Select and Sample — A Model of Efficient Neural Inference and Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

An increasing number of experimental studies indicate that perception encodes a posterior probability distribution over possible causes of sensory stimuli, which is used to act close to optimally in the environment. One outstanding difficulty with this hypothesis is that the exact posterior will in general be too complex to be represented directly, and thus neurons will have to represent an approximation of this distribution. Two influential proposals for how neural populations could achieve an efficient posterior representation are 1) that neural activity represents samples of the underlying distribution, or 2) that they represent a parametric representation of a variational approximation of the posterior. Here we show that these approaches can be combined for an inference scheme that retains the advantages of both: it is able to represent multiple modes and arbitrary correlations, a feature of sampling methods, and it reduces the repre- sented space to regions of high probability mass, a strength of variational approximations. Neurally, the combined method can be interpreted as a feed-forward preselection of the relevant state space, followed by a neural dynamics implementing Markov Chain Monte Carlo (MCMC) to approxi- mate the posterior over the relevant states. We demonstrate the effectiveness and efficiency of this approach on a sparse coding model. In numerical experiments on artificial data and image patches, we compare the performance of the algorithms to the performance of exact EM, variational state space selection alone, MCMC alone, and the combined select-and-sample approach. The select-and-sample ap- proach integrates the advantages of the sampling and variational approximations, and forms a robust, neurally plausible, and very efficient model of processing and learning in cortical networks. For sparse coding we find that it enables applica- tions easily exceeding a thousand observed and a thousand hidden dimensions.

1 Introduction

According to the statistical approach to perception that has been increasingly influential in the recent past, our brain would represent not only the most likely interpretation of a stimulus, but also the uncertainty associated with it. Ideally, the brain would represent the full posterior distribution over all possible interpretations of the stimulus, which is statistically optimal for inference and learning [1, 2, 3]. An increasing number of psychophysical and electrophysiological studies [4, 5, 6, 7, 8, 9] seem to confirm this hypothesis.

An outstanding difficulty with this approach is that, in general, the full posterior distribution is very complex, as it may be very correlated (because of explaining away effects), multimodal (multiple possible interpretations), and highly multi-dimensional. One approach in dealing with this problem in neural circuits is to assume that neuronal activity represent the parameters of a variational approximation of the real posterior [10, 11]. While this approach allows to instantaneously represent an approximate version of the full posterior, the number of neurons still explodes with the number of variables. For example, approximating the posterior with a Gaussian distribution requires N^2 parameters to represent the covariance matrix over N variables. Another approach is to identify neurons with variables, and interpret neural activity as samples from their posterior [12, 13, 3]. This interpretation is consistent with a range of experimental observations, including neural variability (which would result from the uncertainty in the posterior) and spontaneous activity (corresponding to samples from the prior in the absence of a stimulus [3, 9]. The advantage of a sampling-based representation is that the number of neurons scales linearly with the number of variables, while retaining the ability to represent arbitrarily complex posteriors given enough samples. However, an important issue with this approach is the amount of time needed to collect a sufficient number of samples in order to form a complete enough representation of the posterior. Modeling studies have shown that a small number of samples are sufficient to perform well on low-dimensional tasks (intuitively, this is because taking a low-dimensional marginal of the posterior accumulates samples over all dimensions) [14, 15]. However, most sensory data is inherently very high-dimensional. For instance, to faithfully represent visual scenes containing potentially many objects and object parts, one requires a high-dimensional latent space to represent the high number of potential causes. For high dimensionalities, sampling approaches usually suffer from long burn-in times and require in general a high number of samples to represent non-trivial posteriors.

In this paper we propose a novel approach to approximate inference and learning that addresses the drawbacks of sampling as model for neural processing while maintaining its richness in posterior representation and its neural plausibility. We show that sampling can be combined with a pre-selection of candidate units. Such a selection connects sampling to very influencial models of neural processing that emphasise feed-forward processing ([16, 17] and many more), and is consistent with the popular view of neural processing and learning as an interplay between feed-forward and recurrent stages of processing [18, 19, 20, 12]. Our combined approach emerges naturally by interpreting feed-forward selection and sampling as approximations to exact inference in a probabilistic framework for perception.

2 A Select and Sample Approach to Approximate Inference

Inference and learning in neural circuits can be regarded as the task of inferring the true hidden causes of a stimulus. An example is inferring the objects in a visual scene based on the image projected on the retina. We will refer to the sensory stimulus (the image) as a *data point*, $\vec{y} = (y_1, \ldots, y_D)$, and we will refer to the hidden causes (the objects) as $\vec{s} = (s_1, \ldots, s_H)$ with s_h denoting *hidden variable* or *hidden unit* h. The data distribution can then be modelled by a generative data model: $p(\vec{y}\,|\,\Theta) = \sum_{\vec{s}} p(\vec{y}\,|\,\vec{s},\Theta)\,p(\vec{s}\,|\,\Theta)$ with Θ denoting the parameters of the model[1]. If we assume that the data distribution can be optimally modelled by the generative distribution for optimal parameter Θ^*, then the posterior probability $p(\vec{s}\,|\,\vec{y},\Theta^*)$ represents optimal inference given a data point $\vec{y}$. The parameters Θ^* given a set of N data points $Y = \{\vec{y}_1, \ldots, \vec{y}_N\}$ are given by the maximum likelihood parameters $\Theta^* = \mathrm{argmax}_\Theta\{p(\vec{y}\,|\,\Theta)\}$.

