NIPS2018読み会

岐阜大学 加藤研究室 M2 山田智輝

Papers

- Disentangled Representation Learning 系全般
  - VAE, β-VAE (Related Works)
  - β-TCVAE, Dual Swap Disentangling, JointVAE, etc.

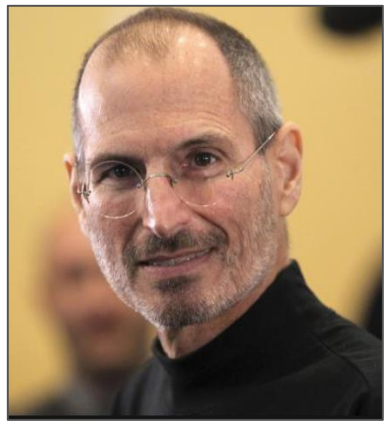
- 選定理由
  - 自分の研究に関わっている
  - Disentangle という単語が採択論文に一定数見られた(タイトルに12件)
  - Disentangle な設計を理論的に理解できる説明が欲しかった
  - 加藤研究室内での Gumbel max trick への関心がすごい

What is Disentangled Representation Learning ?

Disentangled Representation Learning

- Disentangle・・・もつれをほどく, 解き放つ, 解放される
- Disentangled RL・・・もつれのない表現の学習

- Example・・・顔を構成する要素, 日常風景の要素, etc.



- 髪色
- 眼鏡
- 笑顔
- 性別
- 目の色
- .etc

本来, これらの要素は
独立に制御できる

NNの特徴抽出でも
そうであってほしい

Why disentangled RL is important?

- 人間もそういう学習をしている(気がする)
  - 顔の向き, 目の大きさ, 肌の色等, 人間はある程度分けて考えている
- 独立した制御が出来たほうが解釈が容易になる
- 後のタスクに活かしやすくなる(かもしれない)
- 特徴として説得力がある(気がする)
  - 何を見ているか分からない特徴よりも, 解釈できる特徴の方が
    産業応用を目指す企業にとっては嬉しい(企業の方々が産業応用しやすくなる)

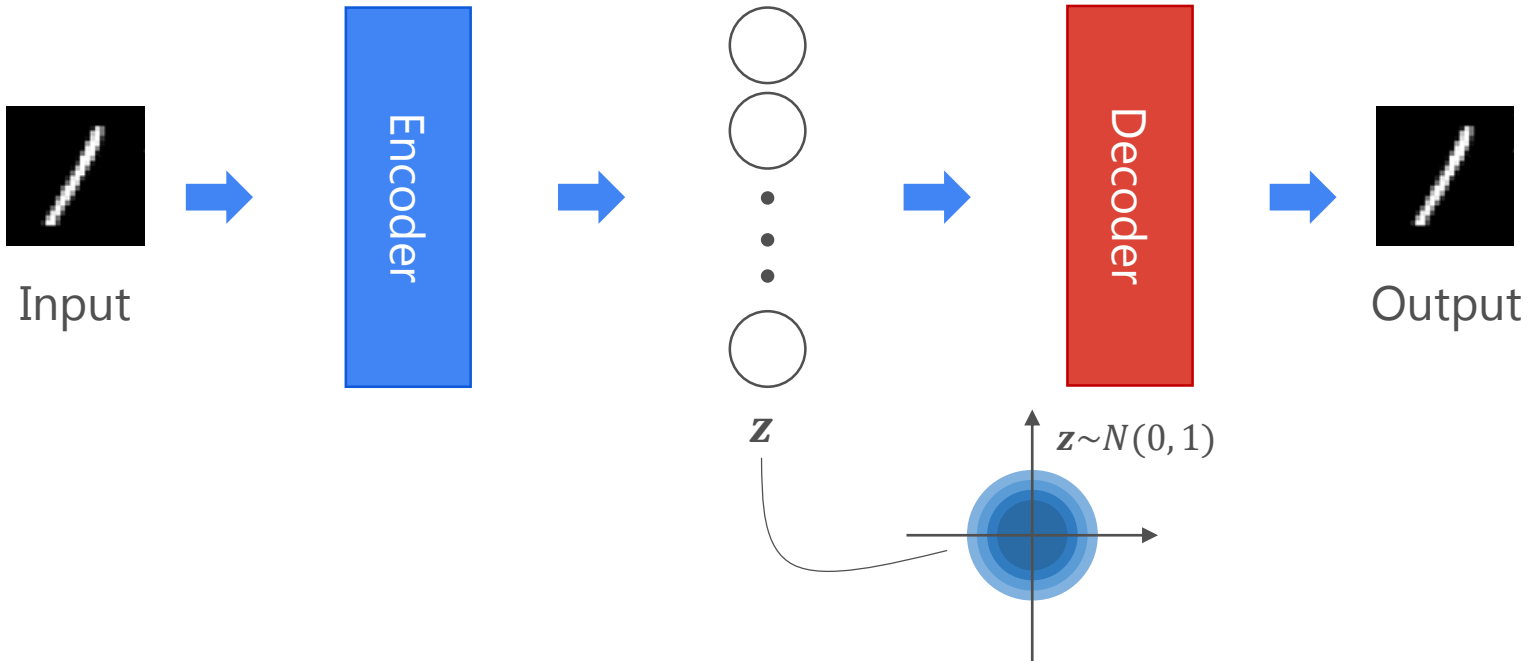
Difficulty

- 教師無しで学習できると望ましい
  - もしかしたら通常のDLでもdisentangleされる可能性はある(特に教師有り学習)
  - 応用を考えると, 潜在的な因子のラベル付けをするのはコストが大きい
  - そもそも、ヒトが潜在的な因子を把握できない場合がある
- 学習手法に根拠が欲しい
  - 「やってみたらdisentangleされてた」ではなく,
    「こういう構造(学習)だからdisentangleされる」という根拠が欲しい
  - 因子が独立していることの定義, 評価は難しい

Related Works

Variational Autoencoder (VAE)

- 入力画像を復元するようなNNを学習するモデル
- AEの構造に加え, zがガウス分布に従うという仮定をする
- Loss関数に, 事前分布, 事後分布(の近似)間のKL距離を組み込む
- 多様体を学習できることが報告されている (AEでも言えること)



Loss Function

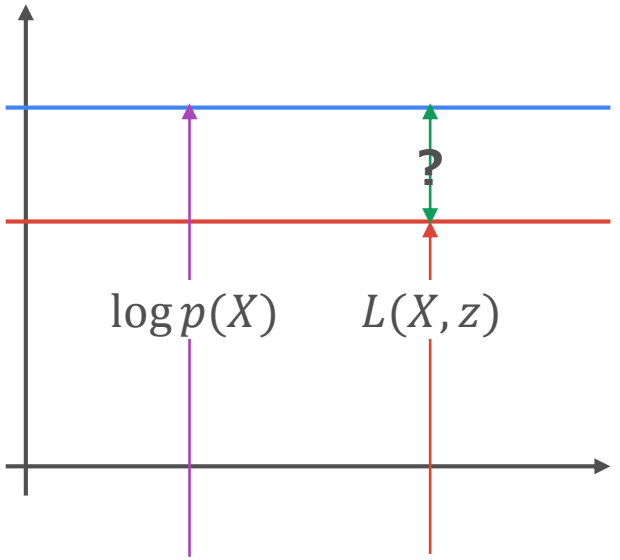
- $\log p(X)$ が最大になるパラメータを学習する。
- 直接最大化することができないので、下限 $L(X,z)$ を最大化する

$$\log p(X) = \log \int p(X,z)dz$$

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

$$\geq \int q(z|X) \log \frac{p(X,z)}{q(z|X)} dz$$

$$= L(X,z)$$



Loss Function

- $\log p(X)$ と $L(X, z)$ の差を計算する

$$\log p(X) - L(X, z)$$

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

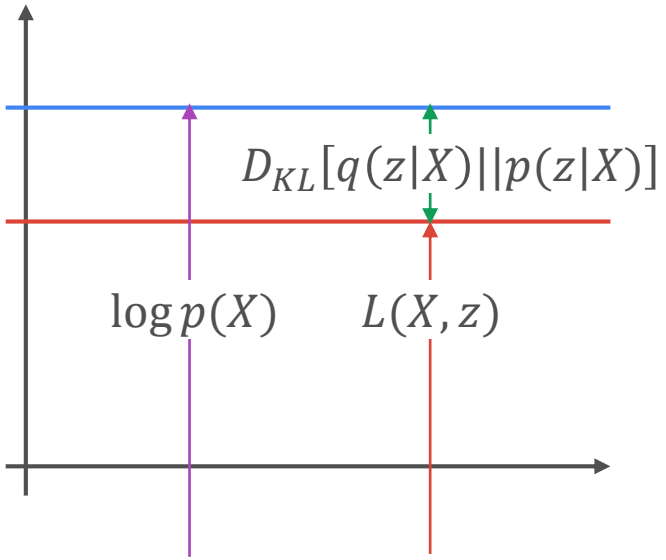
$$= \log p(X) \int q(z|X) dz - \int q(z|X) \log \frac{p(X, z)}{q(z|X)} dz$$

