

# Auto-Encoding Variational Bayes

Diederik P. Kingma

Max Welling

2014-03-01

## 概要

有向確率モデルにおいて、潜在変数が連続でその事後分布が計算不可能であり、かつデータセットが大規模なとき、その学習及び推論はどのように行ったらよいだろうか。我々は、この学習と推論について、データセットが大きくなっても適用可能で、また計算不可能な場合でも微分可能性についてある緩い条件が成立すれば機能する、確率的分変アルゴリズムを導入する。我々は次の2つのことを達成した。1つは、変分下限を再パラメータ化することで、下限の推定を可能にしたことである。なお、これは通常の確率的勾配法を用いて直接的に最適化することができる、もう1つは、各データについて連続潜在変数を持つ i.i.d. なデータセットについて、効率的な事後推定を可能にしたことである。これは、下限の推定器を用い、近似的推論モデル（認識モデルとも呼ばれる）を計算不可能な事後分布に適用することで達成される。この理論の利点は実験の結果に表れている。

## 1 序論

有向確率モデルにおいて、その連続潜在変数またはパラメータが計算不可能な事後分布を持つとき、近似的推論と学習はどのように行ったらよいだろうか。変分ベイズ（variational Bayesian, VB）的アプローチでは、計算不可能な事後分布に対する近似の最適化が行われる。通常の平均場的なアプローチでは、事後分布の近似についての期待値の解析解が必要となるが、一般にはこれもイントラクタブルである。我々は、変分下限の再パラメータ化によって、いかにして単純で微分可能な下限の不偏推定器を得られるのかを示す。この SGVB (Stochastic Gradient Variational Bayes) 推定器は、連続潜在変数またはパラメータを持つほとんどのモデルにおいて効率の良い近似的事後推定に用いることができ、また、通常の確率上昇法を用いて最適化できる。

各データに対し連続な潜在変数があり、かつ i.i.d. なデータセットの場合について、我々は自己符号化変分ベイズ（AutoEncoding Variational Bayes, AEVB）アルゴリズムを提案する。AEVB アルゴリズムによって、学習・推論を非常に効率よく行うことができるようになる。これは、SGVB 推定器を用いて認識モデルを最適化することによって達成される。この認識モデルでは、各データに対して計算コストの高い反復的推論法（MCMC など）を用いず、モデルのパラメータを効率よく学習できるような、単純な伝承サンプリングを用いることによって、非常に効率的な近似的事後推定を行うことができる。学習された近似的事後推定モデルは、認識・ノイズ除去・表現学習・可視化などの、様々なタスクに用いることができる。認識モデルにニューラルネットワークを用いたとき、これを変分自己符号化器（variational auto-encoder）と呼ぶ。

## 2 手法

この節では、連続型潜在変数を持つ様々な有向グラフィカルモデルに対して、下限推定器（確率的目的関数）を得られようような手法を導入する。いま、各データについて潜在変数がある i.i.d. なデータセットという一般の場合を考える。最尤推定や最大事後確率推定を（大域的な）パラメータに対して行い、また変分推論を潜在変数に対して行う。これは例えば、変分推論を大域的なパラメータに適用する場合にも自然に拡張できる。このアルゴリズムは付録に示したが、実験は今後の展望とした。以下では簡単のため定常なデータセットを仮定するが、この手法はストリーミングデータのようなオンラインの非定常なデータセットにも適用できる。

### 2.1 問題の概要

$N$  個の i.i.d. な変数  $\mathbf{x}$  から成るデータセット  $\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^N$  を考える。なお、 $\mathbf{x}$  は連続的でも離散的でもよい。いま、そのデータは、観測されない連続型確率変数  $\mathbf{z}$  を含む、ある確率過程から生成されていると仮定する。その確率過程には次の 2 段階がある。

- (1)  $\mathbf{z}^{(i)}$  がある事前分布  $p_{\theta^*}(\mathbf{z})$  から生成される
- (2)  $\mathbf{x}^{(i)}$  がある条件付確率  $p_{\theta^*}(\mathbf{x}|\mathbf{z})$  から生成される

ここで、事前分布  $p_{\theta^*}(\mathbf{z})$  と尤度  $p_{\theta^*}(\mathbf{x}|\mathbf{z})$  は分布  $p_{\theta}(\mathbf{z})$  と  $p_{\theta}(\mathbf{x}|\mathbf{z})$  の parametric family から得られ、また、これらの確率密度関数は  $\theta$  と  $\mathbf{z}$  の両方についてほとんど至るところで微分可能であると仮定する。不運なことに、この過程の多くの部分が、我々から見えないところで起こるものである。つまり、真のパラメータ  $\theta^*$ 、及び潜在変数  $\mathbf{z}^{(i)}$  について、我々の知りえない。

非常に重要なことだが、我々は、一般的に行われるような周辺確率や事後確率に関する単純化を仮定しない。むしろここでは、以下のような場合でも機能するような、一般的なアルゴリズムが求められているのである。

#### 1. 計算不可能性

周辺尤度の積分  $p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{z})p_{\theta}(\mathbf{x}|\mathbf{z})d\mathbf{z}$  がイントラクタブルな場合（このとき周辺尤度の値を求めたり微分したりすることはできない）。このとき、真の事後確率密度  $p_{\theta}(\mathbf{z}|\mathbf{x}) = p_{\theta}(\mathbf{x}|\mathbf{z})p_{\theta}(\mathbf{z})/p_{\theta}(\mathbf{x})$  は計算不可能である（このとき EM アルゴリズムは使用できない）。また、どんな平均場変分ベイズアルゴリズムに対しても、その積分は計算不可能となる。こういった計算不可能性は一般に起こりうるもので、尤度関数  $p_{\theta}(\mathbf{x}|\mathbf{z})$  にある程度複雑なもの、例えば非線形な隠れ層を持つニューラルネットワークを用いた場合などに現れる性質である。

