
  - Combine VAE with Gaussian Process prior
  - Leveraging the auxiliary data
- Aim
  - Model the relationship between the latent encodings and the auxiliary data
  - Disentangle sample correlations induced by different auxiliary data
  - Predict the latent codes when auxiliary data is unobserved
  - Generate the data for any configuration of the auxiliary data

Settings

- Two auxiliary data : Object & view
  - images of faces with different poses / images of digits with different rotation
- Notation
  - $\{y_n\}_{n=1}^{N}$: K-dimensional data for N samples $\rightarrow$ $Y \in \mathbb{R}^{N\times K}$
  - $\{z_n\}_{n=1}^{N}$: L-dimensional latent representation for the N samples $\rightarrow$ $Z \in \mathbb{R}^{N\times L}$
  - $\{x_p\}_{p=1}^{P}$: M-dimensional object feature vectors for the P unique objects $\rightarrow$ $X \in \mathbb{R}^{P\times M}$
  - $\{w_q\}_{q=1}^{Q}$: R-dimensional view feature vectors for the Q unique views $\rightarrow$ $W \in \mathbb{R}^{Q\times R}$
  - f_θ : $\mathbb{R}^M \times \mathbb{R}^R \rightarrow \mathbb{R}^L$: function that maps auxiliary data to latent representation
  - g_ϕ : $\mathbb{R}^L \rightarrow \mathbb{R}^K$: function that maps latent representation to high dimensional sample space
  - $\mathcal{K}_\theta(X, W)$: $N\times N$ latent covariance
    where $\mathcal{K}_\theta(X,W)_{ij} = \mathcal{K}_\theta^{(\text{object})}\left(\mathrm{x}_{\mathrm{p_i}}, x_{p_j}\right)\mathcal{K}_\theta^{(\text{view})}\left(\mathrm{w}_{\mathrm{q_i}}, \mathrm{w}_{\mathrm{q_j}}\right)$ for $\mathrm{i}, j \in \{1, \dots, N\}$