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

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

$$= D_{KL}[q(z|X)||p(z|X)] \geq 0$$

$$L(X, z) = \log p(X) - D_{KL}[q_\phi(z|X)||p_\theta(z|X)]$$



Loss Funtion

$$
\begin{aligned}
&D_{KL}[q_\phi(z|X)||p_\theta(z|X)] \\
&= \mathbb{E}_{q_\phi(z|X)}\left[\log q_\phi(z|X) - \log p_\theta(z|X)\right] \\
&= \mathbb{E}_{q_\phi(z|X)}\left[\log q_\phi(z|X) - \log p_\theta(X|z) - \log p(z) + \log p(X)\right] \\
&= \mathbb{E}_{q_\phi(z|X)}\left[\log q_\phi(z|X) - \log p_\theta(X|z) - \log p(z)\right] + \log p(X) \\
&= \mathbb{E}_{q_\phi(z|X)}\left[\log q_\phi(z|X) - \log p(z)\right] - \mathbb{E}_{q_\phi(z|X)}[\log p_\theta(X|z)] + \log p(X) \\
&= D_{KL}[q_\phi(z|X)||p(z)] - \mathbb{E}_{q_\phi(z|X)}[\log p_\theta(X|z)] + \log p(X)
\end{aligned}
$$

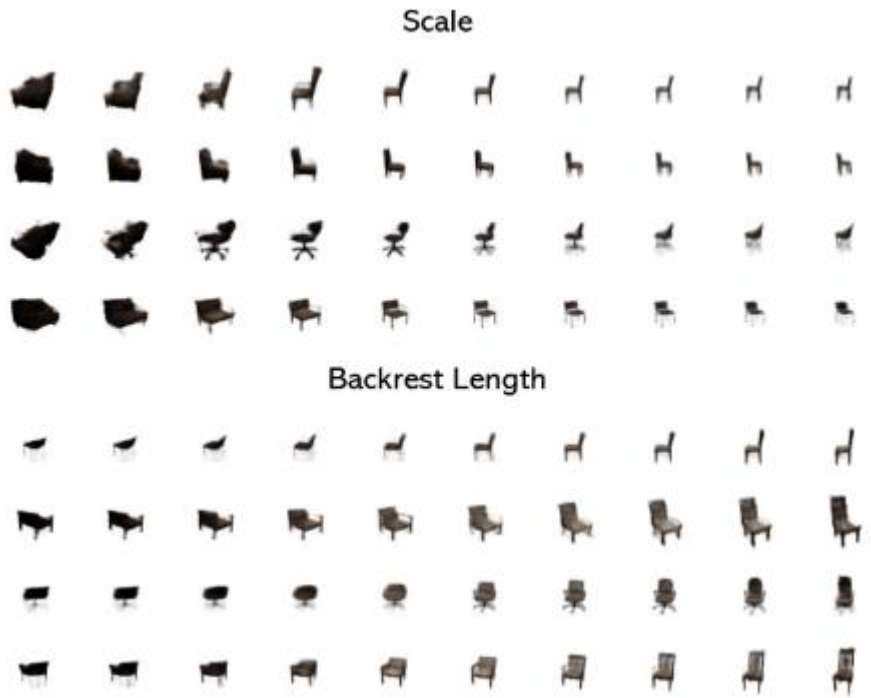
$$
\boxed{L(X,z) = \mathbb{E}_{q(z|X)}[\log p(X|z)] - D_{KL}[q(z|X)||p(z)]}
$$

この式を確率的勾配降下法を用いて最大化する

β-VAE

- 因子分解可能な特徴の学習を制約としたNN
- 下記式を最大化することにより学習が行われる

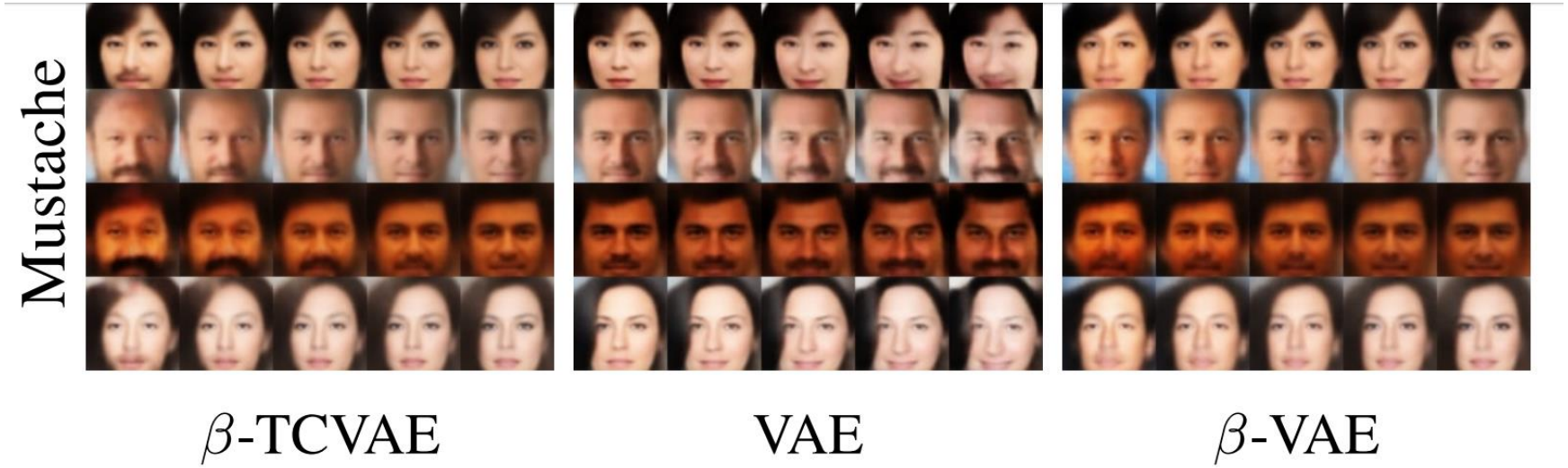
$$L(X,z) = \mathbb{E}_{q(z|X)}[\log p(X|z)] - \beta(D_{KL}[q(z|X)||p(z)])$$



Main Papers

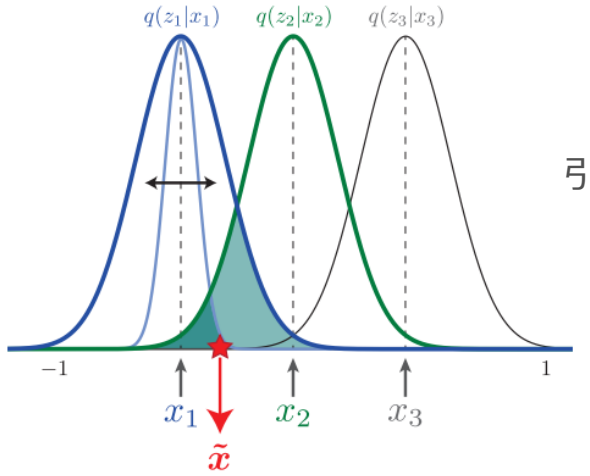
Isolating Sources of Disentanglement in Variational Autoencoders

- beta-VAEベースのDRLに関する手法
- beta-VAEだとbetaが大きくなるにつれて再構成誤差が犠牲になる
  (KLを分解して証明)
- KL項のうちzの総相関の比重を高めて学習することにより
  “betaのトレードオフ問題”を解決してDRLにおける優位性を証明
- 総相関導出のためのq(z)は直接得られないからバッチデータから
  重点サンプリングで導出 (まだ理解できてない)



betaのトレードオフ問題 (感覚的な解釈)

- 推論分布の分散が大きいほどKL距離は小さくなりがち
- 分散が大きいほど別データから推論される分布との重なりが大きくなる
- 重なりが大きいとzがどっち由来のものか分からなくなり再構成で混乱が生じる
- 標準正規分布に近づけるとKL距離は小さくなるが、再構成が犠牲になる (逆もまた然り)



引用: "Understanding disentangling in β-VAE"
https://arxiv.org/pdf/1804.03599.pdf

Figure 1: **Connecting posterior overlap with minimizing the KL divergence and reconstruction error.** Broadening the posterior distributions and/or bringing their means closer together will tend to reduce the KL divergence with the prior, which both increase the overlap between them. But, a datapoint $\tilde{x}$ sampled from the distribution $q(z_2|x_2)$ is more likely to be confused with a sample from $q(z_1|x_1)$ as the overlap between them increases. Hence, ensuring neighbouring points in data space are also represented close together in latent space will tend to reduce the log likelihood cost of this confusion.