#### 2. 大規模データセット

データが非常に多く、バッチ最適化の計算コストが高くなりすぎる場合。このとき、ミニバッチまたはデータ 1 つを用いてパラメータを更新したい。モンテカルロ EM 法などのサンプリングによる方法は、各データに対し計算コストの高いサンプリングを反復的に行うため、一般にかなり遅くなってしまう。

我々は上記のような状況に関する以下の 3 つの問題について関心があり、その解決法を提案する。

1. パラメータ  $\theta$  に対して、どう効率よく MLE や MAP 推定の近似を行うか。パラメータ自身も、自然の作用などを分析するときなどは関心の対象となりうる。また、パラメータによって、非明示的な確率過程を模倣することが可能になり、実データに似た人工データを作り出すことができるようになる。

2. 観測値  $\mathbf{x}$  が与えられたとき、ある  $\theta$  に対して、潜在変数  $\mathbf{z}$  の事後分布の近似をどう効率よく行うか。これは、符号化や表現学習をするときに有用である。
3. 変数  $\mathbf{x}$  の周辺確率の近似をどう効率よく行うか。これを行うことにより、 $\mathbf{x}$  に関する事前分布が必要となるような問題に対応できるようになる。例えば、コンピュータービジョンにおける応用例として、画像のノイズ除去や修復、超解像といったものが挙げられる。

上記のような問題を解くために、認識モデル  $q_\phi(\mathbf{z}|\mathbf{x})$  を導入しよう。これは、計算不可能な真の事後分布  $p_\theta(\mathbf{z}|\mathbf{x})$  の近似である。平均場変分推論における事後分布の近似とは対照的に、このモデルは因子分解可能である必要はなく、またそのパラメータ  $\phi$  は閉形式の期待値から計算されるものでもない。その代わりとして、認識モデルのパラメータ  $\phi$  を生成モデルのパラメータ  $\theta$  とまとめて学習する方法を提案する。

符号理論的な観点からすると、観測されない変数  $\mathbf{z}$  は潜在表現、または「符号」として解釈することができる。そこで本論文では、認識モデル  $q_\phi(\mathbf{z}|\mathbf{x})$  を、確率的「符号器」とも呼ぶことにする。これは、このモデルが、あるデータ  $\mathbf{x}$  が与えられたとき、その  $\mathbf{x}$  を生成する元となった符号  $\mathbf{z}$  の取りうる値の分布（例えばガウス分布など）を与えるからである。同様に、 $p_\theta(\mathbf{x}|\mathbf{z})$  を確率的「復号器」と呼ぶこととする。これは、このモデルが、ある符号  $\mathbf{z}$  が与えられたとき、対応する  $\mathbf{x}$  の値の取りうる範囲の分布を与えるからである。

## 2.2 変分下限

周辺尤度は、各データの周辺尤度の積で表される。つまり、

$$\log p_\theta(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}) = \sum_{i=1}^N \log p_\theta(\mathbf{x}^{(i)})$$

である。また、次が成立する。

$$\log p_\theta(\mathbf{x}^{(i)}) = D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) \quad (1)$$

右辺第一項は、真の事後分布から測った近似的事後分布の KL 情報量である。この KL 情報量は非負であることから、右辺第二項の  $\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$  は、データ  $i$  の周辺尤度に関する（変分）下限と呼ばれる。これは次のよ

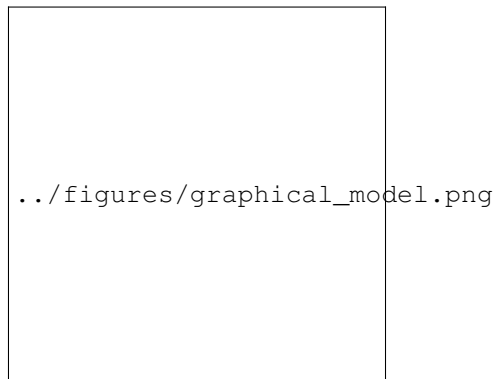


図 1: 対象としている有向グラフモデル。実線は生成モデル  $p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$  を表し、破線は計算不可能な事後分布  $p_\theta(\mathbf{z}|\mathbf{x})$  の変分近似  $q_\phi(\mathbf{z}|\mathbf{x})$  を表す。変分パラメータ  $\phi$  は生成モデルのパラメータ  $\theta$  とまとめて学習される。

うに表される。

$$\log p_\theta(\mathbf{x}^{(i)}) \geq \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x}, \mathbf{z})] \quad (2)$$

また,

$$\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) = -D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})} [\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})] \quad (3)$$

下限  $\mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$  を, 変分パラメータ  $\phi$  と生成パラメータ  $\theta$  について微分し, 最適化することを考える。しかし, 下限の  $\phi$  に関する勾配の計算に少し問題がある。こういった問題に対する, 通常の (ナイーブな) モンテカルロ勾配推定は次のようになる。

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})} [f(\mathbf{z})] = \mathbb{E}_{q_\phi(\mathbf{z})} [f(\mathbf{z}) \nabla_{q_\phi(\mathbf{z})} \log q_\phi(\mathbf{z})] \simeq \frac{1}{L} \sum_{l=1}^L f(\mathbf{z}) \nabla_{q_\phi(\mathbf{z}^{(l)})} \log q_\phi(\mathbf{z}^{(l)})$$

ここで,  $\mathbf{z}^{(l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$  である。このように勾配を推定するとバリエーションが非常に大きくなり (例えば [BJP12]), 我々の目的にとっては実用的でないことがわかる。

## 2.3 SGVB 推定器と AEVB アルゴリズム

この節では, 下限とそのパラメータについての微分に対して, 実用的な推定法を導入する。いま, 近似的事後分布を  $q_\phi(\mathbf{z}|\mathbf{x})$  の形式で仮定しているが, この推定法は  $q_\phi(\mathbf{z})$  の場合, つまり  $\mathbf{x}$  について条件を考えない場合にしか適用できないことに注意されたい。事後分布の推定に関する完全な変分ベイズ法は付録に示した。

