# Truncated Variational Sampling for 'Black Box' Optimization of Generative Models

Jörg Lücke
joerg.luecke@uol.de
Universität Oldenburg and
Cluster of Excellence H4a
Oldenburg, Germany

Zhenwen Dai
zhenwend@amazon.com
Amazon Research
Cambridge, UK

Georgios Exarchakis
georgios.exarchakis@ens.fr
Department d'Informatique
École Normale Supérieure
Paris, France

**Abstract**

We investigate the optimization of two probabilistic generative models with binary latent variables using a novel variational EM approach. The approach distinguishes itself from previous variational approaches by using latent states as variational parameters. Here we use efficient and general purpose sampling procedures to vary the latent states, and investigate the 'black box' applicability of the resulting optimization procedure. For general purpose applicability, samples are drawn from approximate marginal distributions of the considered generative model as well as from the model's prior distribution. As such, variational sampling is defined in a generic form, and is directly executable for a given model. As a proof of concept, we then apply the novel procedure (A) to Binary Sparse Coding (a model with continuous observables), and (B) to basic Sigmoid Belief Networks (which are models with binary observables). Numerical experiments verify that the investigated approach efficiently as well as effectively increases a variational free energy objective without requiring any additional analytical steps.

## 1 Introduction

The use of expectation maximization (EM) for advanced probabilistic data models requires approximations because EM with an exact E-step (computing the full posterior) is typically intractable. Many models of recent interest have binary latents [1, 2, 3, 4], and for such models these intractabilities are primarily computational: exact E-steps can be computed but they scale exponentially with the number of latents. To overcome intractabilities, overcome intractabilities for models with binary latents there are typically two types of approaches applied: sampling approaches or variational EM with the latter having been dominated by factored variational approaches in the past [e.g. 5]. Variational approaches and sampling have also often been combined [6, 7, 8, 4] to leverage the advantages of both methods. However, given a generative model, both approximations require often cumbersome derivations either to derive efficient posterior samplers or to derive update equations for variational parameter optimization. The question how procedures can be defined that automatize the development of learning algorithms for generative models has therefore shifted into the focus of recent research [9, 10, 11, 12, 4]. In this paper, we make use of truncated approximations to EM which have repeatedly been applied before [13, 4, 14]. Here we show how novel theoretical results on truncated variational distributions [15] can be used to couple variational EM and sampling exceptionally tightly. This coupling then enables "black box" applicability.

## 2 Truncated Posteriors and Sampling

Let us consider generative models with $H$ binary latent variables, $\vec{s} = (s_1, \ldots, s_H)$ with $s_h \in \{0, 1\}$. Truncated approximations have been motivated by the observation that the exponentially large sums over states required

for expectation values w.r.t. posteriors are typically dominated by summands corresponding to very few states. If for a given data point $\vec{y}^{(n)}$ these few states are contained in a set $\mathcal{K}^{(n)}$, we can define a posterior approximation as follows [compare 13, 4]:

$$q^{(n)}(\vec{s};\mathcal{K},\Theta) = \frac{p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)\,\delta(\vec{s}\in\mathcal{K}^{(n)})}{\sum\limits_{\vec{s}\,'\in\mathcal{K}^{(n)}} p(\vec{s}\,'\,|\,\vec{y}^{(n)},\Theta)}\,, \tag{1}$$

where $\delta(\vec{s}\in\mathcal{K}^{(n)}) = 1$ if $\mathcal{K}^{(n)}$ contains $\vec{s}$ and zero otherwise. It is straight-forward to derive expectation values w.r.t. these approximate posteriors simply by inserting (1) into the definition of expectation values and by multiplying numerator and denominator by $p(\vec{y}^{(n)}\,|\,\Theta)$, which yields:

$$\langle g(\vec{s})\rangle_{q^{(n)}} \;=\; \frac{\sum\limits_{\vec{s}\in\mathcal{K}^{(n)}} p(\vec{s},\vec{y}^{(n)}\,|\,\Theta)\;g(\vec{s})}{\sum\limits_{\vec{s}\,'\in\mathcal{K}^{(n)}} p(\vec{s}\,',\vec{y}^{(n)}\,|\,\Theta)} \tag{2}$$

where $g(\vec{s})$ is a function of the hidden variables. As the dominating summands are different for each data point $\vec{y}^{(n)}$, the sets $\mathcal{K}^{(n)}$ are different. If a set $\mathcal{K}^{(n)}$ now contains those states $\vec{s}$ which dominate the sums over the joints w.r.t. the exact posterior, then Eqn. 2 is a very accurate approximation.