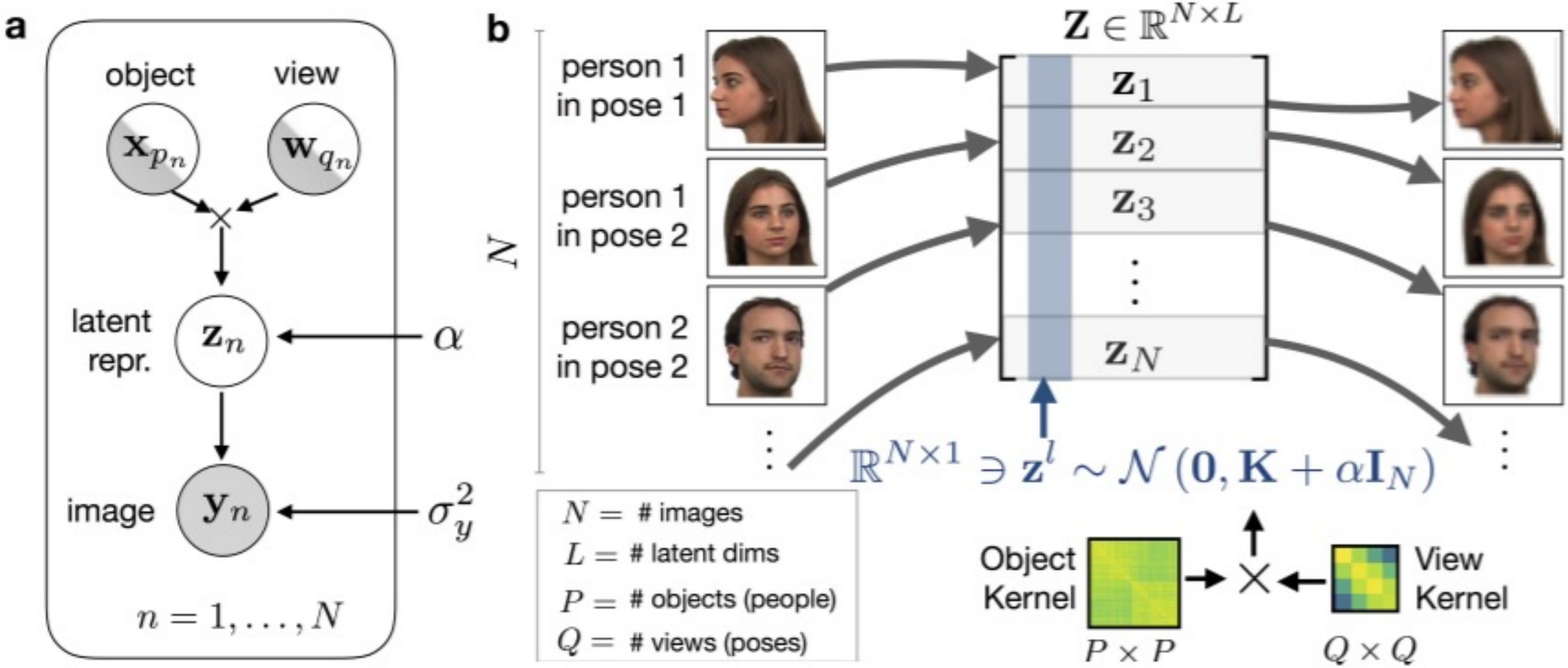
Model construction

- Generative process
  - $z_n = f_\theta\left(x_{p_n}, w_{q_n}\right) + \eta_n \quad where \quad \eta_n \sim \mathcal{N}(0, \alpha I_L)$
  - $y_n = g_\phi(z_n) + \varepsilon_n \quad where \quad \varepsilon_n \sim \mathcal{N}\left(0, \sigma_y^2 I_K\right)$
  - $p\left(Y \middle| Z, \phi, \sigma_y^2\right) = \prod_{n=1}^{N} \mathcal{N}\left(y_n \middle| g_\phi(z_n), \sigma_y^2 I_K\right)$ $(Decoder)$

*** Gaussian Process model

- $p(Z|X, W, \theta, \alpha\ ) = \prod_{l=1}^{L} \mathcal{N}(z^l|0, \mathcal{K}_\theta(X, W) + \alpha I_N)$
  where z^l : $lth\ column\ of\ Z$

- Inference process
  - $q_\psi(Z|Y) = \prod_{n=1}^{N} \mathcal{N}\left(z_n \middle| \mu_\psi^z(y_n), diag\left({\sigma_\psi^z}^2(y_n)\right)\right)$ $(Encoder)$



a
object
view
$\mathbf{x}_{p_n}$
$\mathbf{w}_{q_n}$
latent repr.
$\mathbf{z}_n$
α
image
$\mathbf{y}_n$
σ_y^2
$n = 1, \ldots, N$
b
N
person 1 in pose 1
person 1 in pose 2
person 2 in pose 2
$\mathbf{Z} \in \mathbb{R}^{N \times L}$
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_3$
$\mathbf{z}_N$
$\mathbb{R}^{N \times 1} \ni \mathbf{z}^l \sim \mathcal{N}(\mathbf{0}, \mathbf{K} + \alpha \mathbf{I}_N)$
$N =$ # images
$L =$ # latent dims
$P =$ # objects (people)
$Q =$ # views (poses)
Object Kernel
$P \times P$
View Kernel
$Q \times Q$

ELBO

$$
\begin{aligned}
&\log p\left(Y \middle| X, W, \theta, \alpha, \phi, \sigma_y^2\right) \\
&= \log \int p\left(Y \middle| Z, \phi, \sigma_y^2\right) p(Z|X, W, \theta, \alpha)\, dZ \\
&= \log \int q_\psi(Z|Y) \frac{p\left(Y \middle| Z, \phi, \sigma_y^2\right) p(Z|X, W, \theta, \alpha)}{q_\psi(Z|Y)}\, dZ \\
&\geq \int q_\psi(Z|Y) \log\left(\frac{p\left(Y \middle| Z, \phi, \sigma_y^2\right) p(Z|X, W, \theta, \alpha)}{q_\psi(Z|Y)}\right) dZ \\
&= \ \mathbb{E}_{Z \sim q_\psi}\left[\log p(Y|Z, \phi, \sigma_y^2) + \log p(Z|X, W, \theta, \alpha)\right] - \int q_\psi(Z|Y) \log q_\psi(Z|Y)\, dZ \\
&= \ \mathbb{E}_{Z \sim q_\psi}\left[\sum_{n=1}^{N} \log \mathcal{N}\left(y_n \middle| g_\phi(z_n), \sigma_y^2 I_K\right) + \sum_{l=1}^{L} \log \mathcal{N}(z^l | 0, \mathcal{K}_\theta(X, W) + \alpha I_N)\right] + \frac{1}{2}\sum_{n,l} \log {\sigma_\psi^z}^2(y_n)_l + const. \\
&\approx \ \sum_{n=1}^{N} \log \mathcal{N}\left(y_n \middle| g_\phi\left(z_{\psi_n}\right), \sigma_y^2 I_K\right) + \sum_{l=1}^{L} \log \mathcal{N}(z_\psi^l | 0, \mathcal{K}_\theta(X, W) + \alpha I_N) + \frac{1}{2}\sum_{n,l} \log {\sigma_\psi^z}^2(y_n)_l + const.
\end{aligned}
$$