A standard procedure to find the maximum likelihood solution is expectation maximization (EM). EM iteratively optimize a lower bound of the data likelihood by inferring the posterior distribution over hidden variables given the current parameters (the E-step), and then adjusting the parameters to maximize the likelihood of the data averaged over this posterior (the M-step). The M-step updates typically depend only on a small number of expectation values of the posterior as given by

$$\langle g(\vec{s})\rangle_{p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)} \;=\; \textstyle\sum_{\vec{s}}\, p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)\, g(\vec{s})\,, \tag{1}$$

where $g(\vec{s})$ is usually an elementary function of the hidden variables (e.g., $g(\vec{s}) = \vec{s}$ or $g(\vec{s}) = \vec{s}\vec{s}^T$ in the case of standard sparse coding). The sum is replaced by an integral for continuous latents. The computation of expectation values (1) is the computationally demanding part of EM optimization for any non-trivial generative model. Their exact computation is often intractable and many well-known algorithms [21, 22] have to rely on approximations.

The EM iterations can be associated to neural processing by the assumption that neural activity represents the posterior over hidden variables (E-step), and relating the M-step equations long-term

[1]In the case of continuous variables the sum is replaced by an integral. For a hierarchical model, the prior distribution $p(\vec{s}\,|\,\Theta)$ may be subdivided hierarchically into different sets of variables.

changes in synaptic weights due to synaptic plasticity. Here we will consider two prominent models of neural processing on the ground of approximations to the expectation values (1) and show how they can be combined.

Selection. Feed-forward processing has frequently been discussed as an important component of neural processing [16, 23, 17, 24]. One perspective on this early componenet of neural activity is as a pre-selection of candidate units or hypotheses for a given sensory stimulus ([20, 25, 18] and many more), with the goal of reducing the computational demand of an otherwise too complex computation In the context of probabilistic approaches, it has recently been shown that preselection can be formulated as a variational approximation to exact inference [26]. The variational distribution in this case is given by a truncated sum over possible hidden states:

$$p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta) \approx q_n(\vec{s};\Theta) = \frac{p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta)}{\displaystyle\sum_{\vec{s}\,'\in\mathcal{K}_n} p(\vec{s}\,'\,|\,\vec{y}^{\,(n)},\Theta)}\,\delta(\vec{s}\in\mathcal{K}_n) = \frac{p(\vec{s},\vec{y}^{\,(n)}\,|\,\Theta)}{\displaystyle\sum_{\vec{s}\,'\in\mathcal{K}_n} p(\vec{s}\,',\vec{y}^{\,(n)}\,|\,\Theta)}\,\delta(\vec{s}\in\mathcal{K}_n) \quad (2)$$

where $\delta(\vec{s}\in\mathcal{K}_n) = 1$ if $\vec{s}\in\mathcal{K}_n$ and zero otherwise. The subset $\mathcal{K}_n$ represents the preselected latent states. Given a data point $\vec{y}^{\,(n)}$, Eqn. 2 results in good approximations to the posterior if $\mathcal{K}_n$ contains most posterior mass. Since for many applications the posterior mass is concentrated in small volumes of the state space, the approximation quality can stay high even for relatively small sets $\mathcal{K}_n$. This approximation can be used to compute efficiently the expectation values needed in the M-step (1):

$$\langle g(\vec{s})\rangle_{p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta)} \approx \langle g(\vec{s})\rangle_{q_n(\vec{s};\Theta)} = \frac{\sum_{\vec{s}\in\mathcal{K}_n} p(\vec{s},\vec{y}^{\,(n)}\,|\,\Theta)\,g(\vec{s})}{\sum_{\vec{s}\,'\in\mathcal{K}_n} p(\vec{s}\,',\vec{y}^{\,(n)}\,|\,\Theta)}\,. \quad (3)$$

Eqn. 3 represents a reduction in required computational resources as it involves only summations (or integrations) over the smaller state space $\mathcal{K}_n$. The bottleneck here is that the set $\mathcal{K}_n$ needs to be selected prior to the computation of expectation values, and the final improvement in efficiency relies on such selections being efficiently computable. For sparse coding models, for instance, we can exploit that the posterior mass lie close to low dimensional subspaces to define the sets $\mathcal{K}_n$ [26, 27]. The selection functions compute those candidate hidden units s_h that are most likely to have contributed to a data point $\vec{y}^{\,(n)}$. $\mathcal{K}_n$ can then be defined by:

$$\mathcal{K}_n = \{\vec{s}\,|\,\text{for all } h\notin\mathcal{I}:\ s_h = 0\}\,, \quad (4)$$