Truncated posterior approximations have successfully been applied to a number of elementary and more advanced generative models, and they do not suffer from potential biases introduced by posterior independence assumptions made by factored variational approximations Previously, the sets $\mathcal{K}^{(n)}$ were defined based on sparsity assumptions and/or latent preselection [13, 16]. The approach followed here, in contrast, uses sets $\mathcal{K}^{(n)}$ which contain samples from model and data dependent distributions. By treating the truncated distribution (1) as variational distributions within a free-energy framework [15], we can then derive efficient procedures to update the samples in $\mathcal{K}^{(n)}$ such that the variational free-energy is always monotonically increased. For this we use the following theoretical results: (1) We use that the M-step equations remain unchanged if instead of exact posteriors the truncated posteriors (1) are used; (2) We make use of the result that after each M-step the free-energy corresponding to truncated variational distributions is given by the following simplified and computationally tractable form:

$$\mathcal{F}(\mathcal{K},\Theta) \;=\; \sum_n \log\Big(\sum_{\vec{s}\in\mathcal{K}^{(n)}} p(\vec{s},\vec{y}^{(n)}\,|\,\Theta)\Big)\,, \tag{3}$$

where $\mathcal{K} = (\mathcal{K}^{(1)},\ldots,\mathcal{K}^{(N)})$. The variational E-step then consists of finding a set $\mathcal{K}_{\text{new}}$ which increases $\mathcal{F}(\mathcal{K},\Theta)$ w.r.t. $\mathcal{K}$. The M-step consist of the standard M-step equations but with expectation values estimated by (2).

For any larger scale multiple-cause model we can not exhaustively iterate through all latent states. We therefore here seek to find new sets $\tilde{\mathcal{K}}$ using sampling, such that the free-energy is increased, $\mathcal{F}(\tilde{\mathcal{K}},\Theta) > \mathcal{F}(\mathcal{K},\Theta)$. To keep the computational demand limited, we will take the sets $\mathcal{K}$ and $\tilde{\mathcal{K}}$ to be of constant size after each E-step by demanding $|\mathcal{K}^{(n)}| = |\tilde{\mathcal{K}}^{(n)}| = S$ for all $n$. Instead of explicitly computing and comparing the free-energies (3) w.r.t. $\mathcal{K}$ and $\tilde{\mathcal{K}}$, we can instead use a comparison of joint probabilities $p(\vec{s},\,\vec{y}^{(n)}\,|\,\Theta)$ as a criterion for free-energy increase. The following can be shown [15]:

For a replacement of $\vec{s}\in\mathcal{K}^{(n)}$ by a new state $\vec{s}_{\text{new}}\notin\mathcal{K}^{(n)}$ the free-energy $\mathcal{F}(\mathcal{K},\Theta)$ is increased if and only if

$$p(\vec{s}_{\text{new}},\,\vec{y}^{(n)}\,|\,\Theta) > p(\vec{s},\,\vec{y}^{(n)}\,|\,\Theta)\,. \tag{4}$$

Criterion (4) may directly be concluded by considering the functional form of Eqn. 3 [see 15, for a formal proof]. It means that the free-energy is guaranteed to increase if we replace, e.g., the state with the lowest joint in $\mathcal{K}^{(n)}$ by a newly sampled state $\vec{s}_{\text{new}}\notin\mathcal{K}^{(n)}$ with a higher joint. Instead of comparing single joints, a computationally more efficient procedure is to use batches of many newly sampled states, and then to use criterion (4) to increase

$\mathcal{F}(\mathcal{K}, \Theta)$ as much as possible. Such a procedure is given by Alg. 1: For each data point $n$, we first draw $M$ new samples from a yet to be specified distribution $p^{(n)}_{\text{var}}(\vec{s})$. These samples are then united with the states already in $\mathcal{K}^{(n)}$. Of this union of old and new states, we then take the $S$ states with highest joints to define the new state set $\mathcal{K}^{(n)}$. This last step selects because of (4) the best possible subset of the union. Furthermore, selecting the $S$ states with largest joints represents a standard unsorted partial sorting problem which is solvable in linear time complexity, i.e., with at most $\mathcal{O}(M + S)$ in our case. Instead of selecting the $S$ largest joints, we can also remove the $(|\mathcal{K}^{(n)}| - S)$ lowest ones (last line in Alg. 1). For *any* distribution $p^{(n)}_{\text{var}}(\vec{s})$, Alg. 1 is guaranteed to monotonously increase the free-energy $\mathcal{F}(\mathcal{K}, \Theta)$ w.r.t. $\mathcal{K}$.

**Algorithm 1:** Sampling-based TV-E-step

**for** $n = 1, \ldots, N$ **do**

- draw $M$ samples $\vec{s} \sim p^{(n)}_{\text{var}}(\vec{s})$;
- define $\mathcal{K}^{(n)}_{\text{new}}$ to contain all $M$ samples;
- set $\mathcal{K}^{(n)} = \mathcal{K}^{(n)} \cup \mathcal{K}^{(n)}_{\text{new}}$;
- remove those $(|\mathcal{K}^{(n)}| - S)$ samples $\vec{s} \in \mathcal{K}^{(n)}$ with the lowest $p(\vec{s}, \vec{y}^{(n)} \,|\, \Theta)$;