2.4 節で説明する, 事後分布の近似  $q_\phi(\mathbf{z}|\mathbf{x})$  に関するある緩い条件の下では, 確率変数  $\tilde{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$  を, (補助) ノイズ変数  $\boldsymbol{\epsilon}$  の微分可能な変換  $g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$  を用いて再パラメータ化でき,

$$\tilde{\mathbf{z}} = g_\phi(\boldsymbol{\epsilon}, \mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \quad (4)$$

適切な分布  $p(\boldsymbol{\epsilon})$  と関数  $g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$  を選ぶ方法については, 2.4 節を参照されたい。いま, これを用いて  $q_\phi(\mathbf{z}|\mathbf{x})$  に関するある関数  $f(\mathbf{z})$  の期待値のモンテカルロ推定を定式化すると次のようになる。

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})} [f(\mathbf{z})] = \mathbb{E}_{p(\boldsymbol{\epsilon})} [f(g_\phi(\boldsymbol{\epsilon}, \mathbf{x}^{(i)}))] \simeq \frac{1}{L} \sum_{l=1}^L f(g_\phi(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)})) \quad \text{where} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon}) \quad (5)$$

この手法を変分下限 (式 (2)) に適用すると, 確率的勾配変分ベイズ (SGVB) 推定器  $\tilde{\mathcal{L}}^A(\theta, \phi; \mathbf{x}^{(i)}) \simeq \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$  が得られ,

$$\begin{aligned} \tilde{\mathcal{L}}^A(\theta, \phi; \mathbf{x}^{(i)}) &= \frac{1}{L} \sum_{l=1}^L \log p_\theta(\mathbf{x}^{(i)}, \mathbf{z}^{(i,l)}) - \log q_\phi(\mathbf{z}^{(i,l)}|\mathbf{x}^{(i)}) \\ \text{where} \quad \mathbf{z}^{(i,l)} &= g_\phi(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(i,l)} \sim p(\boldsymbol{\epsilon}) \end{aligned} \quad (6)$$

式 (3) の KL 情報量  $D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z}))$  は解析的に積分可能 (付録 B 参照) であることが多い。再構成期待誤差  $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})} [\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})]$  がサンプリングによる推定しか必要としない場合に限るが。これより, KL 情報量の項は, 近似した事後分布を事前分布  $p(\mathbf{z})$  に近づけ,  $\phi$  を正則化していると解釈できる。このことから, 式 (3) に対応した, もう一つの SGVB 推定器  $\tilde{\mathcal{L}}^B(\theta, \phi; \mathbf{x}^{(i)}) \simeq \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)})$  が得られる。これは一般の推

定器よりもバリエーションが小さくなる傾向にある。

$$\begin{aligned} \tilde{\mathcal{L}}^B(\theta, \phi; \mathbf{x}^{(i)}) &= -D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})) + \frac{1}{L} \sum_{l=1}^L \log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}) \\ \text{where } \mathbf{z}^{(i,l)} &= g_\phi(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(i,l)} \sim p(\boldsymbol{\epsilon}) \end{aligned} \quad (7)$$

$N$  個のデータを持つデータセット  $\mathbf{X}$  のうち複数のデータを用い、ミニバッチに基づいてデータセット全体の周辺尤度の下限を推定すると次のようになる。

$$\mathcal{L}(\theta, \phi; \mathbf{X}) \simeq \tilde{\mathcal{L}}^M(\theta, \phi; \mathbf{X}^M) = \frac{N}{M} \sum_{i=1}^M \tilde{\mathcal{L}}(\theta, \phi; \mathbf{x}^{(i)}) \quad (8)$$

ここで、ミニバッチ  $\mathbf{X}^M = \left\{ \mathbf{x}^{(i)} \right\}_{i=1}^M$  は  $N$  個のデータを持つデータセット  $\mathbf{X}$  全体からランダムに  $M$  個選んで作られたものである。実験において、各データに対するサンプル数  $L$  は、ミニバッチサイズが十分大きければ、例えば  $M = 100$  ならば、1 としてよいことがわかった。以上より、微分  $\nabla_{\theta, \phi} \tilde{\mathcal{L}}(\theta; \mathbf{X}^M)$  は計算できるようになった。この勾配は SGD や Adagrad [DHS10] などの確率的最適化法において用いることができる。確率的勾配を計算する基本的な方法については、アルゴリズム 1 を参照されたい。

自己符号化器との接点は、式 (7) で与えられる目的関数を見ると明らかになる。第一項（事前分布から測った事後分布の近似の KL 情報量）は正則化項として働き、第二項は負の再構成期待誤差となる。関数  $g_\phi(\cdot)$  はデータ  $\mathbf{x}^{(i)}$  とノイズベクトル  $\boldsymbol{\epsilon}^{(i,l)}$  をそのデータの近似的事後分布のある標本に写像するように選ばれる。ここで、 $\mathbf{z}^{(i,l)} = g_\phi(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)})$  where  $\mathbf{z}^{(i,l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$  である。続いて、標本  $\mathbf{z}^{(i,l)}$  は関数  $\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$  に入力される。ここで、関数  $\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$  は  $\mathbf{z}^{(i,l)}$  を与えられたときの生成モデルにおける、データ  $\mathbf{x}^{(i)}$  の確率密度（質量）である。この項は、自己符号化器における負の「再構成誤差」に対応する。