where $\mathcal{I}$ contains the H' indices h with the highest values of a selection function $\mathcal{S}_h(\vec{y},\Theta)$ (compare Fig. 1). Appropriate selection functions $\mathcal{S}_h(\vec{y},\Theta)$, e.g. for sparse coding models, can be found by deriving efficiently computable upper-bounds for probabilities $p(s_h = 1\,|\,\vec{y}^{\,(n)},\Theta)$ [26, 27] or via deterministic relations $\vec{s} = f(\vec{y},\Theta)$ in the limit of no data noise [26, 28]. Often the precise form of the selection function has limited influence on the final approximation accuracy because a) its values are not used for the approximation (3) itself and b) the size of sets $\mathcal{K}_n$ can often be chosen generously such that they easily contain the regions with large posterior mass. The larger $\mathcal{K}_n$ the less precise the selection has to be. For $\mathcal{K}_n$ equal to the entire state space, no selection is required and the approximations (2) and (3) fall back to the case of exact inference.

Sampling. An alternative way to approximate the expectation values in eq. 1 is by sampling from the posterior distribution, and using the samples to compute the average:

$$\langle g(\vec{s})\rangle_{p(\vec{s}\,|\,\vec{y}^{\,(n)},\Theta)} \approx \tfrac{1}{M}\textstyle\sum_{m=1}^{M} g(\vec{s}^{(m)}) \text{ with } \vec{s}^{(m)} \sim p(\vec{s}\,|\,\vec{y},\Theta). \quad (5)$$

The challenging aspect of this approach is to efficiently draw samples from the posterior. In large dimensional sample space, this is mostly done by Markov Chain Monte Carlo (MCMC). This class of methods draws samples from the posterior distribution such that each subsequent sample is drawn relative to the current state, and the resulting sequence of samples form a Markov chain. A new sample is accepted with a probability of $\max(1, \frac{p(\vec{s}^{\,\text{new}})}{p(\vec{s}^{\,\text{current}})})$. In the limit of a large number of samples, MCMC methods are theoretically able to represent any probability distribution. However, the number of samples required in large dimensional space can be very large (Fig. 1A, sampling).

Select and Sample. While preselection and sampling seem to be very different in nature, their formulations as approximations to expectation values (3) allow for a straight-forward combination of both approaches: Given a data point, $\vec{y}^{(n)}$, we first approximate the expectation value (3) using

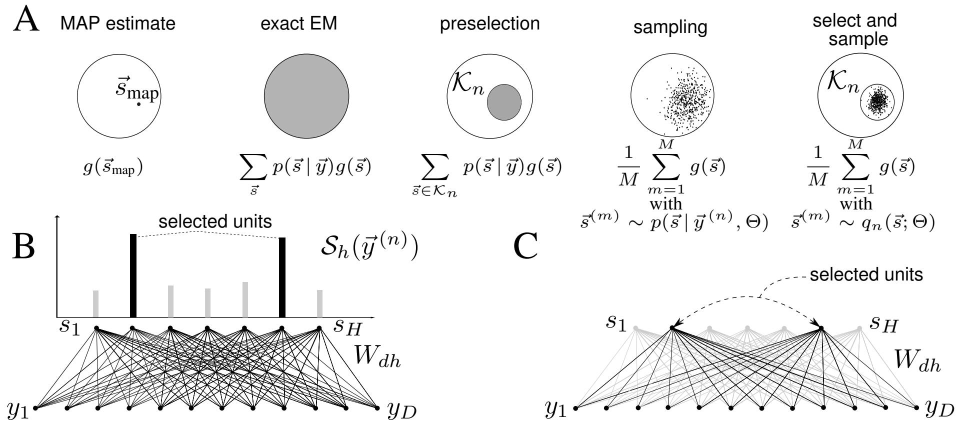

Figure 1: **A** Simplified illustration of the posterior mass and the respective regions each approximation approach uses to compute the expectation values. **B** Graphical model showing full connections W_{dh} between the observed variable $\vec{y}^{(n)}$ and hidden variable $\vec{s}$ and how H' hidden dimensions are selected from $\vec{s}_H$ to form a given set K_n and W_{dh} is affected accordingly. **C** Illustrates how sampling draws samples from this reduced set (corresponds also to the right-most posterior mass depicted in **A**).

the variational distribution $q_n(\vec{s};\Theta)$ as defined by preselection (3). Second, we approximate the expectations w.r.t. $q_n(\vec{s};\Theta)$ using sampling. The combined approach is thus given by:

$$\langle g(\vec{s})\rangle_{p(\vec{s}\,|\,\vec{y}^{(n)},\Theta)} \;\approx\; \langle g(\vec{s})\rangle_{q_n(\vec{s};\Theta)} \;\approx\; \tfrac{1}{M}\textstyle\sum_{m=1}^{M} g(\vec{s}^{(m)}) \quad \text{with} \quad \vec{s}^{(m)} \sim q_n(\vec{s};\Theta), \tag{6}$$

where $\vec{s}^{(1)}$ to $\vec{s}^{(M)}$ denote samples from the truncated distribution q_n. Instead of drawing from a distribution over the entire state space, approximation (6) requires only samples from a potentially very small subspace $\mathcal{K}_n$ (Fig. 1). In the subspace $\mathcal{K}_n$, most of the original probability mass is concentrated in a smaller volume, and thus we expect MCMC algorithms to perform more efficiently, as they need to explore a smaller volume, shortening burn-in times, and reducing the number of samples needed to trace the distribution. Compared to selection alone, the select-and-sample approach will represent an increase in efficiency as soon as the number of samples required for a good approximation is less then the number of states in $\mathcal{K}_n$. In the following, we will systematically investigate the computational efficiency of the select and sample approach in comparison with selection and sampling alone using concrete examples of generative models with real-world scales.