## 3 Posterior, Prior, and Marginal Sampling

While the partial E-step of Alg. 1 monotonously increases the free-energy for any distribution $p^{(n)}_{\text{var}}(\vec{s})$ used for sampling, the specific choice for $p^{(n)}_{\text{var}}(\vec{s})$ is of central importance for the efficiency of the procedure. If the distribution is not chosen well, any significant increase of $\mathcal{F}(\mathcal{K}, \Theta)$ may require unreasonable amounts of time, e.g., because new samples which increase $\mathcal{F}(\mathcal{K}, \Theta)$ are sampled too infrequently. By considering Alg. 1, the requirement for $p^{(n)}_{\text{var}}(\Theta)$ is to provide samples with high joint probability $p(\vec{s}, \vec{y}^{(n)} \,|\, \Theta)$ for a given $\vec{y}^{(n)}$. The first distribution that comes to mind for $p^{(n)}_{\text{var}}(\Theta)$ is the posterior distribution $p(\vec{s} \,|\, \vec{y}^{(n)}, \Theta)$. Samples form the posterior are likely to have high posterior mass and therefore high joint mass relative to the other states because all states share the same normalizer $p(\vec{y}^{(n)} \,|\, \Theta)$. On the downside, however, sampling from the posterior may not be an easy task for models with binary latents and a relatively high dimensionality as we intend to aim at here. Furthermore, the derivation of posterior samplers requires additional analytical efforts for any new generative model we apply the procedure to, and requires potentially additional design choices such as definitions of proposal distributions. All these points are contrary to our goal of a 'black box' procedure which is applicable as generally and generically as possible. Instead, we therefore seek distributions $p^{(n)}_{\text{var}}(\Theta)$ for Alg. 1 that can efficiently optimize the free-energy but that can be defined without requiring model-specific analytical derivations. Candidates for $p^{(n)}_{\text{var}}(\Theta)$ are consequently the prior distribution of the given generative model, $p(\vec{s} \,|\, \Theta)$, or the marginal distribution. A prior sampler is usually directly given by the generative model but may have the disadvantage that finally new samples only very rarely increase the free-energy because the prior sampler is independent of a given data point (only the average over data points has high posterior mass). Marginal samplers, on the other hand, are data driven but the computation of activation probabilities $p(s_h = 1 \,|\, \vec{y}^{(n)}, \Theta)$ is unfortunately not computationally efficient. To obtain data-driven but efficient samplers, we will for our purposes, therefore, use approximate marginal samplers.

**1st Approximation.** First observe that we can obtain an efficiently computable approximation to a marginal sampler by using the truncated distributions $q^{(n)}(\vec{s})$ in (1) themselves. For binary latents $s_h$ we can approximate:

$$p(s_h = 1 \,|\, \vec{y}^{(n)}, \Theta) = \langle s_h \rangle_{p(\vec{s} \,|\, \vec{y}^{(n)}, \Theta)} \approx \langle s_h \rangle_{q^{(n)}(\vec{s})} \,, \tag{5}$$

and accordingly $p(s_h = 0 \,|\, \vec{y}^{(n)}, \Theta)$. Because of the arguments given above the expectations (2) w.r.t. $q^{(n)}(\vec{s})$ are efficiently computable using (2) with $g(\vec{s}) = \vec{s}$. Using 5 we can consequently define for each latent $h$ an approximation of the marginal $p(s_h = 1 \,|\, \vec{y}^{(n)}, \Theta)$. Given a directed generative model, no derivations are required to efficiently generate samples from this approximation because the joint probabilities to estimate $p(s_h = 1 \,|\, \vec{y}^{(n)}, \Theta)$ using (2) can directly be computed. The truncated marginal sampler defined by Eqn. 5 becomes increasingly similar to an exact marginal sampler the better the truncated distributions approximate the exact posteriors.

**2nd Approximation.** To further improve efficiency and convergence times, we optionally apply a second approximation by using the approximate marginal distributions (Eqns. 5) as target objective for a parametric

function $f_h(\vec{y}^{(n)};\Lambda)$ which approximates the truncated marginal. A parametric function from data to marginal probabilities of the latents has the advantage of modeling data similarities by mapping similar data to similar marginal distributions. The mapping incorporates information across the data points, which can facilitate training and, e.g., avoids more expensive $\mathcal{K}^{(n)}$ updates of some data points due to bad initialization. The mapping $f_h(\vec{y}^{(n)};\Lambda)$ is estimated with the training data and the current approximate marginal $q_{\mathrm{mar}}(s_h = 1 \,|\, \vec{y},\Theta)$ defined by (5) with (2). For simplicity, we use a Multi-Layer Perceptron (MLP) for the function mapping and trained with cross-entropy. We use a generic MLP with one hidden layer. As such, the MLP itself is independent of the generative model considered but optimized for the generic truncated approximation (5) which contains the model's joint. The idea of using a parametric function to approximate expectations w.r.t. intractable posteriors is an often applied technique [e.g. 17, 18, and refs therein].