---

**Algorithm 1** 自己符号化変分ベイズ (AEVB) アルゴリズムのミニバッチ版。2.3 節の 2 つの SGVB 推定器は両方とも用いられている。実験では  $M = 100$ ,  $L = 1$  とした。

---

```

 $\theta, \phi \leftarrow$  Initialize parameters
repeat
   $\mathbf{X}^M \leftarrow$  Random minibatch of  $M$  datapoints (drawn from full dataset)
   $\boldsymbol{\epsilon} \leftarrow$  Random samples from noise distribution  $p(\boldsymbol{\epsilon})$ 
   $g \leftarrow \nabla_{\theta, \phi} \tilde{\mathcal{L}}^M(\theta, \phi; \mathbf{X}^M, \boldsymbol{\epsilon})$  (Gradients of minibatch estimator (8))
   $\theta, \phi \leftarrow$  Update parameters using gradients  $g$  (e.g. SGD or Adagrad [DHS10])
until convergence of parameters  $\theta, \phi$ 
return  $\theta, \phi$ 

```

---

## 2.4 再パラメータ化法

$q_\phi(\mathbf{z}|\mathbf{x})$  からサンプルを生成する、新しい手法を導入しよう。本質的なパラメータ化の方法は、至って単純である。 $\mathbf{z}$  を連続型確率変数とし、 $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$  をある条件付確率とする。このとき、多くの場合では、確率変数  $\mathbf{z}$  を決定的変数  $\mathbf{z} = g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$  と表すことができる。ここで、 $\boldsymbol{\epsilon}$  は独立な周辺分布を持つ補助変数であり、 $g_\phi(\cdot)$  は  $\phi$  によってパラメータ化されたあるベクトル値関数である。

再パラメータ化は、今回扱うような場合に対して有用である。これは、再パラメータ化が、 $q_\phi(\mathbf{z}|\mathbf{x})$  の期待値を、そのモンテカルロ推定が  $\phi$  について微分可能であるような、別の表現に書き換えるために用いることができるからである。証明は次の通りである。決定的写像  $\mathbf{z} = g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$  が与えられたとき、 $q_\phi(\mathbf{z}|\mathbf{x}) \prod_i dz_i = p(\boldsymbol{\epsilon}) \prod_i d\epsilon_i$  が自明に導かれる。よって\*1,

$$\int q_\phi(\mathbf{z}|\mathbf{x}) f(\mathbf{z}) d\mathbf{z} = \int p(\boldsymbol{\epsilon}) f(\mathbf{z}) d\boldsymbol{\epsilon} = \int p(\boldsymbol{\epsilon}) f(g_\phi(\boldsymbol{\epsilon}, \mathbf{x})) d\boldsymbol{\epsilon}$$

これに続いて、微分の推定も行われる。

$$\int q_\phi(\mathbf{z}|\mathbf{x}) f(\mathbf{z}) d\mathbf{z} \simeq \frac{1}{L} \sum_{l=1}^L f(g_\phi(\mathbf{x}, \boldsymbol{\epsilon}^{(l)}))$$

ここで、 $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$  である。2.3 節では、変分下限の微分可能な推定器を求めるときにこの手法を適用した。

一変量ガウス分布を例にとって考える。 $z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2)$  とする。このとき、適切な再パラメータを行うと、 $z = \mu + \sigma\epsilon$  となる。ここで、 $\epsilon$  は補助変数  $\epsilon \sim \mathcal{N}(0, 1)$  である。よって、

$$\mathbb{E}_{\mathcal{N}(z;\mu,\sigma^2)}[f(z)] = \mathbb{E}_{\mathcal{N}(\epsilon;0,1)}[f(\mu + \sigma\epsilon)] \simeq \frac{1}{L} \sum_{l=1}^L f(\mu + \sigma\epsilon^{(l)})$$

ここで、 $\epsilon^{(l)} \sim \mathcal{N}(0, 1)$  である。

どのような  $q_\phi(\mathbf{z}|\mathbf{x})$  に対してならば、そのような微分可能な変換  $g_\phi(\cdot)$  と補助変数  $\epsilon \sim p(\epsilon)$  を選べるだろうか。基本的なアプローチとして、次の3つが挙げられる。

#### 1. 計算可能な逆累積分布関数

この場合では、 $\boldsymbol{\epsilon} \sim \mathcal{U}(\mathbf{0}, \mathbf{I})$  とし、 $g_\phi(\boldsymbol{\epsilon}, \mathbf{x})$  が  $q_\phi(\mathbf{z}|\mathbf{x})$  の逆累積分布関数とする。

例：指数・Cauchy・Logistic・Rayleigh・Pareto・Weibull・相反・Gompertz・Gumbel・Erlang 分布

#### 2. 「位置-スケール」族

ガウス分布の例に関するアナロジーとして、任意の「位置-スケール」族の分布については、補助変数  $\epsilon$  として標準分布（location = 0, scale = 1）を用い、 $g(\cdot) = \text{location} + \text{scale} \cdot \epsilon$  とすればよい。

例：Laplace・Elliptical・Student's t・Logistic・Uniform・Triangular・Gaussian 分布

#### 3. 合成

確率変数を異なる補助変数の変換の合成として表すことができる場合も多い。

例：対数正規分布（正規分布に従う変数のべき乗）・Gamma（指数分布に従う変数の和）・Dirichlet（Gamma 分布に従う変数の重み付け和）・Beta・Chi-Squared・F 分布

## 3 具体例：変分自己符号化器

この節では、確率的符号器  $q_\phi(\mathbf{z}|\mathbf{x})$ （生成モデル  $p_\theta(\mathbf{x}, \mathbf{z})$ ）の事後分布の近似）にニューラルネットワークを用い、そのパラメータ  $\phi$  と  $\theta$  を AEVB アルゴリズムでまとめて最適化する例を示す。

潜在変数の事前分布を標準多変量ガウス分布  $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$  とする。このとき、事前分布はパラメータを持たないことに注意されたい。 $p_\theta(\mathbf{x}|\mathbf{z})$  を多変量ガウス分布（実数値データの場合）、もしくはベルヌーイ分