3 Sparse Coding: An Example Application

To explore the efficiency and performance properties of the select and sample approach as well as to study its biological plausibility, we apply it to a sparse coding model of images. The choice of a sparse coding model has numerous advantages. First, it is a non-trivial model that has been extremely well-studied in machine learning research, and for which efficient algorithms exist (e.g., [22, 29]). Second, it has become a standard (albeits somewhat simplistic) model of the organization of receptive fields in primary visual cortex [21, 30, 31]. Here we consider a discrete variant of this model known as Binary Sparse Coding (BSC; [28, 26], also compare [32]), which has binary hidden variables but otherwise the same features as standard sparse coding versions. The generative model for BSC is expressed by

$$p(\vec{s}|\pi) = \textstyle\prod_{h=1}^{H} \pi^{s_h}\big(1-\pi\big)^{1-s_h}, \qquad p(\vec{y}|\vec{s},W,\sigma) = \mathcal{N}(\vec{y};W\vec{s},\sigma^2\mathbb{1}), \tag{7}$$

where $W \in \mathbb{R}^{D\times H}$ denotes the basis vectors and π parameterizes the sparsity ($\vec{s}$ and $\vec{y}$ as above). The M-step updates of the BSC learning algorithm (see e.g. [26]) are given by:

$$W^{\text{new}} = \big(\textstyle\sum_{n=1}^{N} \vec{y}^{(n)}\,\langle\vec{s}\rangle_{q_n}^{T}\big)\,\big(\sum_{n=1}^{N}\left\langle\vec{s}\,\vec{s}^{\,T}\right\rangle_{q_n}\big)^{-1}, \tag{8}$$

$$(\sigma^2)^{\text{new}} = \tfrac{1}{ND}\textstyle\sum_n \left\langle\left|\left|\vec{y}^{(n)} - W\,\vec{s}\right|\right|^2\right\rangle_{q_n}, \; \pi^{\text{new}} = \tfrac{1}{N}\sum_n |\langle\vec{s}\rangle_{q_n}|, \;\text{ where } |\vec{x}| = \tfrac{1}{H}\sum_h x_h. \tag{9}$$

The only expectation values needed for the M-step are thus $\langle\vec{s}\rangle_{q_n}$ and $\left\langle\vec{s}\vec{s}^{\,T}\right\rangle_{q_n}$.

We are going to compare inference and learning using different methods:

BSC$^{\text{exact}}$. An EM algorithm without approximations is obtained if we use the exact posterior for the expectations: $q_n = p(\vec{s}\,|\,\vec{y}^{\,(n)}, \Theta)$. We will refer to this exact algorithm as $\mathrm{BSC}^{\mathrm{exact}}$. While directly computable the expectation values for $\mathrm{BSC}^{\mathrm{exact}}$ require sums over the entire state space, i.e., over 2^H terms. For large numbers of latent dimensions $\mathrm{BSC}^{\mathrm{exact}}$ is thus intractable.

BSC$^{\text{select}}$. An algorithm that more efficiently scales with the number of hidden dimensions is obtained by applying preselection. For the BSC model we use q_n as given in (3) and $\mathcal{K}_n = \{\vec{s}\,|\,(\text{for all } h \notin \mathcal{I}: \; s_h = 0) \text{ or } \sum_h s_h = 1\}$. Note that in addition to states as in (4) we include all states with one non-zero unit (all singletons). Including them avoids EM iterations in the initial phases of learning that leave some basis functions unmodified (see [26]). As selection function $\mathcal{S}_h(\vec{y}^{\,(n)})$ to define $\mathcal{K}_n$ we use:

$$\mathcal{S}_h(\vec{y}^{\,(n)}) = (\vec{W}_h^{\mathrm{T}} \,/\, ||\vec{W}_h||)\, \vec{y}^{\,(n)}, \qquad \text{with} \;\; ||\vec{W}_h|| = \sqrt{\textstyle\sum_{d=1}^{D} (W_{dh})^2}\,. \tag{10}$$

A large value of $\mathcal{S}_h(\vec{y}^{\,(n)})$ strongly indicates that $\vec{y}^{\,(n)}$ contains the basis function $\vec{W}_h$ as a component (see Fig. 1C). Note that (10) can be related to a deterministic ICA-like selection of a hidden state $\vec{s}^{(n)}$ in the limit case of no noise (compare [26]). Further restrictions of the state space are possible but require modified M-step equations (see [26, 28]), which will not be considered here.