$$
where\ \ z_{\psi_n} = \mu_\psi^z(y_n) + \varepsilon_n \odot {\sigma_\psi^z}^2(y_n) \qquad \varepsilon_n \sim N(0, I_L)
$$

$$
when\ \ q_\psi(z_n|y_n) = \mathcal{N}\left(z_n \middle| \mu_\psi^z(y_n), diag\left({\sigma_\psi^z}^2(y_n)\right)\right)
$$

Loss function

- $\mathcal{L}(\phi, \theta, \alpha, \psi)$

$= N\sigma_y^2 \log \sigma_y^2 + \frac{1}{K}\sum_{n=1}^{N} \left\| y_n - g_\phi(z_{\psi_n}) \right\|_2^2 - \lambda \frac{1}{L}\left[\sum_{l=1}^{L} \log \mathcal{N}(z_\psi^l | 0, \mathcal{K}_\theta(X, W) + \alpha I_N) + \frac{1}{2}\sum_{n,l} \log {\sigma_\psi^z}^2 (y_n)_l\right]$

where λ is a hyperparameter for balancing data reconstruction and latent regularization

(selected via cross-validation on standard VAE → maximal ELBO in validation set)

where σ_y^2 is estimated on validation set with selected λ as follow

$$\sigma_y^2 = \frac{1}{N^{(val)}} \sum_{n=1}^{N^{(val)}} \left(y_n^{(val)} - g_{\phi_{\hat{\lambda}}}\left(z_{\psi_{\hat{\lambda}n}}^{(val)} \right) \right)^2$$

where $\left(\phi_{\hat{\lambda}}, \psi_{\hat{\lambda}}\right)$ are the values of the encoder/decoder parameters in VAE trained with $\lambda = \hat{\lambda}$

Problems

- Challenge
  - Unbiasedness of mini-batch gradient estimates no longer holds
  - Gaussian Process requires a lot of computation $\approx O(n^3)$
- Aim
  - Calculate gradients on the whole dataset in a low-memory fashion
  - Achieve linear computations in the number of samples $\approx O(n)$
- How
  - Low rank approximation of Gaussian Process kernel
  - First-order Taylor series expansion on the Gaussian Process term of the loss

Low rank approximation

- $\mathcal{N}\left(z_\psi^l \middle| 0, \mathcal{K}_\theta(X, W) + \alpha I_N\right) = \mathcal{N}\left(z^l \middle| 0, VV^T + \alpha I_N\right) = N\left(z^l \middle| 0, C\right) \quad where\ \ V \in \mathbb{R}^{N \times H},\ \ H \ll N$
    - Woodbury identity

$$
\begin{aligned}
&1.\ (I+P)^{-1} = (I+P-P)(I+P)^{-1} = I - P(I+P)^{-1} \\
&2.\ (I+PQ)P = P(I+QP) \quad \rightarrow \quad P(I+QP)^{-1} = (I+PQ)^{-1}P
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow C^{-1}M &= (VV^T + \alpha I_N)^{-1}M \qquad where\ M \in \mathbb{R}^{N \times K} \\
&= \frac{1}{\alpha}\left(I_N + \frac{1}{\alpha}VV^T\right)^{-1} M \\
&= \frac{1}{\alpha}\left(M - \frac{1}{\alpha}VV^T\left(I_N + \frac{1}{\alpha}VV^T\right)^{-1} M\right) \qquad (\because 1.) \\
&= \frac{1}{\alpha}\left(M - \frac{1}{\alpha}V\left(I_H + \frac{1}{\alpha}V^TV\right)^{-1} V^TM\right) \qquad (\because 2.)
\end{aligned}
$$

$$
\therefore NK + NHK + NH^2 + H^3 + H^2 + NH^2 + NHK \ = O(H^3 + NH^2 + NHK) \quad \rightarrow \quad linear\ in\ \ N
$$

Low rank approximation (cont.)