\*1 無限小に対し  $d\mathbf{z} = \prod_i dz_i$  と表すことができることに注意されたい

布（二値データの場合）とし、その分布のパラメータは  $\mathbf{z}$  を元に MLP（1 層の隠れ層を持つ全結合のニューラルネットワークのこと、付録 C 参照）によって計算されるとしよう。真の事後分布  $p_\theta(\mathbf{z}|\mathbf{x})$  はこの場合では計算不可能であることに注意されたい。 $q_\phi(\mathbf{z}|\mathbf{x})$  としては様々な分布が考えられるが、ここでは近似的に、共分散行列が対角成分しか持たないようなガウス分布に従うとしよう。このとき、変分近似された事後分布は、共分散行列が対角成分しか持たないようなガウス分布に従い\*2、

$$\log q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)} \mathbf{I}) \quad (9)$$

ここで、その平均と標準偏差  $\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)}$  は、データ  $\mathbf{x}^{(i)}$  と変分パラメータ  $\phi$  の非線形関数である符号化 MLP の出力となっている（付録 C 参照）。

2.4 節で説明したように、事後分布  $\mathbf{z}^{(i,l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$  からのサンプリングは、 $\boldsymbol{\varepsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  なる  $\boldsymbol{\varepsilon}$  を用いて  $\mathbf{z}^{(i,l)} = g_\phi(\mathbf{x}^{(i)}, \boldsymbol{\varepsilon}^{(l)}) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\varepsilon}^{(l)}$  とすることで得られる。ここで、 $\odot$  は要素ごとの積を表す。このモデルにおいては、 $p_\theta(\mathbf{z})$ （事前分布）と  $q_\phi(\mathbf{z}|\mathbf{x})$  がガウス分布となる。このとき、式 (7) の推定器を用いることができ、KL 情報量とその微分を、推定でなく計算することができる（付録 B 参照）。結局、データ  $\mathbf{x}^{(i)}$  に対するこのモデルの推定器は次のようになる。

$$\begin{aligned} \mathcal{L}(\theta, \phi; \mathbf{x}^{(i)}) &\simeq \frac{1}{2} \sum_{j=1}^J \left\{ 1 + \log \left( (\sigma_j^{(i)})^2 \right) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2 \right\} + \frac{1}{L} \sum_{l=1}^L \log p_\theta(\mathbf{x}^{(i)} | \mathbf{z}^{(i,l)}) \\ \text{where } \mathbf{z}^{(i,l)} &= \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\varepsilon}^{(l)} \quad \text{and } \boldsymbol{\varepsilon}^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \end{aligned} \quad (10)$$

上と付録 C で説明した通り、復号化項  $\log p_\theta(\mathbf{x}^{(i)} | \mathbf{z}^{(i,l)})$  はベルヌーイ MLP またはガウス MLP で、データの種類によって使い分けることとなる。

## 4 関連研究

我々の知る限りでは、連続型の潜在変数を持つモデルに適用可能なオンライン学習の手法としては、提案手法を除くと唯一 Wake-Sleep アルゴリズム [HDFN95] が挙げられる。提案手法と同様に、Wake-Sleep アルゴリズムでは認識モデルを用いて真の事後分布の近似を行っている。Wake-Sleep アルゴリズムの欠点としては、2 つの目的関数の同時最適化が必要となることが挙げられる。この同時最適化は周辺尤度の（下限の）最適化にはならない。Wake-Sleep アルゴリズムの利点としては、離散型の潜在変数にも対応可能な点が挙げられる。Wake-Sleep アルゴリズムは、各データに対して AEVB と同等の計算量がかかる。

近年、確率の変分推論 [HBWP13] に注目が集まっている。[BJP12] では制御変分法によって 2.1 節で議論されたような単純な勾配推定器における高いバリエーションを軽減し、事後分布の指数分布族の近似に適用した。[RGB13] では、制御変分法など、一部の一般に用いることのできる手法を用いて、原形の勾配推定器のバリエーションを軽減した。[SK13] では、本論文と類似した再パラメータ化を行い、指数分布族の自然パラメータの学習に対する、確率の変分推論アルゴリズムの効率化を図った。

AEVB アルゴリズムは有向確率モデル（変分的目的関数で訓練される）と自己符号化器の間の関係性を明らかにした。線形自己符号化器と一部の線形ガウス生成モデルの間の関係は既に知られていた。[Row98] では、PCA が、十分小さい  $\varepsilon$  について、事前分布が  $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ 、条件付き確率が  $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z}, \varepsilon \mathbf{I})$  である線形ガウスモデルの最尤解に対応することが示された。

\*2 これは（単純化した）1 つの選択肢であり、この手法はこれにしか適用できないわけではない。

自己符号化器 [VLL+10] に関する最近の研究では、正則化を行わずに訓練した自己符号化器は、入力  $\mathbf{X}$  と潜在表現  $\mathbf{z}$  の間の相互情報量の下限の最大化に対応しているということが示された（情報量最大化原理 [Lin89] を参照のこと）。（パラメータについての）相互情報量の最大化は条件付きエントロピーの最大化と等価である。なお、自己符号化器 [VLL+10] において条件付きエントロピーは期待対数尤度、つまり負の再構成誤差で下から抑えられている。しかし、再構成はそれ自体では有用な表現を学習できないことが知られている [BCV13]。自己符号化器に有用な表現を学習させるために、ノイズ除去自己符号化器・スパース自己符号化器・収縮自己符号化器などの正則化技術が提案されてきた [BCV13]。SGVB の目的関数には変分下限による正則化項があり（式 (10) など）、よくある正則化のためのハイパーパラメータは必要ない。我々が着想を得たものでもあるが、予測的スパース分解 (PSD) [KRL08] などの符号-復号的な構造も関連研究として挙げられる。最近提案された、データ分布からのサンプリングを行うマルコフ連鎖の遷移演算子を、ノイズ自己符号化器によって担う生成確率ネットワーク [BTL13] も関連研究として挙げられる。[SL10] では深層ボルツマンマシンの学習に認識モデルが用いられた。提案手法は有向確率モデルの学習にのみ用いられるのに対し、これらの手法は、正則化されていないモデル（つまりボルツマンマシンのような無向モデル）においても、スパースな符号化モデルにおいても用いられる。