BSC$^{\text{sample}}$. An alternative non-deterministic approach can be derived using Gibbs sampling. Gibbs sampling is an MCMC algorithm which systematically explores the sample space, considering each dimension individually, conditioning the acceptance of a new sample based on values of the remaining samples. In other words, the transition probability from the current sample to a new candidate sample is given by $p(s_h^{\mathrm{new}}|\vec{s}_{\backslash h}^{\,\mathrm{current}})$. In our case of a binary sample space, this equates to selecting one random axis $h \in \{1, \ldots, H\}$ and toggling its bit value (thereby changing the binary state in that dimension), leaving the remaining axes unchanged. Specifically, the posterior probability computed for each candidate sample is expressed by:

$$p(s_h = 1\,|\,\vec{s}_{\backslash h}, \vec{y}) = \frac{p(s_h = 1, \vec{s}_{\backslash h}, \vec{y})^{\beta}}{p(s_h = 0, \vec{s}_{\backslash h}, \vec{y})^{\beta} + p(s_h = 1, \vec{s}_{\backslash h}, \vec{y})^{\beta}}, \tag{11}$$

where β is used to smooth out the posterior distribution. To ensure an appropriate mixing behavior over a wide range of σ (note that σ is a model parameter that changes with learning), we define $\beta = \frac{T}{\sigma^2}$, where T is a temperature parameter that is set manually and is selected such that good mixing behavior of the MCMC chains is acheived. The samples drawn by applying the described procedure can then be used to approximate the expectation values in (9) to (8) using (5).

BSC$^{\text{s+s}}$. The EM learning algorithm given by combining selection and sampling is obtained by applying (6). First note that inserting the BSC generative model into (2) results in:

$$q_n(\vec{s};\,\Theta) \;\; = \;\; \frac{\mathcal{N}(\vec{y}; W\vec{s}, \sigma^2 \mathbb{1})\, \mathrm{Bernoulli}_{\mathcal{K}_n}(\vec{s}; \pi)}{\sum_{\vec{s}\,' \in \mathcal{K}_n} \mathcal{N}(\vec{y}; W\vec{s}\,', \sigma^2 \mathbb{1})\, \mathrm{Bernoulli}_{\mathcal{K}_n}(\vec{s}\,'; \pi)}\, \delta(\vec{s} \in \mathcal{K}_n) \tag{12}$$

where $\mathrm{Bernoulli}_{\mathcal{K}_n}(\vec{s}; \pi) = \prod_{h \in \mathcal{I}} \pi^{s_h}\, (1-\pi)^{1-s_h}$. The remainder of the Bernoulli distribution cancels out. If we define $\tilde{\vec{s}}$ to be the binary vector consisting of all entries of $\vec{s}$ of the selected dimensions, and if $\tilde{W} \in \mathbb{R}^{D \times H'}$ contains all basis functions of the selected units, we observe that the distribution is equal to the posterior w.r.t. a BSC generative model with H' instead of H hidden dimensions:

$$p(\tilde{\vec{s}}\,|\,\vec{y}, \Theta) \;\; = \;\; \frac{\mathcal{N}(\vec{y}; \tilde{W}\tilde{\vec{s}}, \sigma^2 \mathbb{1}_{H'})\, \mathrm{Bernoulli}(\tilde{\vec{s}}; \pi)}{\sum_{\tilde{\vec{s}}'} \mathcal{N}(\vec{y}; \tilde{W}\tilde{\vec{s}}', \sigma^2 \mathbb{1}_{H'})\, \mathrm{Bernoulli}(\tilde{\vec{s}}'; \pi)} \;\; = \;\; p(\tilde{\vec{s}}\,|\,\vec{y}, \Theta)$$

Instead of drawing samples from $q_n(\vec{s};\,\Theta)$ we can thus draw samples from the exact posterior w.r.t. the BSC generative model with H' dimensions. The sampling procedure for $\mathrm{BSC}^{\mathrm{sample}}$can thus be applied simply by ignoring the non-selected dimensions and their associated parameters. For different data points different latent dimensions will be selected such that averaging over data points can update all model parameters. For selection we again use selection functions (10) and again define $\mathcal{K}_n$ similar to (4):

$$\mathcal{K}_n = \{\vec{s}\,|\,\text{for all } h \notin \mathcal{I}: \; s_h = 0\}\,, \tag{13}$$

where $\mathcal{I}$ contains the H'-2 indices h with the highest values of a selection function $\mathcal{S}_h(\vec{y}, \Theta)$ and two randomly selected dimensions (drawn from a uniform distribution over all non-selected dimensions).

The two randomly selected dimensions fulfill the same purpose as the inclusion of singleton states for BSC$^{\text{select}}$. Preselection and Gibbs sampling on the selected dimensions define an approximation to the required expectation values (3) and result in an EM algorithm referred to as BSC$^{\text{s+s}}$.