betaのトレードオフ問題 (数式的な解釈)

$$
\begin{aligned}
&\frac{1}{N}\sum_{n=1}^{N} \mathrm{D_{KL}}\left(q(z|x_n)||p(z)\right) = \mathbb{E}_{p(n)}\left[\mathrm{D_{KL}}\left(q(z|n)||p(z)\right)\right] \\
&= \mathbb{E}_{p(n)}\left[\mathbb{E}_{q(z|n)}\left[\log q(z|n) - \log p(z) + \log q(z) - \log q(z) + \log \prod_j q(z_j) - \log \prod_j q(z_j)\right]\right] \\
&= \mathbb{E}_{q(z,n)}\left[\log \frac{q(z|n)}{q(z)}\right] + \mathbb{E}_{q(z)}\left[\log \frac{q(z)}{\prod_j q(z_j)}\right] + \mathbb{E}_{q(z)}\left[\sum_j \log \frac{q(z_j)}{p(z_j)}\right] \\
&= \mathbb{E}_{q(z,n)}\left[\log \frac{q(z|n)p(n)}{q(z)p(n)}\right] + \mathbb{E}_{q(z)}\left[\log \frac{q(z)}{\prod_j q(z_j)}\right] + \sum_j \mathbb{E}_{q(z)}\left[\log \frac{q(z_j)}{p(z_j)}\right] \\
&= \mathbb{E}_{q(z,n)}\left[\log \frac{q(z|n)p(n)}{q(z)p(n)}\right] + \mathbb{E}_{q(z)}\left[\log \frac{q(z)}{\prod_j q(z_j)}\right] + \sum_j \mathbb{E}_{q(z_j)q(z_{\setminus j}|z_j)}\left[\log \frac{q(z_j)}{p(z_j)}\right] \\
&= \mathbb{E}_{q(z,n)}\left[\log \frac{q(z|n)p(n)}{q(z)p(n)}\right] + \mathbb{E}_{q(z)}\left[\log \frac{q(z)}{\prod_j q(z_j)}\right] + \sum_j \mathbb{E}_{q(z_j)}\left[\log \frac{q(z_j)}{p(z_j)}\right] \\
&= \underbrace{\mathrm{D_{KL}}\left(q(z,n)||q(z)p(n)\right)}_{\text{(i) Index-Code MI}} + \underbrace{\mathrm{D_{KL}}\left(q(z)||\prod_j q(z_j)\right)}_{\text{(ii) Total Correlation}} + \underbrace{\sum_j \mathrm{D_{KL}}\left(q(z_j)||p(z_j)\right)}_{\text{(iii) Dimension-wise KL}}
\end{aligned}
$$

相互情報量

総相関

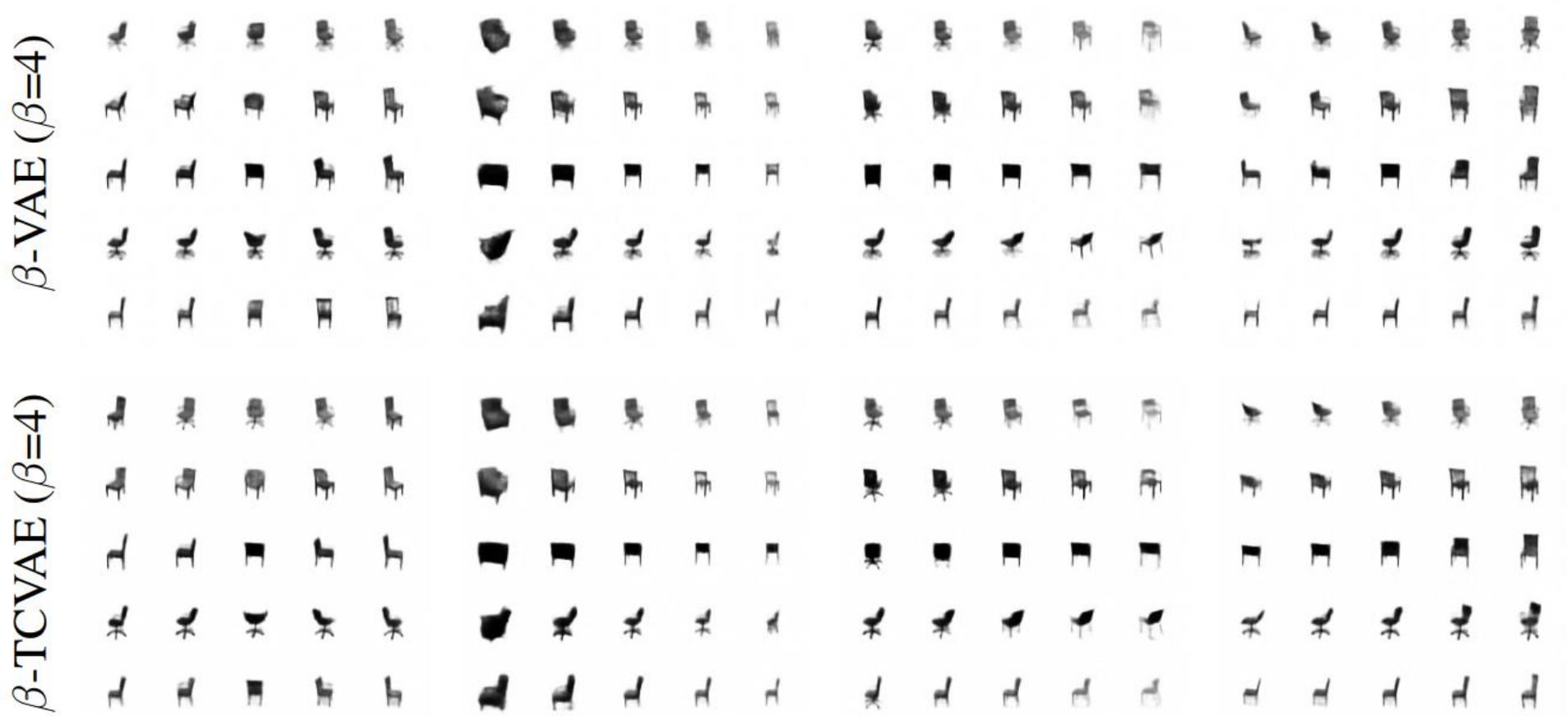
Loss Function

- 分解された項ごとに重みのバランスを調整してあげる
- おそらく、総相関がDRLに最も関わっているから β を調整してあげる
  (α, γ は1で固定)

- どうやって総相関を計算するの？(q(z)をどうやって求める？)
  $q(z|x)$を周辺化すれば良いが、ミニバッチ情報だけだと偏る
  →重点サンプリングを用いて近似する

$$\mathcal{L}_{\beta-TC} := \frac{1}{N}\sum_{n=1}^{N}\left(\mathbb{E}_{q(z|n)}[\log p(n|z)]\right) - \alpha I_q(z;n)$$
$$- \beta\, \mathrm{D_{KL}}\left(q(z)||\prod_j q(z_j)\right)$$
$$- \gamma \sum_j \mathrm{D_{KL}}\left(q(z_j)||p(z_j)\right)$$

Results



(a) Azimuth (b) Chair size (c) Leg style (d) Backrest

Results

(a) Baldness (-6, 6)
禿げ

(b) Face width (0, 6)

(c) Gender (-6, 6)

ちゃんと、Factorの分離ができていてイイネ

Dual Swap Disentangling

- 画像をペアで入力し、ボトルネック特徴の一部を入れ替えても、元の画像に戻るように学習されるEncoder-Decoderネットワーク (CVPR2018に似た論文が・・・)
- ラベルがある場合は、同じラベルを持つペアを入力し、1回の swap でも生成が変わらないように、ラベルなしの場合は 2回 swap を行って、元に戻るように学習する
  →半教師有り学習への適用が可能に

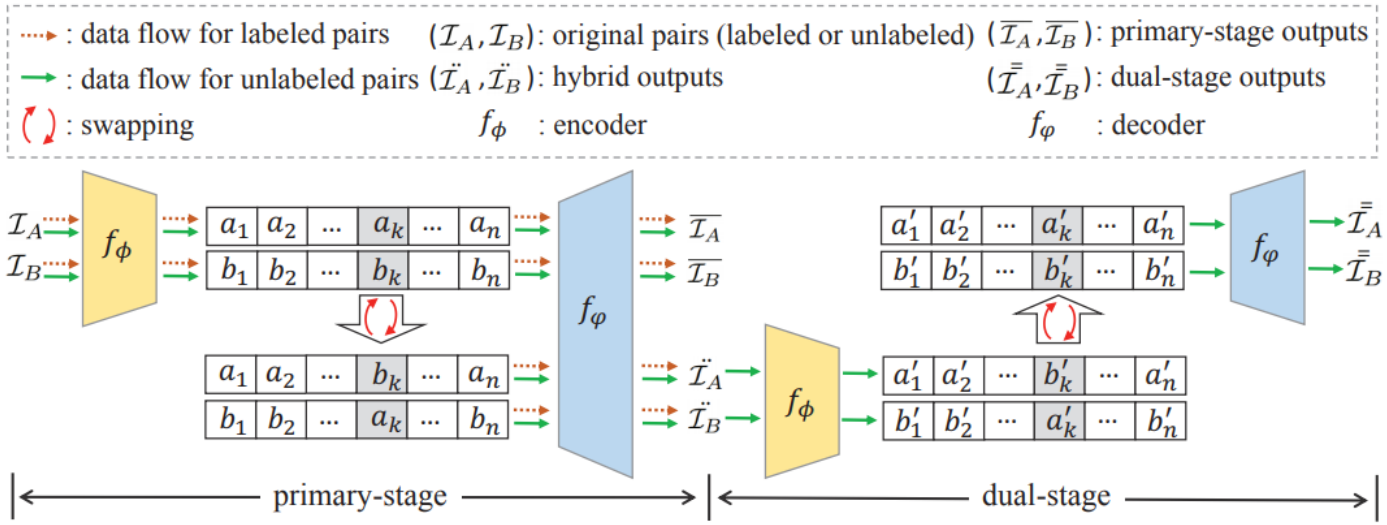


Figure 1: Architecture of the proposed DSD. It comprises two stages: primary-stage and dual-stage. The former one is employed for both labeled and unlabeled pairs while the latter is for unlabeled only.