近年提案された DARN 手法 [GMW13] においても自己符号化的な構造を用いて有向確率モデルの学習を行っているが、このモデルは二値の潜在変数にしか適用できない。また更に最近のものでは、[RMW14] で自己符号化器、有向確率モデルと確率の変分推論の関係性を、本論文で用いたような再パラメータ化を用いて説明している。彼らの研究は我々のものとは独立に発展したもので、AEVB に対して新たな知見を与えた。

## 5 実験

我々は、画像の生成モデルを、MNIST と Frey Face データセット<sup>\*3</sup> を用いて訓練し、学習アルゴリズムを変分下限と推定周辺尤度に関して比較した。

第 3 節の生成モデル（符号器）と変分近似（復号器）を用い、それらの隠れユニット数は同一とする。Frey Face データセットのデータは連続的であるため、復号器には符号器と同じガウス分布のパラメータを出力させる。ただし、復号器の出力はシグモイド活性化関数に入れられていて、その平均値は  $(0, 1)$  の区間に制限される点は異なっている。なお、「隠れユニット」は、符号器・復号器のニューラルネットワークの隠れ層のユニットを指す。

パラメータは確率的勾配上昇法を用いて更新され、勾配は下限推定器  $\nabla_{\theta, \phi} \mathcal{L}(\theta, \phi; \mathbf{X})$ （アルゴリズム 1 参照）に、事前分布  $p(\theta) = \mathcal{N}(\mathbf{0}, \mathbf{I})$  に対応した重み減衰項を足したものを微分して得られる。この目的関数を最適化することは、尤度勾配が下限の勾配で近似されている点から、MAP 推定の近似を行うことに対応する。

AEVB アルゴリズムは wake-sleep アルゴリズム [HDFN95] と比較した。wake-sleep アルゴリズムと変分自己符号化器には同一の符号器（認識モデルとも呼ぶ）を用いた。変分モデル・生成モデルの全てのパラメータは  $\mathcal{N}(0, 0.01)$  からのサンプリングにより初期化され、MAP 基準を用い、まとめて確率的に最適化された。学習率は Adagrad [DHS10] を用いて調整した。Adagrad の大域的な学習率のパラメータは、学習初期の数イタレーションでの、訓練データにおける性能に基づいて  $\{0.01, 0.02, 0.1\}$  から選ばれた。ミニバッチサイズは  $M = 100$  とし、各データについて  $L = 1$  サンプルを用いた。

<sup>\*3</sup> <https://cs.nyu.edu/roweis/data.html>



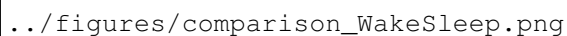
■尤度の下限 我々は、生成モデル（復号器）と対応する符号器（認識モデル）を、MNIST では 500 隠れユニット、Frey Face では 200 隠れユニットとして訓練した（Frey Face は比較的小さいデータセットであり、過学習を防ぐために少なくした）。隠れユニット数は自己符号化器の先行研究に基づいて選んだ。なお、隠れユニット数は、異なるアルゴリズム間の性能評価に対しては大きくは寄与しない。図 2 に下限の値の比較を示した。興味深いことに、余計な潜在変数を加えても過学習は起こらなかった。これは変分下限が持つ正則化的な性質から説明される。

■周辺尤度 潜在空間が非常に低次元な場合は、学習された生成モデルの周辺尤度を、MCMC 推定器を用いて推定することが可能になる。周辺尤度の推定器に関しては、付録に詳しく示した。符号器・復号器には隠れユニット数 100、潜在変数 3 のニューラルネットワークを用いる。高次元の潜在空間では信頼性が下がるためである。また、データセットは MNIST を用いた。詳細は付録に示すが、AEVB と Wake-Sleep 法をハイブリッドモンテカルロサンプリング [DKPR87] を用いたモンテカルロ EM 法（MCEM 法）と比較した。訓練データのサイズとして、大きいものと小さいものに対し、3 つのアルゴリズムの収束速度を比較した。図 3 に結果を示した。

■高次元データの可視化 潜在空間を低次元（2 次元など）とすれば、学習させた符号器（認識モデル）を用いて高次元データを低次元の多様体に写像することができる。付録 A に MNIST と Frey Face データセットに対する 2 次元の潜在多様体を可視化したものを示した。

## 6 結論

本論文では、連続型の潜在変数に対して効率的に近似推論を行うために、変分下限についての新しい推定器、確率的勾配変分ベイズ（SGVB）を導入した。提案した推定器は標準的な確率的勾配法を用いて直接的に



```
../figures/comparison_WakeSleep.png
```

図 2: 様々な潜在空間の次元  $N_z$  に対する、下限の最適化という観点での AEVB アルゴリズムと Wake-Sleep アルゴリズムの比較。我々の手法は全ての設定で非常に早くかつより良い解に至っている。興味深いことに、潜在変数を増やしても過学習は起こらなかった。これは下限の正則化効果から説明される。縦軸は推定平均下限である。推定器のバリエーションは小さく（ $< 1$ ）無視できる。横軸は訓練データ数である。100 万訓練データの計算には、40 GFLOPS で稼働する Intel Xeon CPU を用いて 20–40 分かかった。



../figures/comparison\_MCMC.png

図 3: 推定された周辺尤度についての, AEVB と Wake-Sleep アルゴリズム, モンテカルロ EM 法との比較。モンテカルロ EM 法はオンライン学習のできるアルゴリズムではなく, (AEVB や Wake-Sleep 法とは異なり) MNIST データセット全体に対して効率的に適用することができない。