- $\mathcal{N}\left(z_\psi^l \middle| 0, \mathcal{K}_\theta(X,W) + \alpha I_N\right) = \mathcal{N}\left(z^l \middle| 0, VV^T + \alpha I_N\right) = N\left(z^l \middle| 0, C\right) \quad where\ \ V \in \mathbb{R}^{N\times H},\ H \ll N$
  - Determinant Lemma

1. $\det\begin{pmatrix}\alpha I_H & -V^T \\ V & I_N\end{pmatrix}\begin{pmatrix}I_H & V^T \\ 0 & \alpha I_N\end{pmatrix} = \det\begin{pmatrix}\alpha I_H & 0 \\ V & VV^T + \alpha I_N\end{pmatrix} = \det(\alpha I_H)\det(VV^T + \alpha I_N)$

2. $\det\begin{pmatrix}I_H & V^T \\ 0 & \alpha I_N\end{pmatrix}\begin{pmatrix}\alpha I_H & -V^T \\ V & I_N\end{pmatrix} = \det\begin{pmatrix}\alpha I_H + V^T V & 0 \\ \alpha V & \alpha I_N\end{pmatrix} = \det(\alpha I_N)\det(\alpha I_H + V^T V)$

$$\Rightarrow \alpha^H \det(\alpha I_N + VV^T) = \alpha^N \alpha^H \det\left(I_H + \frac{1}{\alpha}V^T V\right)$$
$$\Rightarrow \log\det(\alpha I_N + VV^T) = N\log\alpha + \log\det(I_H + \frac{1}{\alpha}V^T V)$$

$$\therefore H^3 + H^2 + NH^2 = O(H^3 + NH^2) \quad \rightarrow \quad linear\ in\ \ N$$

Taylor series expansion

- $f(z_\psi^l, V, \alpha) \coloneqq \log \mathcal{N}(z_\psi^l | 0, \mathcal{K}_\theta(X, W) + \alpha I_N) = -\frac{N}{2} \log \det C - \frac{1}{2} {z_\psi^l}^T C^{-1} z_\psi^l + const.$

  $= a^T z_\psi^l + tr(B^T V) + c\alpha \qquad (First - order\ Taylor\ series\ expansion)$

$$where \quad a = \left(\frac{\partial f}{\partial z_\psi^l}\right)_{\xi_0} = -\left(C^{-1} z_\psi^l\right)_{\xi_0}$$

$$B = \left(\frac{\partial f}{\partial V}\right)_{\xi_0} = -\left(N C^{-1} V - C^{-1} z_\psi^l {z_\psi^l}^T C^{-1} V\right)_{\xi_0}$$

$$c = \left(\frac{\partial f}{\partial \alpha}\right)_{\xi_0} = -\frac{1}{2}\left(N\, Tr(C^{-1}) - z_\psi^l C^{-1} C^{-1} z_\psi^l\right)_{\xi_0}$$

$where \quad \xi_0 = \{\psi_0, \theta_0, \alpha_0\}$ is the set of parameter values at certain iteration

⇒ The GP term can be expressed in linear manner by latent representation z_ψ^l

⇒ Locally it has the same gradient as the original loss

∴The gradient can be easily accumulated across mini-batches making this step memory efficient

Training process

- Step1
  - Compute latent encodings from the high-dimensional data in a mini-batch unit
- Step2
  - Compute the coefficients of Taylor series expansion with the encodings
- Step3
  - Computes a proxy loss by replacing the GP term by Taylor series expansion
- Step4
  - Accumulated the gradients across data mini-batches
- Step5
  - Update the parameters using the full gradients as in standard gradient descent