Algorithm & Loss function

Algorithm 1 The Dual Swap Disentangling (DSD) algorithm

Input: Paired observation groups $\{\mathcal{G}_k, k = 1, 2, .., n\}$, unannotated observation set $\mathbb{G}$.

Initialize ϕ^1 and φ^1.
for t =1, 3, ..., T epochs **do**
  Random sample $k \in \{1, 2, ..., n\}$.
  Sample paired observation $(\mathcal{I}_A, \mathcal{I}_B)$ from group $\mathcal{G}_k$.
  Encode $\mathcal{I}_A$ and $\mathcal{I}_B$ into $\mathcal{R}_A$ and $\mathcal{R}_B$ with encoder f_{ϕ^t}.
  Swap the k-th part of $\mathcal{R}_A$ and $\mathcal{R}_B$ and get two hybrid representations $\ddot{\mathcal{R}}_A$ and $\ddot{\mathcal{R}}_B$.
  Construct $\mathcal{R}_A$ and $\mathcal{R}_B$ into $\bar{\mathcal{I}}_A = f_{\varphi^t}(\mathcal{R}_A)$ and $\bar{\mathcal{I}}_B = f_{\varphi^t}(\mathcal{R}_B)$.
  Construct $\ddot{\mathcal{R}}_A$ and $\ddot{\mathcal{R}}_B$ into $\ddot{\mathcal{I}}_A = f_{\varphi^t}(\ddot{\mathcal{R}}_A)$ and $\ddot{\mathcal{I}}_B = f_{\varphi^t}(\ddot{\mathcal{R}}_B)$.
  Update $\phi^{t+1}, \varphi^{t+1} \leftarrow \phi^t, \varphi^t$ by ascending the gradient estimate of $\mathcal{L}_{\mathbf{p}}(\mathcal{I}_A, \mathcal{I}_B; \phi^t, \varphi^t)$.
  Sample unpaired observation $(\mathcal{I}_A, \mathcal{I}_B)$ from unannotated observation set $\mathbb{G}$.
  Encode $\mathcal{I}_A$ and $\mathcal{I}_B$ into $\mathcal{R}_A$ and $\mathcal{R}_B$ with encoder $f_{\phi^{t+1}}$.
  swap the k-th part of $\mathcal{R}_A$ and $\mathcal{R}_B$ and get two hybrid representations $\ddot{\mathcal{R}}_A$ and $\ddot{\mathcal{R}}_B$.
  Construct $\mathcal{R}_A$ and $\mathcal{R}_B$ into $\bar{\mathcal{I}}_A = f_{\varphi^{t+1}}(\mathcal{R}_A)$ and $\bar{\mathcal{I}}_B = f_{\varphi^{t+1}}(\mathcal{R}_B)$.
  Construct $\ddot{\mathcal{R}}_A$ and $\ddot{\mathcal{R}}_B$ into $\ddot{\mathcal{I}}_A = f_{\varphi^{t+1}}(\ddot{\mathcal{R}}_A)$ and $\ddot{\mathcal{I}}_B = f_{\varphi^{t+1}}(\ddot{\mathcal{R}}_B)$.
  Encode $(\ddot{\mathcal{I}}_A, \ddot{\mathcal{I}}_B)$ into $\ddot{\mathcal{R}}'_A$ and $\ddot{\mathcal{R}}'_B$ with encoder $f_{\phi^{t+1}}$.
  Swap the k-th parts of $\ddot{\mathcal{R}}'_A$ and $\ddot{\mathcal{R}}'_B$ backward and get $\mathcal{R}'_A$ and $\mathcal{R}'_B$.
  Construct $\mathcal{R}'_A$ and $\mathcal{R}'_B$ into $\bar{\bar{\mathcal{I}}}_A = f_{\varphi^{t+1}}(\mathcal{R}'_A)$ and $\bar{\bar{\mathcal{I}}}_B = f_{\varphi^{t+1}}(\mathcal{R}'_B)$.
  Update $\phi^{t+2}, \varphi^{t+2} \leftarrow \phi^{t+1}, \varphi^{t+1}$ by ascending the gradient estimate of $\mathcal{L}_{\mathbf{u}}(\mathcal{I}_A, \mathcal{I}_B; \phi^{t+1}, \varphi^{t+1})$.
end for

Output: ϕ^T, φ^T

We take the total loss $\mathcal{L}_{\mathbf{p}}$ for the labeled pairs to be the sum of the original autoencoder loss $\mathcal{L}_{\mathbf{o}}$ and swap loss $\mathcal{L}_{\mathbf{s}}(\mathcal{I}_A, \mathcal{I}_B; \phi, \varphi)$:

$$\mathcal{L}_{\mathbf{p}}(\mathcal{I}_A, \mathcal{I}_B; \phi, \varphi) = \mathcal{L}_{\mathbf{o}}(\mathcal{I}_A, \mathcal{I}_B; \phi, \varphi) + \alpha \mathcal{L}_{\mathbf{s}}(\mathcal{I}_A, \mathcal{I}_B; \phi, \varphi), \tag{3}$$

Results

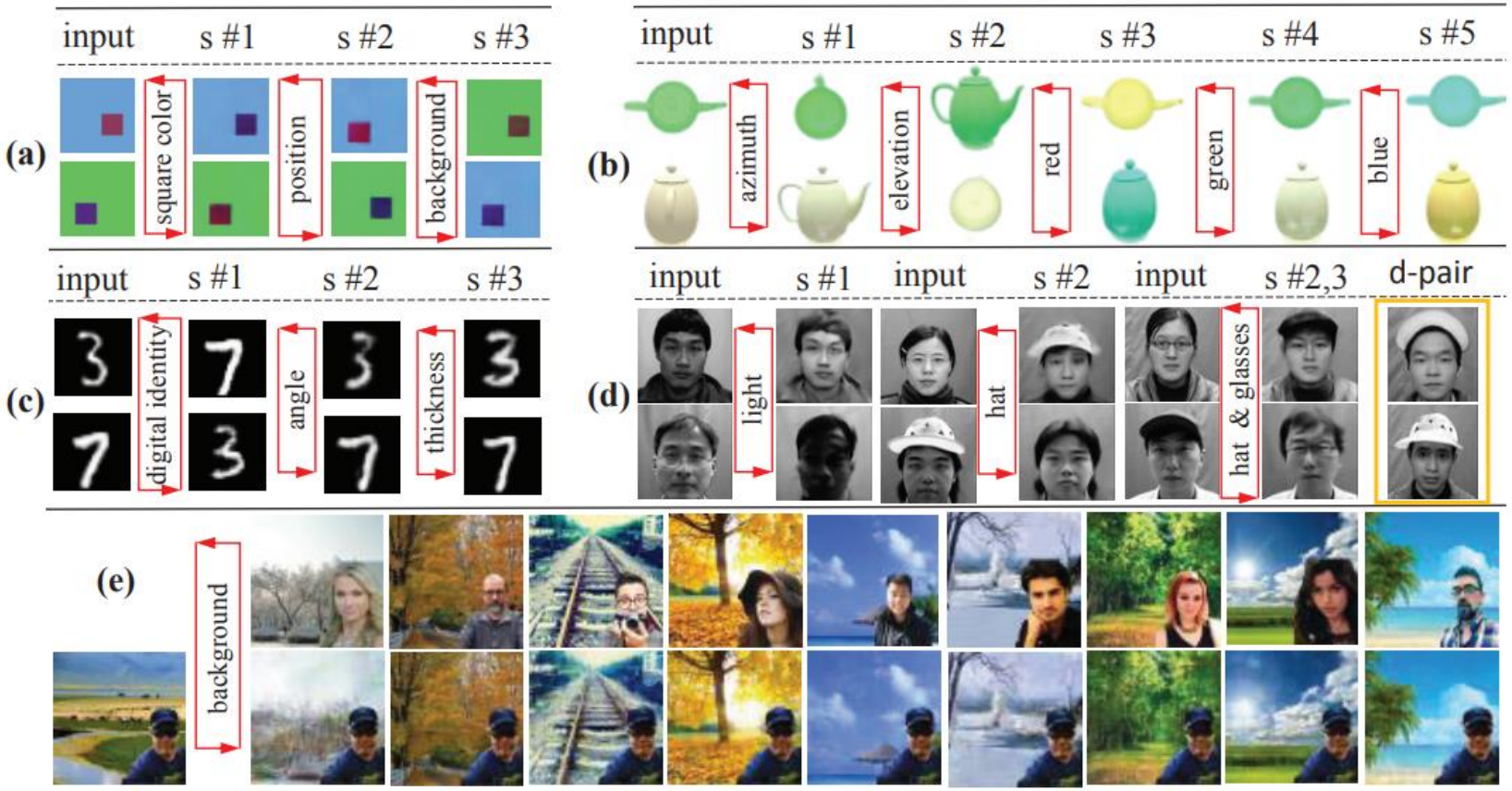


Figure 2: Visual results on 5 dataset. "d-pair" indicates disturbed pair.