微分し最適化することができる。また, 各データに対し連続型の潜在変数があるような i.i.d なデータセットに対して効率的に学習・推論を行うために, 近似推論モデルを SGVB 推定器を用いて学習する, 自己符号化変分ベイズ (AEVB) アルゴリズムを導入した。この理論の利点は実験結果に表れている。

## 7 今後の展望

SGVB 推定器と AEVB アルゴリズムは, 連続型の潜在変数を考えるほとんど全ての学習・推論問題に適用することができるため, 今後の展望としては様々なものが挙げられる。

- (i) 深層ニューラルネットワーク (畳み込みネットワークなど) を符号器と復号器に用い, AEVB を用いて訓練を行うことによる, 階層的な生成構造の学習
- (ii) 時系列モデル (つまり動的ベイジアンネットワーク)
- (iii) 大域的パラメータへの SGVB の適用
- (iv) 複雑なノイズ分布の学習に対して有用となる, 潜在変数のある教師あり学習

## 付録 A 可視化

図 4, 5 に SGVB で学習を行ったモデルについて, 潜在空間と対応する観測空間の可視化を示した。



図 4: AEVB で学習した 2 次元の潜在空間を持つ生成モデルの，学習されたデータの多様体の可視化。潜在空間の事前分布はガウス分布であるため，方眼状に線形に配置された座標をガウス分布の逆累積分布関数によって変換することで潜在変数  $\mathbf{z}$  を得る。これらの  $\mathbf{z}$  の値それぞれに対し，学習した  $\theta$  について対応する  $p_{\theta}(\mathbf{x}|\mathbf{z})$  をプロットした。

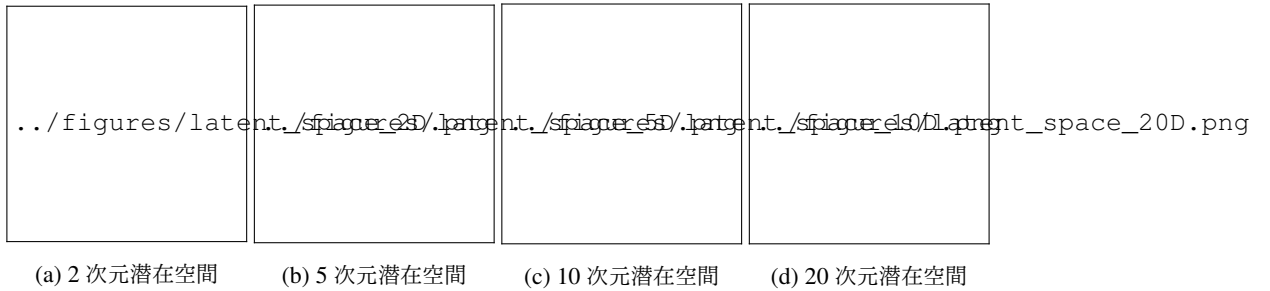


図 5: 様々な潜在空間の次元について，学習した MNIST の生成モデルからランダムにサンプリングしたもの。

## 付録 B ガウス分布に対する $D_{\text{KL}}(q_{\phi}(\mathbf{z})||p_{\theta}(\mathbf{z}))$ の解

変分下限（最大化したい目的関数）は KL 情報量の項を含んでいるが，これは解析的に積分することで計算されることが多い。ここでは，事前分布  $p_{\theta}(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$  かつ事後分布の近似  $q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})$  がガウス分布であるときの解を与える。 $\mathbf{z}$  の次元を  $J$  とする。 $\boldsymbol{\mu}$  と  $\boldsymbol{\sigma}$  をデータ  $i$  に対して計算された変分平均と標準偏差とし， $\mu_j$

と  $\sigma_j$  でそれらのベクトルの  $j$  番目の要素を指すこととする。このとき、

$$\begin{aligned}\int q_\theta(\mathbf{z}) \log p(\mathbf{z}) d\mathbf{z} &= \int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \log \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}) d\mathbf{z} \\ &= -\frac{J}{2} \log(2\pi) - \frac{1}{2} \sum_{j=1}^J (\mu_j^2 + \sigma_j^2)\end{aligned}$$

また、

$$\begin{aligned}\int q_\theta(\mathbf{z}) \log q_\theta(\mathbf{z}) d\mathbf{z} &= \int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) d\mathbf{z} \\ &= -\frac{J}{2} \log(2\pi) - \frac{1}{2} \sum_{j=1}^J (1 + \log \sigma_j^2)\end{aligned}$$

よって、

$$\begin{aligned}-D_{\text{KL}}(q_\theta(\mathbf{z}) || p_\theta(\mathbf{z})) &= \int q_\theta(\mathbf{z}) (\log p_\theta(\mathbf{z}) - \log q_\theta(\mathbf{z})) d\mathbf{z} \\ &= \frac{1}{2} \sum_{j=1}^J \{1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2\}\end{aligned}$$

本文で例示した通り、認識モデル  $q_\theta(\mathbf{z}|\mathbf{x})$  を用いるときは、 $\boldsymbol{\mu}$  と標準偏差  $\boldsymbol{\sigma}$  は  $\mathbf{x}$  と変分パラメータ  $\boldsymbol{\phi}$  のみの関数となる。

## 付録 C 確率的符号器・復号器としての MLP

変分自己符号化器においては、ニューラルネットワークが確率的符号器・復号器として用いられる。符号器・復号器には、データとモデルの種類に応じて様々なものを用いることができる。我々は比較的単純なニューラルネットワーク、所謂多層パーセプトロン（MLP）を用いた。符号器にはガウス分布のパラメータを出力する MLP を、復号器にはデータの種類に応じてガウス分布もしくはベルヌーイ分布のパラメータを出力する MLP を用いた。