Prediction of latent representation

- $z^l|X,W \sim \mathcal{N}(0, \mathcal{K}_\theta(X,W) + \alpha I_N)$

$$\sim \mathcal{N}\left(0, \begin{bmatrix} \mathcal{K}_\theta^{object}(x_{p_1}, x_{p_1})\mathcal{K}_\theta^{view}(w_{q_1}, w_{q_1}) + \alpha & \cdots & \mathcal{K}_\theta^{object}(x_{p_1}, x_{p_n})\mathcal{K}_\theta^{view}(w_{q_1}, w_{q_n}) \\ \vdots & \ddots & \vdots \\ \mathcal{K}_\theta^{object}(x_{p_n}, x_{p_1})\mathcal{K}_\theta^{view}(w_{q_n}, w_{q_1}) & \cdots & \mathcal{K}_\theta^{object}(x_{p_n}, x_{p_n})\mathcal{K}_\theta^{view}(w_{q_n}, w_{q_n}) + \alpha \end{bmatrix}\right)$$

- $\begin{bmatrix} z^l \\ z_* \end{bmatrix} |X, W, X_*, W_* \sim \mathcal{N}(0, \begin{bmatrix} \mathcal{K}_\theta(X,W) + \alpha I_N & k(X,W,X_*,W_*) \\ k(X_*,W_*,X,W) & \mathcal{K}_\theta(X_*,W_*) + \alpha I_N \end{bmatrix}$

$$where \quad k(X_1,W_1,X_2,W_2) = \begin{bmatrix} \mathcal{K}_\theta^{object}\left(x_{1_{p_1}}, x_{2_{p_1}}\right)\mathcal{K}_\theta^{view}\left(w_{1_{q_1}}, w_{2_{q_1}}\right) & \cdots & \mathcal{K}_\theta^{object}\left(x_{1_{p_1}}, x_{2_{p_n}}\right)\mathcal{K}_\theta^{view}\left(w_{1_{q_1}}, w_{2_{q_n}}\right) \\ \vdots & \ddots & \vdots \\ \mathcal{K}_\theta^{object}\left(x_{1_{p_n}}, x_{2_{p_1}}\right)\mathcal{K}_\theta^{view}\left(w_{1_{q_n}}, w_{2_{q_1}}\right) & \cdots & \mathcal{K}_\theta^{object}\left(x_{1_{p_n}}, x_{2_{p_n}}\right)\mathcal{K}_\theta^{view}\left(w_{1_{q_n}}, w_{2_{q_n}}\right) \end{bmatrix}$$

- $z_*|\mathrm{z^l}, \mathrm{X}, \mathrm{W}, \mathrm{X}_*, \mathrm{W}_* \sim \mathcal{N}(\mu_{z_*}, \Sigma_{z_*})$

$$where \quad \mu_{z_*} = k(X_*,W_*,X,W)\ (\mathcal{K}_\theta(X,W) + \alpha I_N)$$

$$\Sigma_{z_*} = \mathcal{K}_\theta(X_*,W_*) + \alpha I_N - k(X_*,W_*,X,W)\ (\mathcal{K}_\theta(X,W) + \alpha I_N)^{-1} k(X,W,X_*,W_*)$$