Learning Disentangled Joint Continuous and Discrete Representations

- Beta-VAEベースのDRLに関する手法
- Beta-VAEだと連続特徴と離散特徴を同じ構造で表現している
- Reparameterization trickは連続分布に対するサンプリング手法
- Gumbel max trick (既存手法) を用いて離散分布も表現

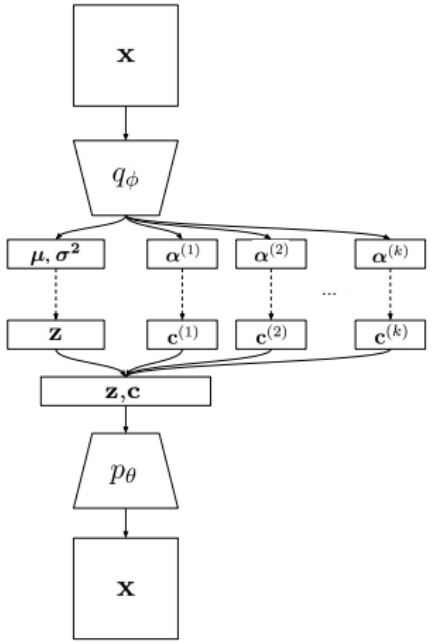


Figure 1: JointVAE architecture. The input $\mathbf{x}$ is encoded by q_ϕ into the parameters of the latent distributions. Samples are drawn from each of the latent distributions using the reparametrization trick (indicated by dashed arrows on the diagram). The samples are then concatenated and decoded through p_θ.

Loss function

- Loss関数はすごく単純
- 問題は式(7) 第 3 項をどうやって導出するか・・・
  - Gumbel max trickを使う

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{c}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})] - \beta D_{KL}(q_\phi(\mathbf{z}, \mathbf{c}|\mathbf{x}) \parallel p(\mathbf{z}, \mathbf{c})) \tag{4}$$

$$D_{KL}(q_\phi(\mathbf{z}, \mathbf{c}|\mathbf{x}) \parallel p(\mathbf{z}, \mathbf{c})) = D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) + D_{KL}(q_\phi(\mathbf{c}|\mathbf{x}) \parallel p(\mathbf{c})) \tag{5}$$

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{c}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) - \beta D_{KL}(q_\phi(\mathbf{c}|\mathbf{x}) \parallel p(\mathbf{c})) \tag{6}$$

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z},\mathbf{c}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})] - \gamma|D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) - C_z| - \gamma|D_{KL}(q_\phi(\mathbf{c}|\mathbf{x}) \parallel p(\mathbf{c})) - C_c| \tag{7}$$

連続分布のKL距離　　離散分布のKL距離

Gumbel max trick 概要

- 離散分布からサンプリングを行いつつ逆伝播もしたい
- カテゴリカル分布に適用されるReparameterization trickの手法

1. カテゴリカル分布のカテゴリ数だけGumbel分布からサンプリングした値を
   カテゴリカル分布の摂動として与えてargmaxを取るだけ
2. argmaxだと他のユニットが逆伝播されないのでsoftmaxにする
3. softmaxにすると確率変数がカテゴリカル分布に従わなくなるからConcrete分布を導入する

accept-reject algorithm

- カテゴリ数 m の カテゴリカル分布からサンプリングを行うアルゴリズム

カテゴリカル分布からサンプリングされる確率

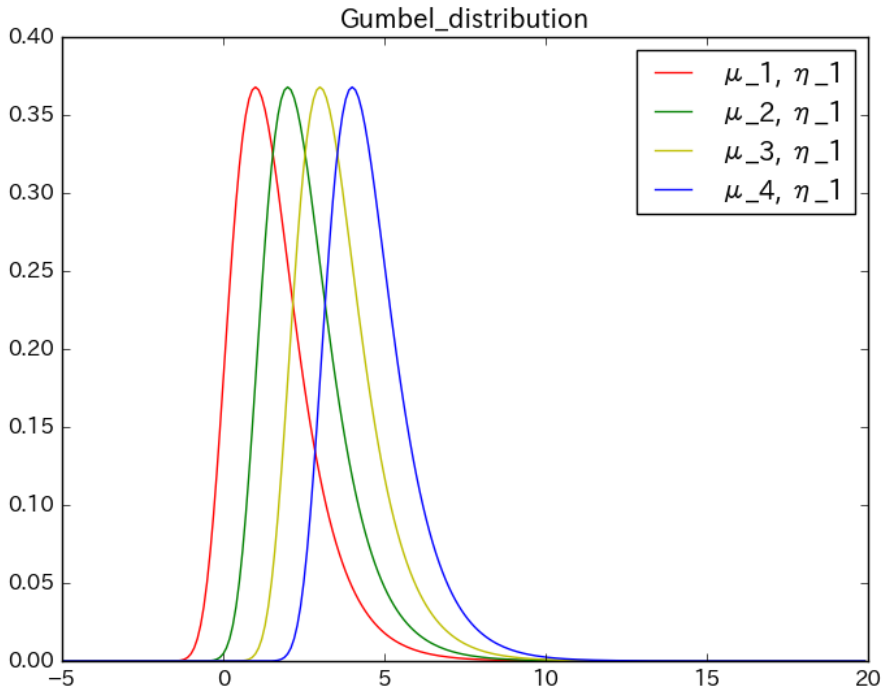
$$\alpha = \alpha_1, \alpha_2, \cdots \alpha_m$$

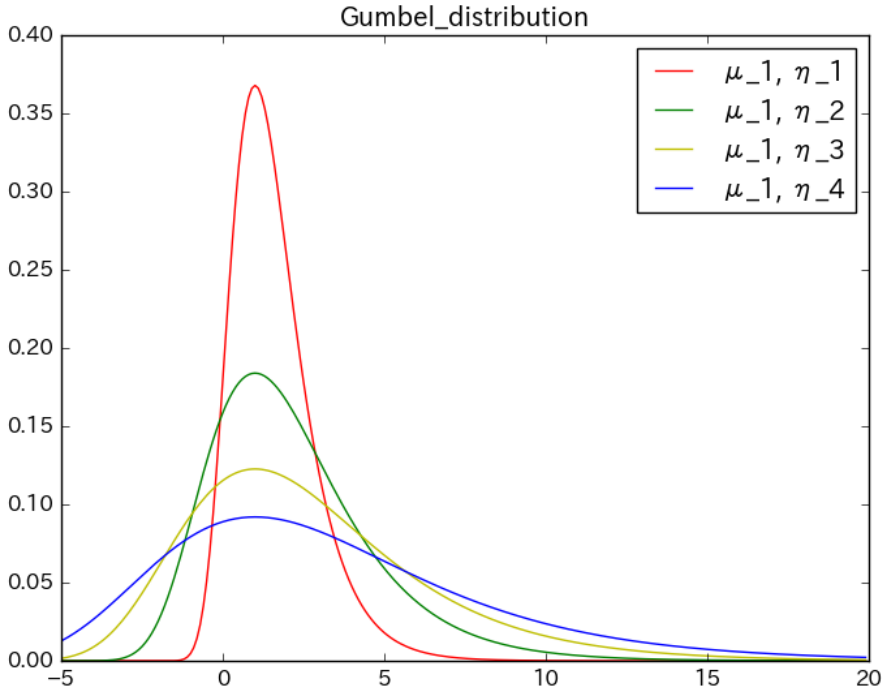
1. $u \sim \mathrm{Uniform}(0, \max(a_i))$ と $j \sim \{1, 2, \cdots, m\}$を一様にサンプリング
2. $a_j \geq u$の時はサンプリングを accept, そうじゃなければ reject

- 問題点
  - 全体的なサンプリング効率が悪い (少なくても 2 回)
  - accept されるまではずっとサンプリングを続けなければならない

Gumbel分布

- $Gu_{\mu,\eta}(x) = \frac{1}{\eta}\exp\left(-\frac{x-\mu}{\eta} - \exp\left(-\frac{x-\mu}{\eta}\right)\right)$ 確率密度関数
- $F_{\mu,\eta}(x) = \exp\left(-\exp\left(-\frac{x-\mu}{\eta}\right)\right)$ 累積分布関数





Gumbel max trick 手順

1. $g_i \sim \mathrm{Gumbel}$ Gumbel分布からサンプリング
2. $j = argmax(\log \alpha_i + g_i)$ サンプリング完了

- この手法だと, カテゴリ数 m 回のサンプリングだけでサンプリングができる
- どうしてこれだけでカテゴリカル分布からサンプリングできるのか？
  - j がサンプリングされる確率 ($\log \alpha_i + g_i$が最大値をとる確率) が
    カテゴリカル分布からサンプリングされる確率 α_i と一致すれば良い

証明

- $\mu = \log \alpha_i$, $\eta = 1$のGumbel分布からサンプリングされた値を$z_0, \cdots, z_m$とする
  (m はカテゴリ数)
- $p(z_i$が最大$|z_i, \alpha_{1,\cdots m}) = \alpha_i$を証明する

- μは正規分布でいう平均と同じ役割を果たしているので
  reparameterization trickと同様の変形ができる
  - $Gu_{\log \alpha_i,\, 1}(x) = \log \alpha_i + Gu_{0,\ 1}(x)$
  - よって, z_iは i.i.d にサンプリングされる