Complexity. Collecting the number of operations necessary to compute the expectation values for all four BSC cases, we arrive at

$$\mathcal{O}\big(NS(\underbrace{D}_{p(\vec{s},\vec{y})} + \underbrace{1}_{\langle\vec{s}\rangle} + \underbrace{H}_{\langle\vec{s}\vec{s}^T\rangle})\big) \tag{14}$$

where S denotes the number of hidden states that contribute to the calculation of the expectation values. For the approaches with preselection (BSC$^{\text{select}}$, BSC$^{\text{s+s}}$), all the calculations of the expectation values can be performed on the reduced latent space; therefore the H is replaced by H'. For BSC$^{\text{exact}}$this number scales exponentially in H: $S^{\text{exact}} = 2^H$, and in in the BSC$^{\text{select}}$case, it scales exponentially in the number of preselected hidden variables: $S^{select} = 2^{H'}$. However, for the sampling based approaches (BSC$^{\text{sample}}$and BSC$^{\text{s+s}}$), the number S directly corresponds to the number of samples to be evaluated and is obtained empirically. As we will show later, $S^{\text{s+s}} = 200 \times H'$ is a reasonable choice for the interval of H' that we investigate in this paper ($1 \leq H' \leq 40$).

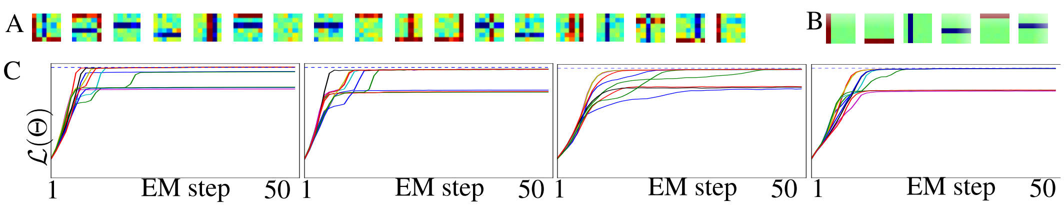

Figure 2: Experiments using artificial bars data with $H = 12$, $D = 6 \times 6$. Dotted line indicates the ground truth log-likelihood value. **A** Random selection of the $N = 2000$ training data points $\vec{y}^{(n)}$ **B** Learned basis functions W_{dh} after an successful training run **C** Development of the log-likelihood over a period of 50 EM steps for all 4 investigated algorithms.

4 Numerical Experiments

We compare the select and sample approach with selection and sampling applied individually on different data sets. For all experiments using the two sampling approaches, we draw 20 independent chains that are initialized at random states in order to increase the mixing of the samples. Also, of the samples drawn per chain, $\frac{1}{3}$ were used to as burn-in samples, and $\frac{2}{3}$ were retained samples.

Artificial data. Before we apply the select-and-sample approach to large scale learning on image patches, we investigate its convergence properties on artificial data sets where ground truth is available. As the following experiments were run on a small scale problem, all four algorithms (BSC$^{\text{exact}}$, BSC$^{\text{select}}$, BSC$^{\text{sample}}$and BSC$^{\text{s+s}}$) can be applied and compared. Furthermore, for all the experiments we computed the exact likelihood for each EM step.

Data for these experiments consisted of bars data generated by creating $H = 12$ basis functions $\vec{W}_h^{\text{gt}}$ in the form of horizontal and vertical bars on a $D = 6 \times 6 = 36$ pixel grid. Each bar was randomly assigned to be either positive ($W_{dh}^{\text{gt}} \in \{0.0, 10.0\}$) or negative ($W_{h'd}^{gt} \in \{-10.0, 0.0\}$). $N = 2000$ data points $\vec{y}^{(n)}$ were generated by linear combining these basis functions (see e.g., [33]). Using a sparseness value of $\pi_{\text{gt}} = \frac{2}{H}$ resulted in, on average, two active bars per data point. According to the model, we added Gaussian noise ($\sigma_{\text{gt}} = 2.0$) to the data (Fig. 3 **A**). We applied algorithms to the same dataset and monitored the exact likelihood while these algorithms converged over a period of 50 EM steps (Fig. 3 **C**). When running models using preselection (BSC$^{\text{select}}$and BSC$^{\text{s+s}}$), we set H' to 6, effectively halving the number of hidden units participating in the calculation of the expectation values. For BSC$^{\text{sample}}$we used the Gibbs sampler to draw $S^{\text{sample}} = 200 \times H = 2400$ samples from the posterior $p(\vec{s}\,|\,\vec{y}^{(n)})$ of each data point. For BSC$^{\text{s+s}}$we drew $S^{\text{s+s}} = 200 * H' = 1200$ samples from the posterior of each data point. To ensure an appropriate mixing behavior, we furthermore set the annealing temperature to $T = 50$. In all these experiments we initialized the basis functions to the data mean plus Gaussian noise ($\vec{W}_h^{\text{init}} = \frac{1}{N}\sum_{n=1}^{N} \vec{y}^{(n)} + \eta_{\sigma=1}$), the prior probability to $\pi_{\text{init}} = \frac{1}{H}$ and the data noise to the variance of the data ($\sigma^2_{\text{init}} = \sum_{d=1}^{D} \left[\sum_{n=1}^{N} (y_d^{(n)})^2 - (\sum_{n=1}^{N} y_d^{(n)})^2 \right]$) All algorithms recover the correct set of bases functions in $> 50\%$ of the trials, and recover the sparseness prior π and the data noise σ with high accuracy. Comparing the computational costs of algorithms shows the benefits of preselection already for this small scale problem: While $\text{BSC}^{\text{exact}}$evaluates the expectation values using the full set of $2^H = 4096$ hidden states, $\text{BSC}^{\text{select}}$only considers $2^{H'} + (H - H') = 70$ states. The pure sampling based approaches performs 2400 evaluations while $\text{BSC}^{\text{s+s}}$requires 1200 evaluations.