Prediction on generated data

- $p(y_*|x_*, w_*, Y, X, W)$

$$= \frac{p(y_*, Y|x_*, w_*, X, W)}{p(Y|X, W)}$$

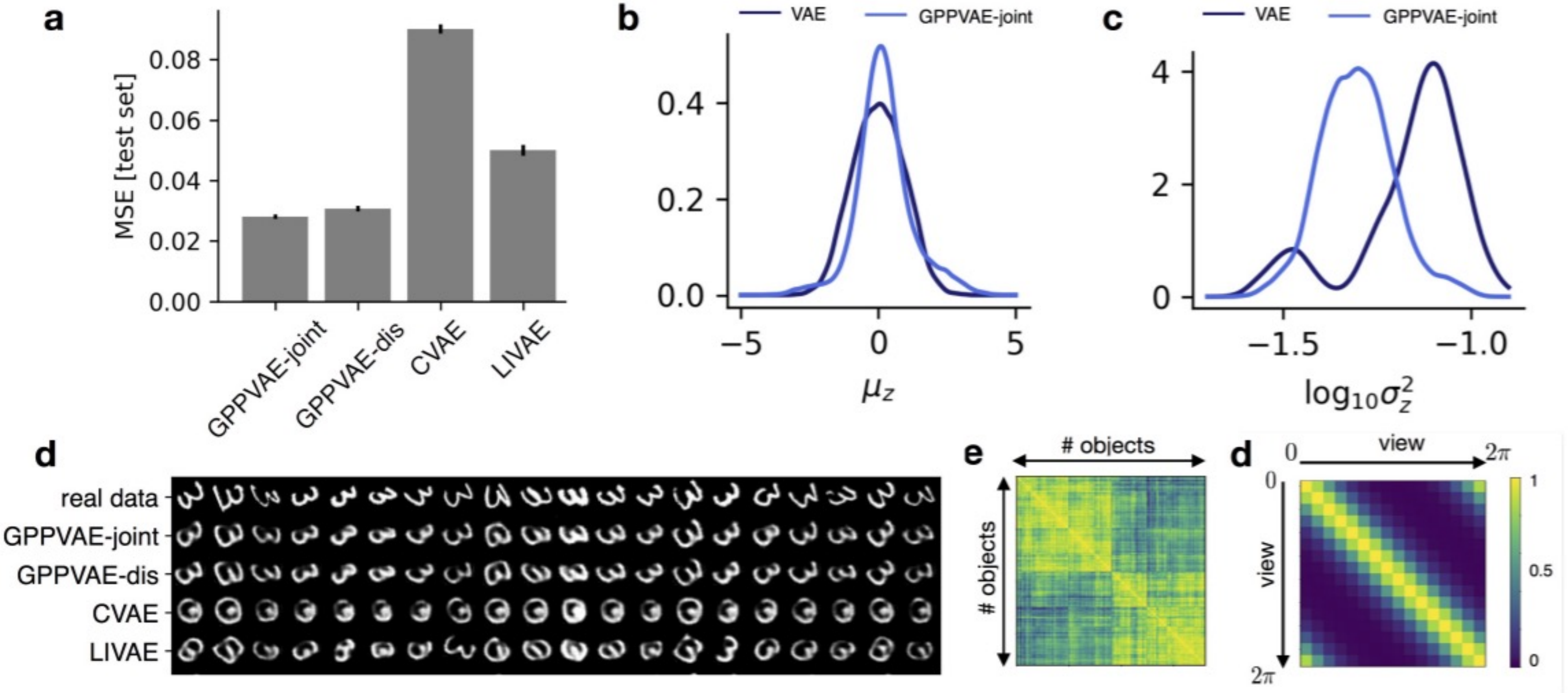
$$= \frac{1}{p(Y|X, W)} \int p(y_*|z_*)\, p(Y|Z)\, p(z_*, Z|x_*, w_*, X, W)\, dz_* dZ$$