### C.1 復号器にベルヌーイ分布のパラメータを出力する MLP を用いるとき

このとき、 $p_\theta(\mathbf{x}|\mathbf{z})$  を、 $\mathbf{z}$  から単層の隠れ層を持つ全結合ニューラルネットワークを用いて計算される多変量ベルヌーイ分布とする。

$$\begin{aligned}\log p(\mathbf{x}|\mathbf{z}) &= \sum_{i=1}^D x_i \log y_i + (1 - x_i) \cdot \log(1 - y_i) \\ \text{where } \mathbf{y} &= f_\sigma(\mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2)\end{aligned} \tag{11}$$

ここで、 $f_\sigma(\cdot)$  は要素ごとのシグモイド活性化関数であり、また、 $\theta = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$  は MLP の重みとバイアスである。

## C.2 符号器・復号器にガウス分布のパラメータを出力する MLP を用いるとき

このとき，符号器または復号器を共分散行列が対角成分のみを持つ多変量ガウス分布として，

$$\begin{aligned}\log p(\mathbf{x}|\mathbf{z}) &= \log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2 \mathbf{I}) \\ \text{where } \boldsymbol{\mu} &= \mathbf{W}_4 \mathbf{h} + \mathbf{b}_4 \\ \log \boldsymbol{\sigma}^2 &= \mathbf{W}_5 \mathbf{h} + \mathbf{b}_5 \\ \mathbf{h} &= \tanh(\mathbf{W}_3 \mathbf{z} + \mathbf{b}_3)\end{aligned}\tag{12}$$

ここで， $\{\mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5\}$  は MLP の重みとバイアスであり，復号器として用いるときは  $\theta$  の一部となる。このネットワークが復号器  $q_\phi(\mathbf{z}|\mathbf{x})$  として用いられるときは， $\mathbf{z}$  と  $\mathbf{x}$  が交換され，重みとバイアスは変分パラメータ  $\phi$  となることに注意されたい。

## 付録 D 周辺尤度の推定器

我々は，低次元空間（5 次元以下）でかつ十分なサンプル数があるときにいい推定を与える周辺尤度の推定器を導出した。 $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$  を生成モデルとし，ここからサンプリングすることを考える。データ  $\mathbf{x}^{(i)}$  に対して，周辺尤度  $p_\theta(\mathbf{x}^{(i)})$  を推定する。

推定の過程は以下の 3 段階に分けられる。

1. 事後分布から， $\nabla_{\mathbf{z}} \log p_\theta(\mathbf{z}|\mathbf{x}) = \nabla_{\mathbf{z}} \log p_\theta(\mathbf{z}) + \nabla_{\mathbf{z}} \log p_\theta(\mathbf{x}|\mathbf{z})$  を使い，ハイブリッドモンテカルロ法などの勾配に基づく MCMC 法によって， $L$  個の値  $\{\mathbf{z}^{(l)}\}$  をサンプリングする
2. 密度推定器  $q(\mathbf{z})$  にこれらのサンプル  $\{\mathbf{z}^{(l)}\}$  を代入する
3. 再び事後分布から  $L$  個のサンプリングを行う。これらのサンプルを， $q(\mathbf{z})$  と同様に，次の推定器に代入する。

$$p_\theta(\mathbf{x}^{(i)}) \simeq \left( \frac{1}{L} \sum_{l=1}^L \frac{q(\mathbf{z}^{(l)})}{p_\theta(\mathbf{z})p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(l)})} \right)^{-1} \quad \text{where } \mathbf{z}^{(l)} \sim p_\theta(\mathbf{z}|\mathbf{x}^{(i)})$$

この推定器の導出は以下の通りである。

$$\begin{aligned}\frac{1}{p_\theta(\mathbf{x}^{(i)})} &= \frac{\int q(\mathbf{z}) d\mathbf{z}}{p_\theta(\mathbf{x}^{(i)})} \\ &= \frac{\int q(\mathbf{z}) \frac{p_\theta(\mathbf{x}^{(i)}, \mathbf{z})}{p_\theta(\mathbf{x}^{(i)}, \mathbf{z})} d\mathbf{z}}{p_\theta(\mathbf{x}^{(i)})} \\ &= \int \frac{p_\theta(\mathbf{x}^{(i)}, \mathbf{z})}{p_\theta(\mathbf{x}^{(i)})} \frac{q(\mathbf{z})}{p_\theta(\mathbf{x}^{(i)}, \mathbf{z})} d\mathbf{z} \\ &= \int p_\theta(\mathbf{z}|\mathbf{x}^{(i)}) \frac{q(\mathbf{z})}{p_\theta(\mathbf{x}^{(i)}, \mathbf{z})} d\mathbf{z} \\ &\simeq \frac{1}{L} \sum_{l=1}^L \frac{q(\mathbf{z}^{(l)})}{p_\theta(\mathbf{z})p_\theta(\mathbf{x}^{(i)}|\mathbf{z})} \\ \text{where } \mathbf{z}^{(l)} &\sim p_\theta(\mathbf{z}|\mathbf{x}^{(i)})\end{aligned}$$

## 付録 E モンテカルロ EM 法

モンテカルロ EM 法では符号器を用いず，代わりに潜在変数の事後分布から，その勾配を  $\nabla_{\mathbf{z}} \log p_{\theta}(\mathbf{z}|\mathbf{x}) = \nabla_{\mathbf{z}} \log p_{\theta}(\mathbf{z}) + \nabla_{\mathbf{z}} \log p_{\theta}(\mathbf{x}|\mathbf{z})$  を用いて計算し，サンプリングを行う。モンテカルロ EM 法は 10HMC リープフロッグステップから成り，その学習率は，その引受率を 90% とし，得られたサンプルを用いて 5 ステップで重みを更新することで，自動的に調整される。全てのアルゴリズムに対し，パラメータは Adagrad の学習率を用いて更新された（焼きなまし法も用いた）。

周辺尤度は訓練データ及びテストデータの初めの 1,000 データから推定した。ハイブリッドモンテカルロ法をリープフロッグステップを 4 とし，各データに対し潜在変数の事後分布から 50 個をサンプリングした。