For our numerical experiments we combine prior and (approximate) marginal sampling to suggest new variational states. The easy to use prior samplers are not data driven and represent rather an *exploration* strategy. Marginal sampling, on the other hand, is rather an *exploitation* strategy that produces good results when sufficiently much from the data is already known. Mixing the two has therefore turned out best for our purposes. Posterior samplers do require additional derivations but, to our experience so far, are also not necessarily better than combined prior and marginal sampling in optimizing the truncated free-energy.

**Algorithm 2:** Truncated Variational Sampling.

initialize model parameters $\Theta$;
for all $n$ init $\mathcal{K}^{(n)}$ such that $|\mathcal{K}^{(n)}| = S$;
set $M_p$; (# samples from prior distribution)
set $M_q$; (# samples from marginal distr.)
**repeat**
  update $M_p$ and $M_q$ (sampler adjustment)
  **for** $(n = 1,\ldots,N)$ **do**
    draw $M_p$ samples from $p(\vec{s}\,|\,\Theta) \rightarrow \mathcal{K}_p^{(n)}$;
    draw $M_q$ samples from $q_{\mathrm{mar}}^{(n)}(\vec{s};\Theta) \rightarrow \mathcal{K}_q^{(n)}$,
    $\mathcal{K}^{(n)} = \mathcal{K}^{(n)} \cup \mathcal{K}_p^{(n)} \cup \mathcal{K}_q^{(n)}$;
    remove those $(|\mathcal{K}^{(n)}| - S)$ elements
    $\vec{s} \in \mathcal{K}^{(n)}$ with lowest $p(\vec{s},\vec{y}^{(n)}\,|\,\Theta)$;
  $\mathcal{K} = (\mathcal{K}^{(1)},\ldots,\mathcal{K}^{(N)})$;
  use M-steps with (2) to change $\Theta$;
**until** $\Theta$ *has sufficiently converged*;

Before we consider concrete generative models, let us summarize the general novel procedure in the form of the pseudo code given by Alg. 2. First, we have to initialize the model parameters $\Theta$ and the sets $\mathcal{K}^{(n)}$. While initializing $\Theta$ can be done as for other EM approaches, one option for an initialization of $\mathcal{K}^{(n)}$ would be the use of samples from the prior given $\Theta$ (more details are given below). The inner loop (the variational E-step) of Alg. 2 is then based on a mix of prior and marginal samplers, and each of these samplers is directly defined in terms of a considered generative model, no model-specific derivations are used. The same does not apply for the M-step but we will consider two examples how this point can be addressed: (A) either by using well-known standard M-step or (B) by applying automatic differentiation. Alg. 2 will be referred to as *truncated variational sampling* (TVS).

## 4 Applications of TVS

Exemplarily, we consider two genrative models: Binary Sparse Coding and Sigmoid Belief Networks. The models are complementary in many aspects, and thus serve well as example appications.

**Binary Sparse Coding.** In the first example we will consider dictionary learning – a typical application domain of variational EM approaches and sampling approaches in general. Probabilistic sparse coding models are not computationally tractable and common approximations such as maximum a-posteriori approximations can result in suboptimal solutions. Factored variational EM as well as sampling approaches have therefore been routinely applied to sparse coding. Of particular interest for our purposes are sparse coding models with discrete or semi-discrete latents [e.g. 19, 20, 1, 4], where binary sparse coding [BSC; 19, 20] represents an elementary example.

BSC assumes independent and identically distributed (iid) binary latent variables following a Bernoulli prior

distribution, and it uses a Gaussian noise model:

$$p(\vec{s}\,|\,\Theta) \;=\; \prod_{h=1}^{H} \pi^{s_h}\,(1-\pi)^{1-s_h}\,, \qquad p(\vec{y}\,|\,\vec{s},\Theta) \;=\; \mathcal{N}(\vec{y};W\vec{s},\sigma^2\mathbb{1})\,, \tag{6}$$

where $\pi \in [0,1]$ and where $\Theta = (\pi, W, \sigma^2)$ is the set of model parameters.

As TVS is an approximate EM approach, let us first consider exact EM which seeks parameters $\Theta$ that optimize the data likelihood for the BSC data model (6). Parameter update equations are canonically derived and given by [e.g. 20]:

$$\pi \;=\; \frac{1}{N}\sum_{n=1}^{N}\sum_{h=1}^{H}\langle s_h\rangle_{q_n}\,, \qquad W \;=\; \Big(\sum_{n=1}^{N}\vec{y}^{\,(n)}\langle\vec{s}\rangle_{q_n}^{T}\Big)\Big(\sum_{n=1}^{N}\langle\vec{s}\vec{s}^{\,T}\rangle_{q_n}\Big)^{-1} \tag{7}$$

$$\sigma^2 \;=\; \frac{1}{ND}\sum_{n=1}^{N}\langle\|\vec{y}^{\,(n)} - W\vec{s}\|^2\rangle_{q_n} \tag{8}$$

where the $q_n$ are equal to the exact posteriors for exact EM, $q_n = p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta)$.