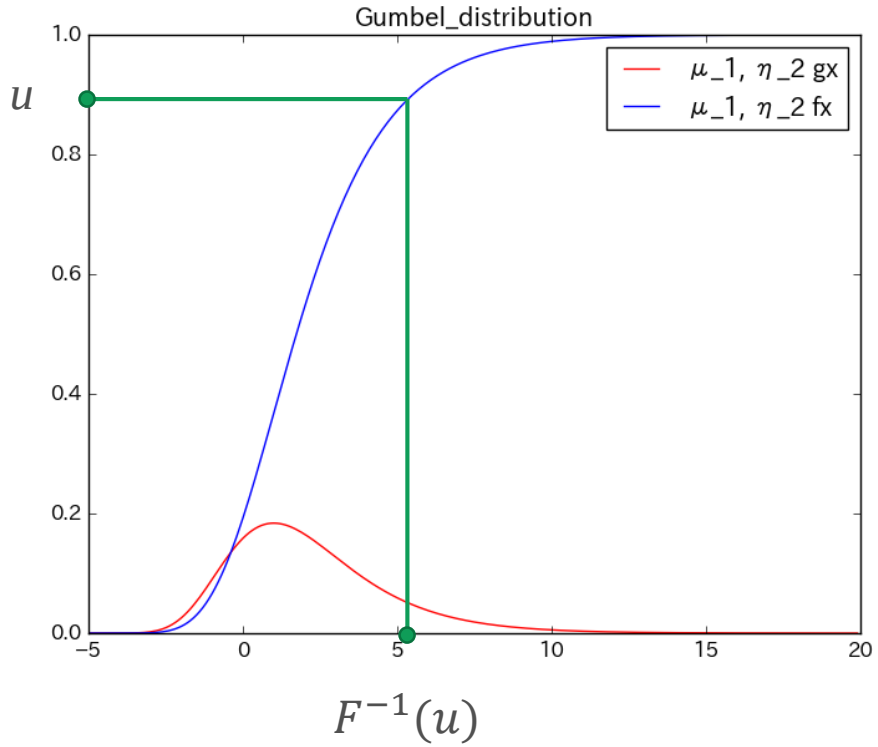
  - $p(z_i$が最大$|z_i, \alpha_{1,\cdots m}) = p(z_i > z_1, \ldots, z_i > z_m|\alpha_{1,\cdots m}) = p(z_i > z_1|\alpha_1) \ldots p(z_i > z_m|\alpha_m)$
    (ちょっと理解が怪しい...)

証明

- 前スライドの式は累積分布関数の掛け合わせになる
  - $p(z_i\text{が最大}|z_i, \alpha_{1,\cdots m}) = \prod_{j\neq i} \exp\left(-\exp\left(-(z_i - \log \alpha_j)\right)\right)$
- 上式を z_i について周辺化する(詳細は~~面倒なので~~省略) あとは計算するだけ
  - ざっくり言うと, 下式に代入して置換積分をするだけ
  - $p(z_i\text{が最大}|\alpha_{1,\cdots m}) = \int p(z_i|\alpha_i) p(z_i\text{が最大}|z_i, \alpha_{1,\cdots m}) dz_i$
- 最終的に下式が導出され, 証明された
  - $p(z_i\text{が最大}|\alpha_{1,\cdots m}) = \frac{\exp(\log \alpha_i)}{\sum_j \exp(\log \alpha_j)} = \frac{\alpha_i}{\sum_j \alpha_j} = \alpha_i$
- 詳細はこちら http://peluigi.hatenablog.com/entry/2018/06/21/142753

Gumbel分布からサンプリング

- 逆関数法を用いると容易にサンプリングできる
  - 一様分布からサンプリングされた u を累積分布関数の逆関数に代入する
    (赤 : Gumbel分布の確率密度関数　青 : Gumbel分布の累積分布関数)
- Gumbel分布の累積分布関数の逆関数は $-\log(-\log(u))$



ここまでの Gumbel max trick まとめ

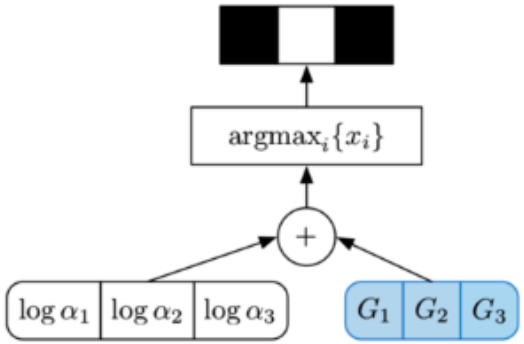
- カテゴリカル分布から効率的にサンプリングする手法を確認
  - 証明の理解が大変だった

- Gumbel分布からのサンプリングは逆関数法で容易にできる

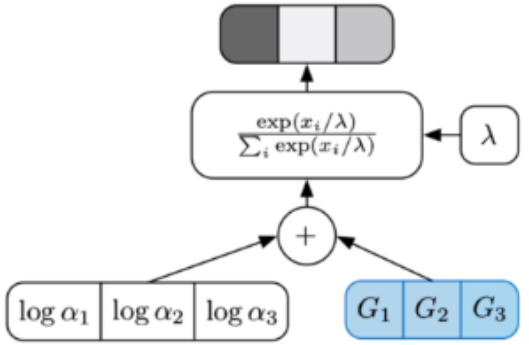
Gumbel max trickの問題点

- サンプリングされたユニットのみが 1 になり, 他のユニットが 0 になる
  - 大部分のパラメータが更新されなくなる
- 解決策
  - argmaxじゃなくてsoftmaxをとれば 良さそう
    (Concrete変数 or Gumbel softmax分布と呼ばれる)
  - でも、勝手に softmax にしちゃったからカテゴリカル分布のサンプリングじゃなくなった
  - カテゴリカル分布は諦めて
    CONtinuous relaxation of disCRETE (Concrete) 分布に従うことにしよう

$$X_k = \frac{\exp\big((\log \alpha_k + G_k)/\lambda\big)}{\sum_i^n \exp\big((\log \alpha_i + G_i)/\lambda\big)}$$



(a) Discrete(α)



(b) Concrete(α, λ)

Concrete分布

- Concrete分布の確率密度関数は下式で表される

$$p_{\alpha,\lambda}(x) = (n-1)!\,\lambda^{n-1}\prod_{k=1}^{n}\left(\frac{\alpha_k x_k^{-\lambda-1}}{\sum_{i=1}^{n}\alpha_i x_i^{-\lambda}}\right)$$

- Concrete分布の確率密度関数は下の命題を満たす

1. G_k~Gumbel が i.i.d ならば $X_k = \frac{\exp\left((\log\alpha_k + G_k)/\lambda\right)}{\sum_i^n \exp\left((\log\alpha_i + G_i)/\lambda\right)}$ (これは当然)
2. $P(X_k > X_i\ for\ i \neq k) = \alpha_k / \sum_{i=1}^{n}\alpha_i$ (Concrete分布を変形したら離散確率変数になる)
3. $P(\lim_{\lambda\to 0} X_k = 1) = \alpha_k / \sum_{i=1}^{n}\alpha_i$ (温度変数を0に近づけると離散確率変数になる)
4. $\lambda \leq (n-1)^{-1}$ ならば $p_{\alpha,\lambda}(x)$ は log-convex である

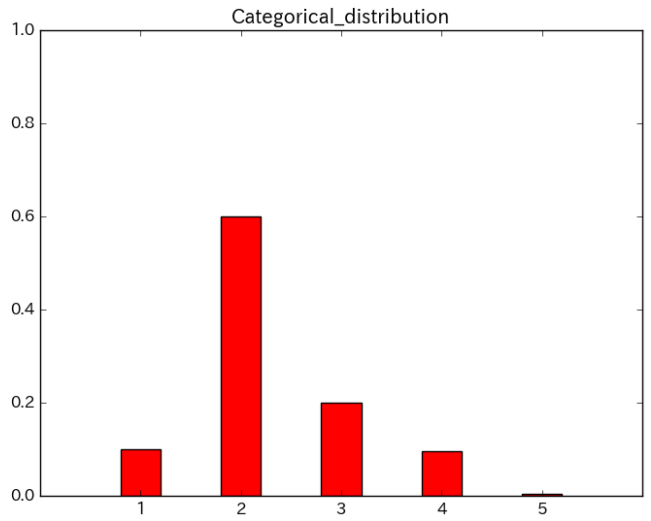
- 正直, 分かりませんでした… (恐らく、著者実装はこっちじゃない)
  - もし、実装するならG_kをConcrete分布からのサンプリングに置き換えればよい