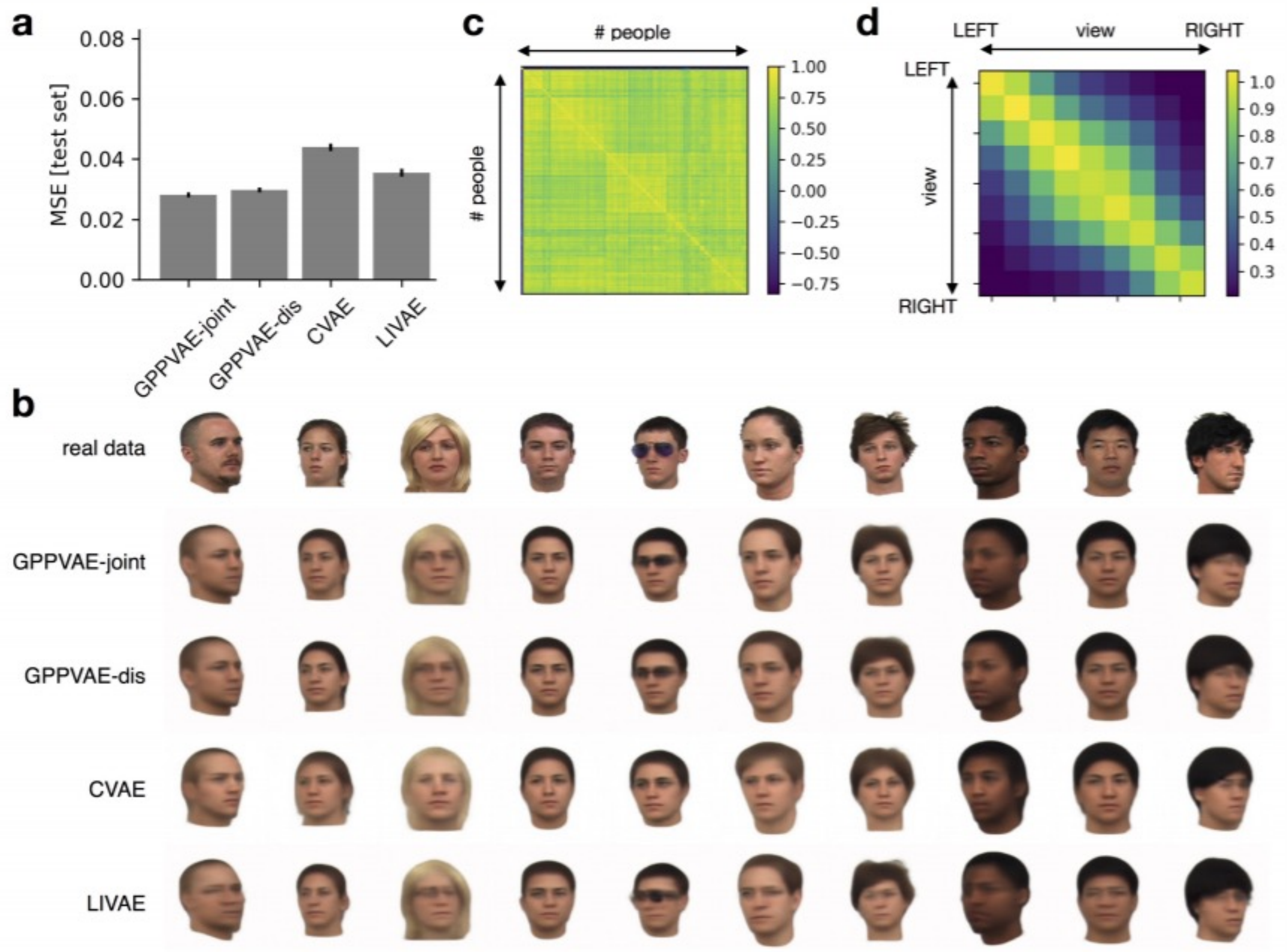
$$= \frac{1}{P(Y|X, W)} \int p(y_*|z_*)\, p(Y|Z)\, p(z_*|x_*, w_*, Z, X, W)\, p(Z|X, W)\, dz_* dZ$$

$$= \int p(y_*|z_*)\, p(z_*|x_*, w_*, Z, X, W)\, \frac{p(Y|Z)\, p(Z|X, W)}{p(Y|X, W)}\, dz_* dZ$$

$$= \int p(y_*|z_*)\, p(z_*|x_*, w_*, Z, X, W)\, p(Z|Y, X, W)\, dz_* dZ$$

$$\approx \int p(y_*|z_*)\, p(z_*|x_*, w_*, Z, X, W)\, q(Z|Y)\, dz_* dZ$$


a
MSE [test set]
0.00
0.02
0.04
0.06
0.08
GPPVAE-joint
GPPVAE-dis
CVAE
LIVAE
b
VAE
GPPVAE-joint
0.0
0.2
0.4
−5
0
5
μ_z
c
VAE
GPPVAE-joint
0
2
4
−1.5
−1.0
$\log_{10}\sigma_z^2$
d
real data
GPPVAE-joint
GPPVAE-dis
CVAE
LIVAE
e
objects
objects
d
view
view
0
2π
0
2π
1
0.5
0


a
MSE [test set]
0.08
0.06
0.04
0.02
0.00
GPPVAE-joint
GPPVAE-dis
CVAE
LIVAE
c
people
people
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
d
LEFT
view
RIGHT
LEFT
view
RIGHT
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
b
real data
GPPVAE-joint
GPPVAE-dis
CVAE
LIVAE

Conclusion

- VAE
  - Introduced improved and general approach for variational inference
  - Strong assumptions are used
- Beta VAE
  - Encourage disentanglement by simply adding one hyperparameter
  - Need more explanation
  - Information theoretic approach can be further considered
- GPP-VAE
  - Correlation among latent samples are considered with GP prior
  - Posterior Predictive distribution can be utilized
  - Efficiently learnable with several relaxations
  - Further improvement seems difficult

Thank you for your attention