A standard variational EM approach for BSC would now replace these posteriors by variational distributions $q_n$. Applications of (mean-field) variational distributions as, e.g., applied by [19], entails (A) a choice which family of distributions to use; and (B) additional derivations in order to derive update equations for the introduced variational parameters. Also the derivation of sampling based approaches would require derivations. The same is not the case for the application of TVS (Alg. 2). In order to obtain a TVS learning algorithm for BSC, we do (for the update equations) just have to replace the expectation values in Eqns. 7 to 8 by (2). For the E-step, we then use the generative model description (Eqns. 6) in order to update the sets $\mathcal{K}^{(n)}$ using prior and approximate marginal distributions as described by Alg. 2.

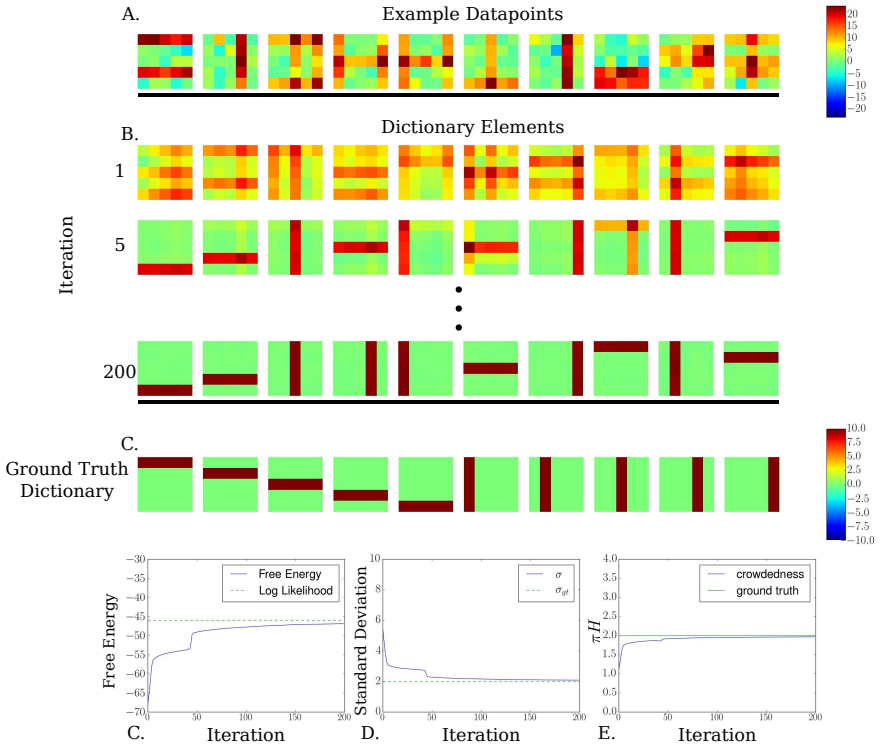


Figure 1: **Linear Bars Test**. **A.** A subset of the generated datapoints. **B.** The evolution of the dictionary over TVS iterations. Note that permutations of the dictionary elements would yield the same likelihood. **C.** The ground truth dictionary. **C.** The evolution of free-energy over TVS iterations plotted next to the exact log-likelihood. **D.** The evolution of the model standard deviation plotted next to the ground truth. **E.** The evolution of the expected number of active units $\pi H$ plotted against the ground truth.

**Artificial Data.** Firstly, we verify and study the novel approach using artificial data generated by the BSC data model using ground-truth generating parameters $\Theta_{gt}$. We use $H = 10$ latent variables, $s_h$, sampled independently