Gumbel-softmax 分布

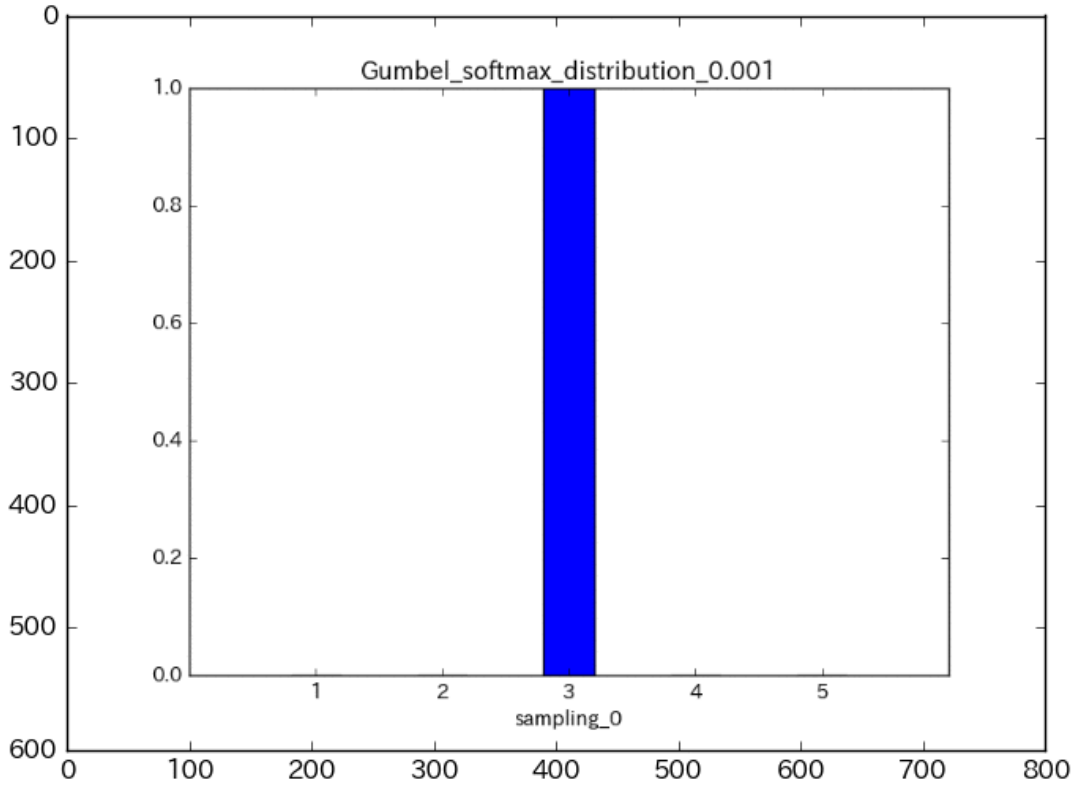
- 下式で表される分布 (λはハイパラ)
  - Gumbel max trickによるサンプリングを緩和したもの
  - λ が大きいほど一様分布に近づき, 0に近いほど argmaxを取るようになる

$$X_k = \frac{\exp\left((\log \alpha_k + G_k)/\lambda\right)}{\sum_i^n \exp\left((\log \alpha_i + G_i)/\lambda\right)} \ , \sum_{i=1}^n \alpha_k = 1$$

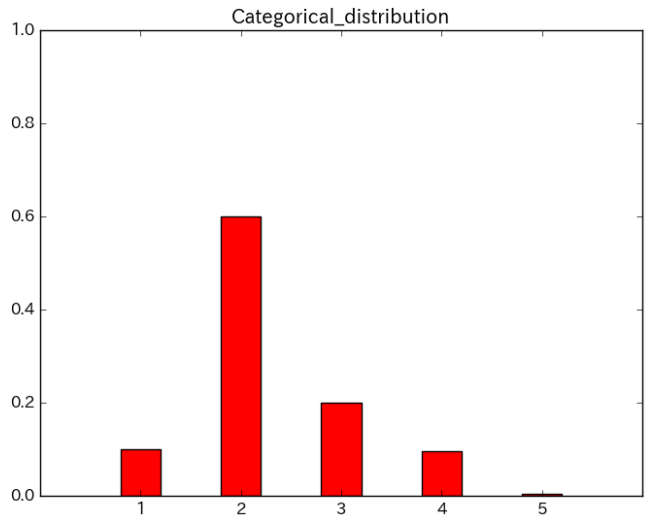
Gumbel-softmax Sampling



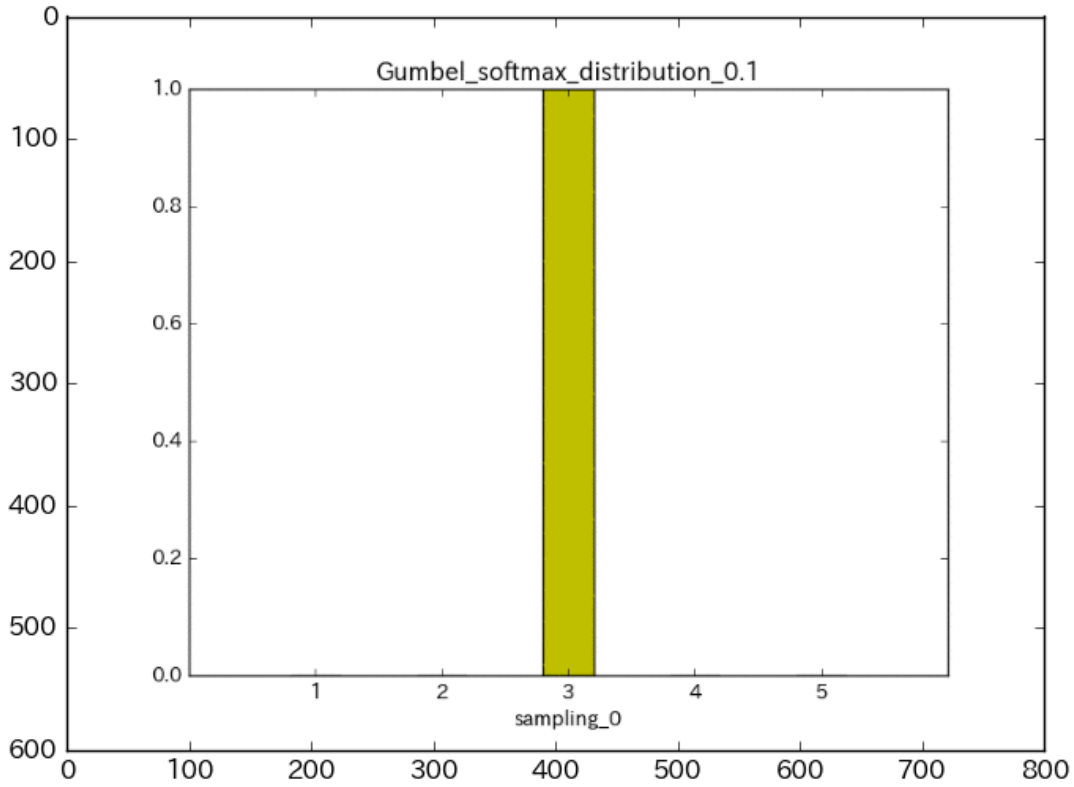
[0.100, 0.600, 0.200, 0.095, 0.005]



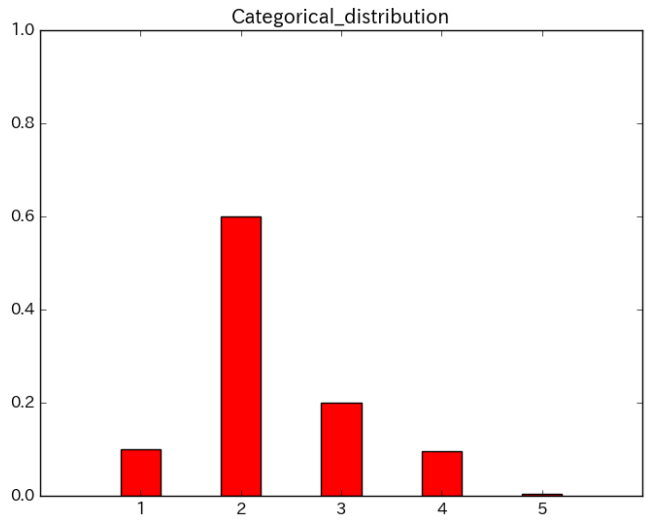
Gumbel-softmax Sampling



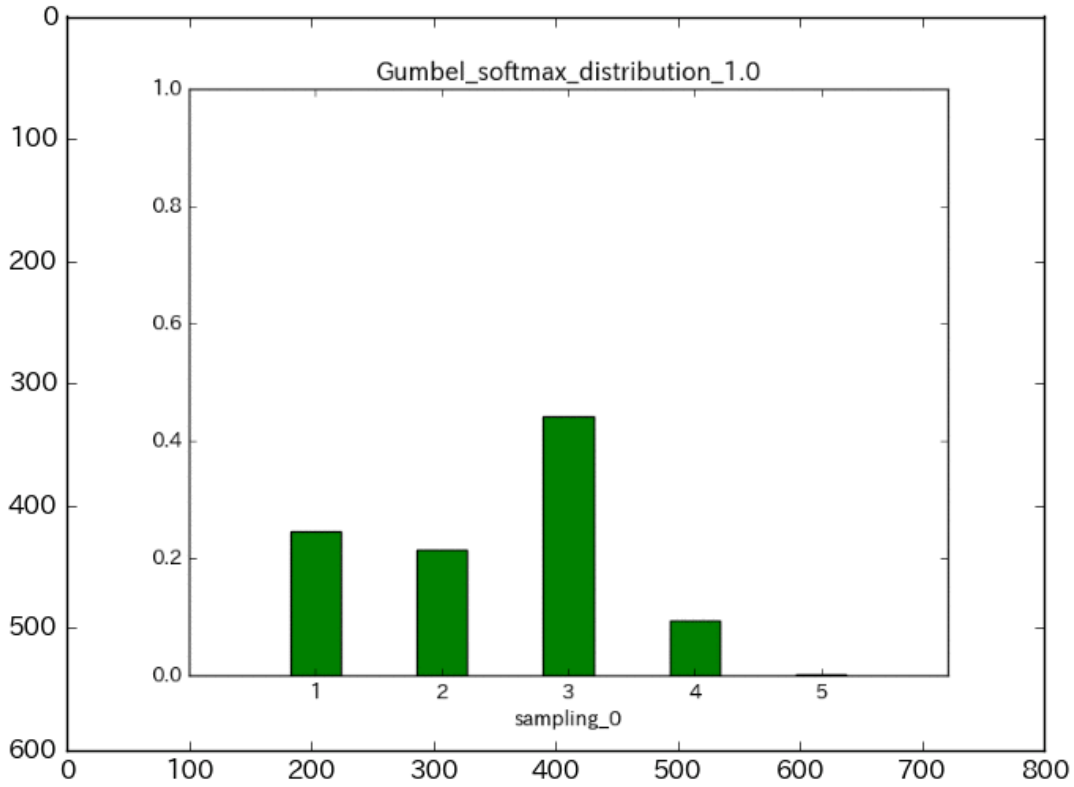
[0.100, 0.600, 0.200, 0.095, 0.005]