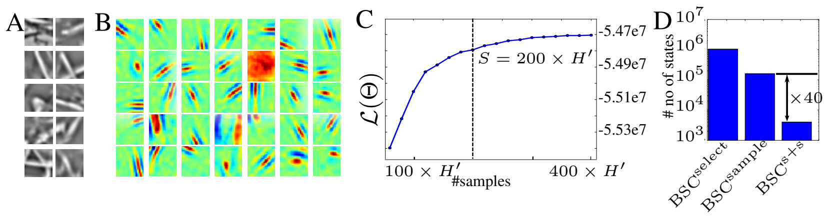

Figure 3: Experiments on $D = 26 \times 26$ image patches: **A** a set of 10 randomly chosen image patches after DoG preprocessing. **B** a selection of the learned basis functions for $H = 800, H' = 20$ and number of samples set to $200 \times H'$. See the Supp. material for the full set of basis functions. **C** the final approximate log-likelihood then running the $\text{BSC}^{\text{s+s}}$for various number of samples drawn per datapoint. **D** Assuming $H = 800$ and $H' = 20$ the number of hidden states to be evaluated are shown.

Image patches. To demonstrate the applicability of the select-and-sample approach to larger scale problems, we tested our approach $\text{BSC}^{\text{s+s}}$on the most prominent application domain of sparse coding: image patches. We extracted $N = 40,000$ patches of size $D = 26 \times 26 = 676$ pixels from the van Hateren image database[30] [2]. The patches were preprocessed using a Difference of Gaussians (DoG) filter which approximates the sensitivity of center-on and center-off neurons found in the early stages of the mammalian visual processing. Filter parameters where chosen as in [34, 27]. We for the following experiments ran 100 EM iterations to ensure proper convergence. The annealing temperature was set to $T = 20$.

We systematically ran experiments in order to investigate the effect of the number of samples used on the approximated log-likelihood for values of H' ranging between 12 and 36. For these experiments we use the data described above, $H = 800$, and run 100 EM steps. We observe with $\text{BSC}^{\text{s+s}}$that 200 samples per hidden dimension (total samples $= 200 \times H'$) are sufficient in that the final value of the likelihood after 100 EM steps begins to saturate. Particularly, increasing the number of samples does not increase the likelihood by more than 1%. See Fig. 4**C**. In the shown set of experiments we used $H' = 20$, but this trend was observed in all experiments for all values of $H' \in [12, 36]$. Furthermore, we tested this number of samples $(200 \times H)$ in the pure sampling case ($\text{BSC}^{\text{sample}}$) in order to monitor the likelihood behavior. We observed two consistent trends: 1) the algorithm was never observed to converge to a high-likelihood solution, and 2) even when initialized at solution with high likelihood, the likelihood always decreases. This example demonstrates the gains of using select-and-sample above pure sampling: while $\text{BSC}^{\text{s+s}}$only needs $200 \times 20 = 4,000$ samples to robustly reach a high-likelihood solutions, by following the same regime with $\text{BSC}^{\text{sample}}$, not only did the algorithm poorly converge on a high-likelihood solution, but it used $200 \times 800 = 160,000$ samples to do so. See Figure 4**D**.

Large scale experiment on image patches. Comparison of the above results shows that the most efficient algorithm is obtained by a combination of preselection and sampling, our select-and-sample approach ($\text{BSC}^{\text{s+s}}$). This allows for increased computational efficiency with no or only minimal effect on the performance of the algorithm – as depicted in Fig. 3 and 4. This efficiency allows for applications to much larger scale problems than would be possible by individual approximation approaches (Fig. 4 and 4). To demonstrate the efficiency of the combined approach we applied $\text{BSC}^{\text{s+s}}$to a task with a very high number of observed and hidden dimensions. The dataset used was again extracted from the van Hateren image database. The same preprocessing as described earlier

[2]We restricted the set of images to 900 images without man-made structures (see Fig. 4A). The brightest 2% of the pixels were clamped to the maximal value of the remaining 98% (influences of light-reflections were reduced this way)