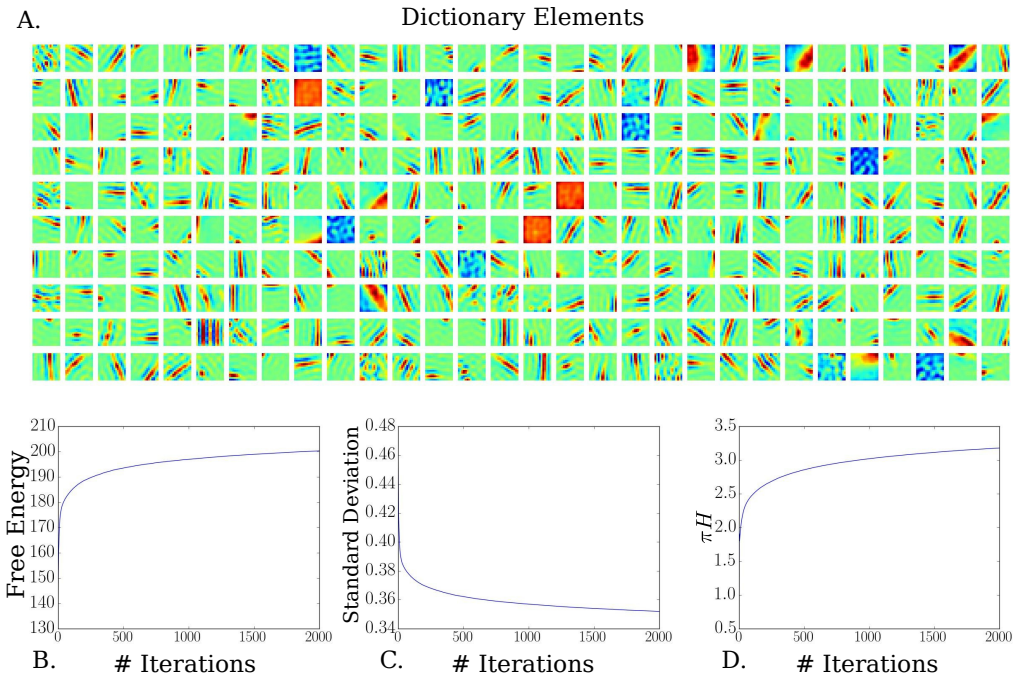


Figure 2: **Image Patches**. **A.** The dictionary at convergence. **B.** The evolution of the free-energy over TVS iterations. **C.** The evolution of the model standard deviation over TVS iterations. **E.** Evolution of the expected number of active units $\pi H$ over TVS iterations. Please enlarge for better visibility.

by a Bernoulli distribution parameterized by $\pi_{gt} = 0.2$. We set the ground truth parameters for the dictionary matrix, $W \in \mathbb{R}^{D\times H}$ to appear like vertical and horizontal bars [compare 20] when rasterized to $5 \times 5$ images, see Figure 1, with a value of 10 for a pixel that belongs to the bar and 0 for a pixel that belongs to the background. We linearly combine the latent variables with the dictionary elements to generate a $D = 25$-dimensional datapoint, $\vec{y}$ to which we add mean-free Gaussian noise with standard deviation $\sigma_{gt} = 2.0$. In this way we generate $N = 10\,000$ datapoints that form our artificial dataset.

We now use TVS for BSC to fit another instance of the BSC model to the generated data. The model is initialized with a noise parameter $\sigma$ equal to the average standard deviation of each observation in the data $\vec{y}^{(n)}$, the prior parameter is initialized as $\pi = 1/H$ were the latent variable $H = 10$ is maintained from the generating model. We initialize the columns of the dictionary matrix with the mean datapoint plus mean Gaussian samples with a standard deviation $\sigma/4$.

We train the model using the TVS algorithm for 200 TV-EM iterations maintaining the number of variational states at $|\mathcal{K}^{(n)}| = S = 64$ for all datapoints throughout the duration of the training. We use $M_q = 32$ samples drawn from the marginal distribution (only 1st approximation) and $M_p = 32$ samples drawn from the prior to vary $\mathcal{K}$ according to Alg. 2. The evolution of the parameters during training is presented in Figure 1. We were able to extract very precise estimates of the ground truth parameters of the dataset. Convergence is faster for the dictionary elements $W$ while we finally also achieve very good estimates for the noise scale $\sigma$ and prior $\pi$. We also appear to achieve a very close approximation of the exact log-likelihood using the truncated free-energy (Fig. 1), which shows that our free-energy bound is very tight for this data.

**Image Patches.** For training, we now use $N = 100\,000$ patches of size $D = 16 \times 16$ from a subset of the Van Hateren image dataset [21] that excludes images containing artificial structures. We used the same preprocessing as in [22]. We trained BSC with TVS for 2000 EM iterations and used a sampler adjustment (see Alg. 2): the first 100 iterations used $M_p = 200$ samples from the prior and $M_q = 0$ samples from the marginal distribution (only 1st approximation); from iteration 100 to iteration 200 we then linearly decreased the number of prior samples to $M_p = 0$ and increased the number of marginal samples to $M_q = 200$ (at all times $M_p + M_q = 200$). Fig. 2 shows the basis functions $W$ to converge to represent, e.g., Gabor functions [compare 20].

**Sigmoid Belief Networks.** The second example we will consider here is a typical representative of a Bayesian Network: Sigmoid Believe Networks [SBNs; 23]. While sparse coding approaches are applied to continuous (Gaussian distributed) observed variables, SBNs have binary observed and hidden variables. A further difference is that SBNs require gradients for parameter updates (partial M-steps), while parameter updates of sparse coding

models including BSC have well-known updates that fully maximize a corresponding free-energy (full M-steps). SBNs thus serve as an example complementary to BSC, and is well suited for our purposes of studying generality and effectiveness of TVS.

For simplicity, we will here consider an SBN with the same graphical model architecture as BSC: one observed and one hidden layer. The SBN generative model is then given by:

$$p(s_h) = \prod_h \pi_h^{s_h}(1-\pi_h)^{(1-s_h)}, \qquad p(\vec{y}|\vec{s}) = \prod_d g_d^{y_d}(1-g_d)^{(1-y_d)} \tag{9}$$

where $\pi_h$ parameterizes the prior distribution and where $g_d = \sigma(\sum_h W_{dh}s_h + b_d)$ is a post-linear non-linearity with Sigmoid function $\sigma$.