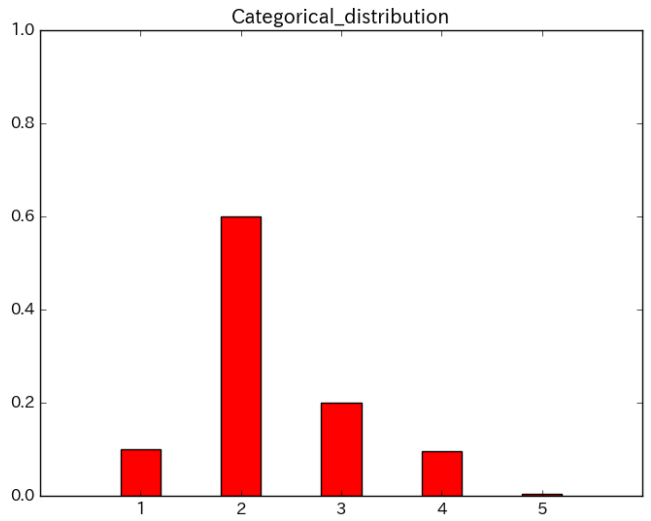
Gumbel-softmax Sampling



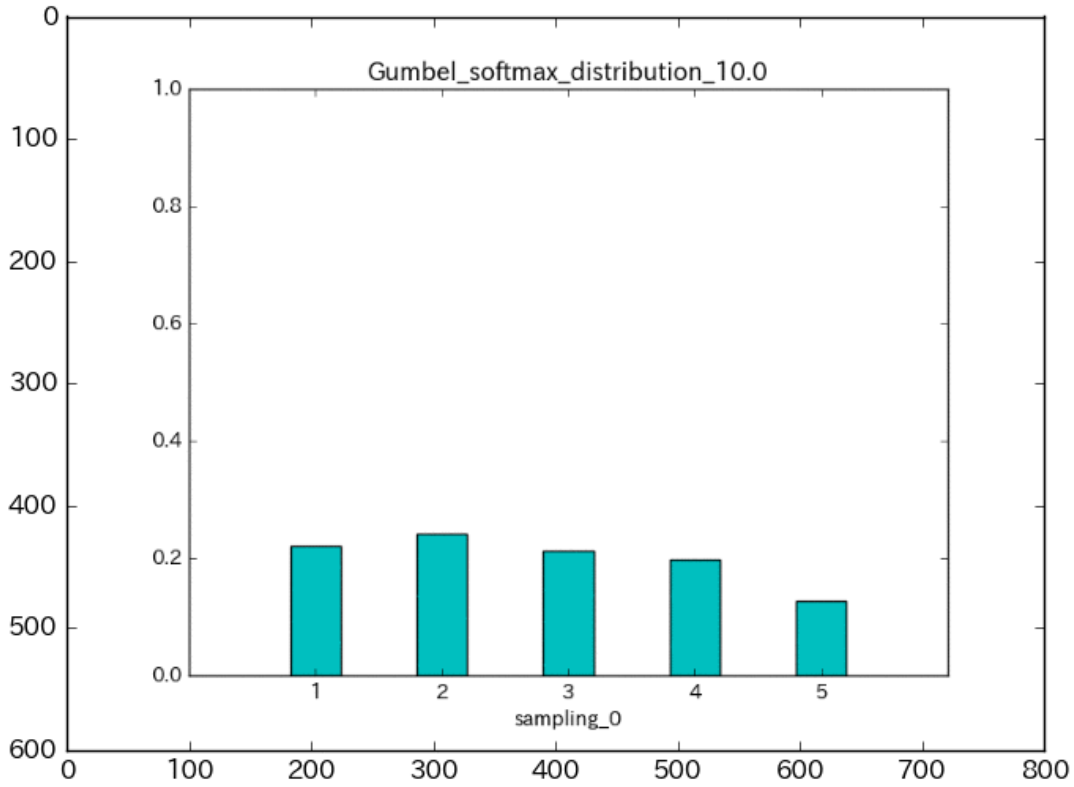
[0.100, 0.600, 0.200, 0.095, 0.005]



Gumbel-softmax Sampling



[0.100, 0.600, 0.200, 0.095, 0.005]



Results

(a) Angle (continuous)

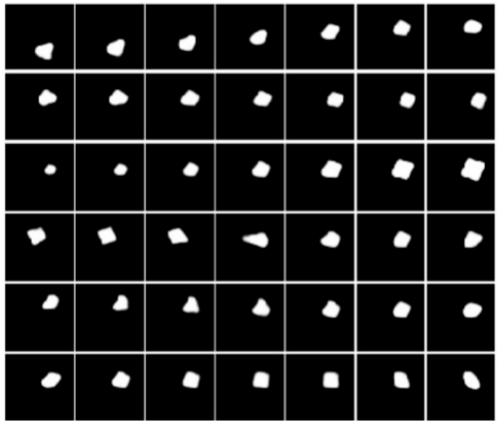
(b) Thickness (continuous)

(c) Digit type (discrete)

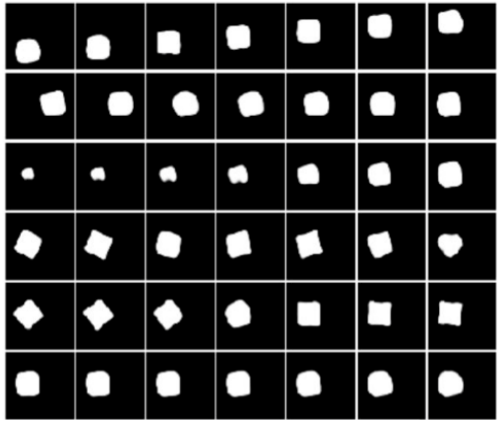
(d) Width (continuous)

Model	Score
β-**VAE**	0.73
FactorVAE	0.82
JointVAE	0.69

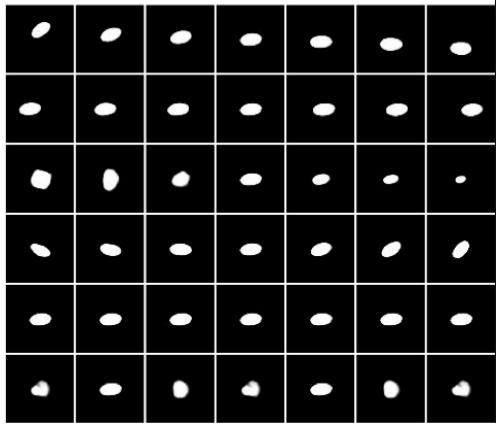
β-VAE



FactorVAE



JointVAE



読んだ感想 (まとめ)

- NIPSの中で (最低限流れの理解ができた) 数件のDRL論文を紹介した
  - 全体的に、domain transfer, life-long learning 等の大目標の前座とした基礎実験が主
    (まだ、面白そうなことができているだけで, 自然画像等の実利用応用は遠そう・・・)
  - イメージとして, 構造メインの貢献, 数式メインの貢献が半々
    (どっちベースでアプローチしても沼)
  - モデル構造は圧倒的に Encoder-Decoder (β-VAE) ベースが多かった印象
  - 評価法はMNIST, CelebA, d-sprites 等の基礎データセットを用いた視覚的評価は殆どの論文で共通して行われ, 定量評価は論文の貢献に合わせて様々な手法を用いている
    (定量評価法のコレジャナイ感は否めない)
- (個人的に) VAE の KL項 の分解, Gumbel max trick は良い知見だった
- (個人的に) Swappingも面白いとは思ったが DRL において
  ラベル付けはしたくない。Domain Transfer ベースは実装できる気がしない

参考論文・資料

- [DL輪読会]Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies
  https://www.slideshare.net/DeepLearningJP2016/dllifelong-disentangled-representation-learning-with-crossdomain-latent-homologies-115315318
- THE CONCRETE DISTRIBUTION: A CONTINUOUS RELAXATION OF DISCRETE RANDOM VARIABLES
  https://arxiv.org/pdf/1611.00712.pdf
- Gumbel Max Trickについて勉強した
  http://peluigi.hatenablog.com/entry/2018/06/21/142753
- Concrete Distributionについて論文を読んだ
  http://peluigi.hatenablog.com/entry/2018/06/23/192435
- Categorical Reparameterization with Gumbel-Softmax [arXiv:1611.01144]
  http://musyoku.github.io/2016/11/12/Categorical-Reparameterization-with-Gumbel-Softmax/
- Isolating Sources of Disentanglement in Variational Autoencoders
  https://arxiv.org/abs/1802.04942
- Dual Swap Disentangling
  https://arxiv.org/abs/1805.10583
- Learning Disentangled Joint Continuous and Discrete Representations
  https://arxiv.org/abs/1804.00104