In general, inference for SBNs is challenging because of potentially show complex dependencies among its variables. Because of this, direct applications of standard variational approaches [e.g. 23] are challenging, and also popular recent variational methods applying reparameterization [24, 25] are not directly applicable. Also (variational) sampling approaches require additional mechanisms, e.g., the score function based approach needs to reduce the variance of estimation [3].

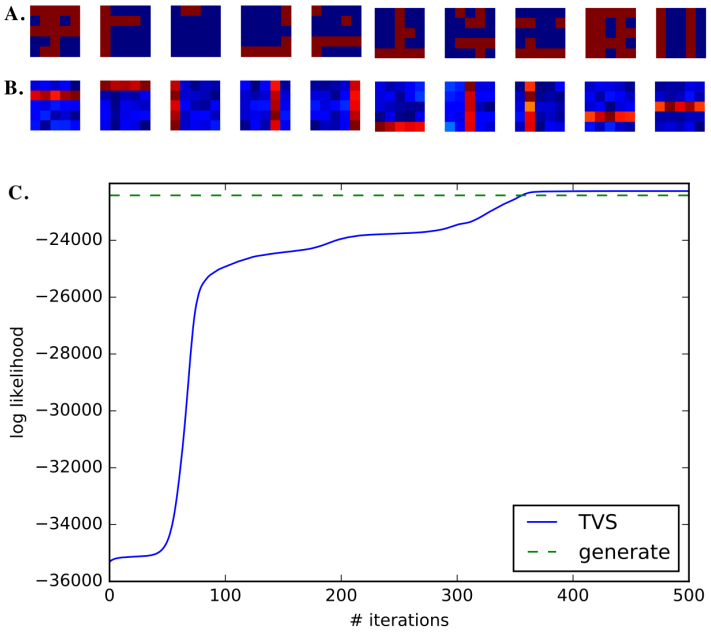


Figure 3: Application of a shallow SBN with TVS to artificial data (bars test). **A** Ten examples of the training data points. **B** Visualization of the learned weight matrix $W$ of the shallow SBN. All bars are discovered, one for each hidden unit. **C** Learning curve of the free energy. The dashed green line shows the true log likelihood of the SBN with the parameters used for generating the training data. Enlarge for better visibility.

**Artificial Data.** As for BSC, the optimization of SBNs by applying Alg. 2 does not require additional derivations. We again first use a bars test as for BSC but the linearly superimposed bars now go through the sigmoid function and produce binary representations, i.e., generation according to (9). We optimized an SBN with $H = 10$ hidden units on $N = 2000$ data points of such a bars test. For Alg. 2 we used $|\mathcal{K}^{(n)}| = S = 50$ and very few samples for variation were found sufficient ($M_q = M_p = 5$). Results are shown in Fig. 3. The free energy (3) converges to even somewhat higher values than the (here still computable) ground-truth likelihood because of the limited size of training data.

**Binarized MNIST.** Fianlly we apply SBNs to the Binarized MNIST dataset (downloaded from [28], converted as in [29]). We used Alg. 2 with $|\mathcal{K}^{(n)}| = S = 50$, $M_p = 10$, $M_q = 20$ (no sampler adjustment) and truncated marginal distributions approximated using an MLP with one hidden layer of 500 hidden units and *tanh* activation (2nd Approximation). Tab. 1 compares SBNs optimized by TVS with other models and optimization approaches.

## 5 Conclusion

The TVS approach studied here is different from previous approaches [9, 10, 11, 6, 12] as it does *not* rely on a parametric form of a variational distribution which is then, e.g., sampled from to estimate parameter

| model | $H$ | approx. log-LL |
|---|---|---|
| SBN (TVS) | 100 | -121.91 |
| SBN (TVS) | 200 | -111.23 |
| SBN (Gibbs)$^*$ | 200 | -94.3 |
| SBN (VB)$^*$ | 200 | -117.0 |
| SBN (NVIL)$^\diamond$ | 200 | -113.1 |
| SBN (WS)$^\dagger$ | 200 | -120.7 |
| SBN (RWS)$^\dagger$ | 200 | -103.1 |
| SBN (AIR)$^\ddagger$ | 200 | -100.9 |

Table 1: Comparison of different models with two layers and different numbers of latents $H$ on binarized MNIST. ($*$) taken from [2], ($\diamond$) from [3], ($\dagger$) from [26], ($\ddagger$) from [27]. For TVS the final free energy on the test set can directly be computed by iterating the TV-E-step. The right column reports these values as the estimated test log-likelihood (log-LL). Hence, for TVS the log-LL values are a lower bound estimation while results by [26, 27] are from Monte Carlo estimations (and not necessarily lower bounds). For SBNs with 200 hidden units, we observed that TVS slightly outperforms NVIL, and its performance is comparable to RWS. The results of [2] can not directly serveas a comparison of variational approaches (additional knowledge in the form of sparse priors were used).

updates. In contrast, for TVS, the drawn samples *themselves* define the variational distribution and act as its variational parameters. Changing the used samples changes the variational distribution. TVS is thus by definition directly coupling sampling and variational EM which in conjunction with its 'black box' applicability is the main contribution of this study. One benefit of the tight coupling seems to be that none of the diverse variance reduction techniques (which were central to BBVI or NVIL) are required. TVS can thus be considered as the most directly applicable 'black box' approach. We have here shown a proof of concept. More advanced models and further algorithmic improvments will be the subject of future studies.

## References


[1] Goodfellow, I., Courville, A.C., Bengio, Y.: Large-scale feature learning with spike-and-slab sparse coding. In: ICML. (2012)

[2] Gan, Z., Henao, R., Carlson, D., Carin, L.: Learning Deep Sigmoid Belief Networks with Data Augmentation. In: AISTATS. (2015)

[3] Mnih, A., Gregor, K.: Neural variational inference and learning in belief networks. In: ICML. (2014)

[4] Sheikh, A.S., Lücke, J.: Select-and-sample for spike-and-slab sparse coding. In: Advances in Neural Information Processing Systems (NIPS). Volume 29. (2016) 3927–3935

[5] Jordan, M., Ghahramani, Z., Jaakkola, T., Saul, L.: An introduction to variational methods for graphical models. Machine Learning **37** (1999) 183–233

[6] Hoffman, M., Blei, D.: Stochastic structured variational inference. In: AISTATS. (2015) 361–369

[7] Salimans, T., Kingma, D.P., Welling, M., et al.: Markov chain monte carlo and variational inference: Bridging the gap. In: International Conference on Machine Learning. (2015) 1218–1226

[8] Hernandez-Lobato, J., Li, Y., Rowland, M., Bui, T., Hernandez-Lobato, D., Turner, R.: Black-box alpha divergence minimization. In: International Conference on Machine Learning. (2016) 1511–1520

[9] Ranganath, R., Gerrish, S., Blei, D.M.: Black box variational inference. In: AISTATS. (2014) 814–822

[10] Tran, D., Blei, D., Airoldi, E.M.: Copula variational inference. In: Advances in Neural Information Processing Systems. (2015) 3564–3572

[11] Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows. International Conference on Machine Learning (2015)

[12] Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., Blei, D.M.: Automatic differentiation variational inference. CoRR **abs/1603.00788** (2016)

[13] Lücke, J., Eggert, J.: Expectation truncation and the benefits of preselection in training generative models. Journal of Machine Learning Research **11** (2010) 2855–900

[14] Forster, D., Lücke, J.: Can clustering scale sublinearly with its clusters? A variational EM acceleration of GMMs and k-means. In: arXiv:1711.03431, AISTATS to appear. (2018)

[15] Lücke, J.: Truncated variational expectation maximization. arXiv preprint, arXiv:1610.03113 (2016)

[16] Shelton, J.A., Gasthaus, J., Dai, Z., Lücke, J., Gretton, A.: GP-select: Accelerating EM using adaptive subspace preselection. Neural Computation **29**(8) (2017) 2177–2202

[17] Gu, S., Ghahramani, Z., Turner, R.E.: Neural adaptive sequential Monte Carlo. In: NIPS. (2015)

[18] Mnih, A., Rezende, D.J.: Variational inference for monte carlo objectives. In: ICML. (2016)

[19] Haft, M., Hofman, R., Tresp, V.: Generative binary codes. Pattern Anal Appl **6** (2004) 269–84

[20] Henniges, M., Puertas, G., Bornschein, J., Eggert, J., Lücke, J.: Binary sparse coding. In: Proceedings LVA/ICA. LNCS 6365, Springer (2010) 450–57

[21] van Hateren, J.H., van der Schaaf, A.: Independent component filters of natural images compared with simple cells in primary visual cortex. Proceedings of the Royal Society of London B **265** (1998) 359–66

[22] Exarchakis, G., Lücke, J.: Discrete sparse coding. Neural Computation **29** (2017) 2979–3013

[23] Saul, L.K., Jaakkola, T., Jordan, M.I.: Mean field theory for sigmoid belief networks. Journal of artificial intelligence research **4**(1) (1996) 61–76

[24] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR. (2014)

[25] Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: ICML. (2014)

[26] Bornschein, J., Bengio, Y.: Reweighted wake-sleep. In: ICLR. (2015)

[27] Hjelm, R.D., Cho, K., Chung, J., Salakhutdinov, R., Calhoun, V., Jojic, N.: Iterative refinement of the approximate posterior for directed belief networks. In: NIPS. (2016)

[28] Larochelle, H.: Binarized MNIST dataset. `http://www.cs.toronto.edu/~larocheh/` (2011)

[29] Murray, I., Salakhutdinov, R.: Evaluating probabilities under high-dimensional latent variable models. In: NIPS. (2008)