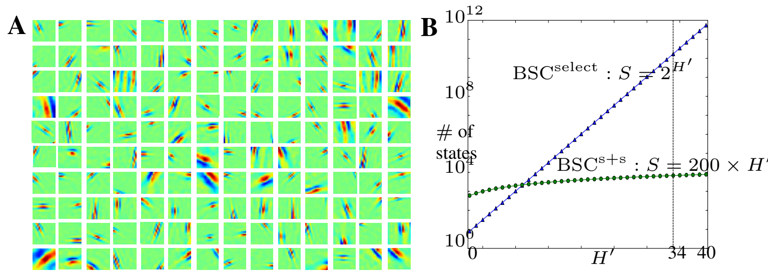

Figure 4: Results of the large-scale experiment of $\mathrm{BSC}^{\mathrm{s+s}}$on $N = 500,000$ image patches with $D = 40 \times 40 = 1600$ pixels, $H = 1600$ hidden dimensions, and $H' = 34$. **A** Random selection of the learned basis functions W_{dh}, exhibiting Gabor-like features. **B** Corresponding complexity in terms of the most costly operations for both of the preselection models: number of states the algorithm needs to consider. Note the difference in complexity at the $H' = 34$. Here $\mathrm{BSC}^{\mathrm{select}}$scales exponentially, whereas $\mathrm{BSC}^{\mathrm{s+s}}$scales linearly.

was performed, but this time $N = 500,000$ patches of size $D = 40 \times 40 = 1,600$ pixels were used. $\mathrm{BSC}^{\mathrm{s+s}}$was applied with the number of hidden units set to $H = 1,600$ with approximation parameter set to $H' = 34$. Using the same conditions as in the previous experiments (notably $S = 200 * H' = 64,000$ samples and 100 EM iterations) we again obtain a set of Gabor-like basis functions (see Fig. 4**A**). To the knowledge of the authors, the presented results represent the largest application of a sparse coding version with sophisticated posterior representation reported so far.

5 Discussion

In this paper, we introduced a novel efficient method for inference and unsupervised learning in probabilistic models that could plausibly be implemented in neural circuits, allowing the brain to use large scale models of the sensory input to make sense of its environment. The method could be plausibly realized in neural networks using two mechanisms that had been independently suggested in the context of a the statistical framework for perception: feed-forward input pre-selection [26], and sampling [12, 13, 3]. We showed that the two seemingly contrasting approaches can be combined based on their interpretation as approximate inference methods, resulting in a considerable increase in computational efficiency.

We investigated the applicability and efficiency of the method analytically and numercially using sparse coding model of natural images—a standard model for neural response properties in V1 [21, 30]. Comparisons to exact inference and learning, selection alone, and sampling alone showed a very favorable scaling behavior with the number of observed and hidden dimensions. To the best of our knowledge, the only other sparse coding implementation that reached a comparable problem size ($D = 20 \times 20, H = 2000$) assumed a Laplace prior, which results in a simple uni-modal posterior, and used a MAP approximation, which further reduces the posterior to a single point, and is known to produces biased parameters [22]. Our method is able to achieve convergence while still representing a significant part of the posterior, which is crucial for learning parameters like data noise and sparsity, and to correctly act when faced with uncertain input [2, 8, 3].

Concretely, here we used a sparse coding model with binary latent variables, mainly because it allowed us a systematic comparison with exact EM for low-dimensional problems, but it should be straight-forward to extend the method to the continuous case. In the model, the selection step results in a simple, local and neurally plausible integration of input data, given by (10). Sampling was implemented using the Gibbs sampling method, which is also neurally plausible, as neurons can individually sample their next state based on the current state of the other neurons, as transmitted through recurrent connections [15].

We expect the select-and-sample strategy to be widely applicable to machine learning models, whenever the posterior probability masses can be expected to be concentrated in a small sub-space of the whole latent space. The use of more sophisticated pre-selection mechanisms and sampling schemes could lead to further reduction in computational effort, although the details will depend in general on the particular generative model and input data.

References

[1] P. Dayan and L. F. Abbott. *Theoretical Neuroscience*. MIT Press, Cambridge, 2001.

[2] RPN Rao, BA Olshausen, and MS Lewicki. *Probabilistic Models of the Brain: Perception and Neural Function*. MIT Press, 2002.

[3] J Fiser, P Berkes, G Orban, and M Lengyel. Statistically optimal perception and learning: from behavior to neural representations. *Trends in Cognitive Sciences*, 14:119–130, 2010.

[4] MO Ernst and MS Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415:419–433, 2002.

[5] Y Weiss, EP Simoncelli, and EH Adelson. Motion illusions as optimal percepts. *Nature Neuroscience*, 5:598–604, 2002.

[6] KP Kording and DM Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427:244–247, 2004.

[7] JM Beck, WJ Ma, R Kiani, T Hanks, AK Churchland, J Roitman, MN Shadlen, PE Latham, and A Pouget. Probabilistic population codes for bayesian decision making. *Neuron*, 60(6):1142–1152, 2008.

[8] J Trommershäuser, LT Maloney, and MS Landy. Decision making, movement planning and statistical decision theory. *Trends in Cognitive Science*, 12:291–297, 2008.

[9] P Berkes, G Orban, M Lengyel, and J Fiser. Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, 331(6013):83–87, 2011.

[10] WJ Ma, JM Beck, PE Latham, and A Pouget. Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 9:1432–1438, 2006.

[11] R Turner, P Berkes, and J Fiser. Learning complex tasks with probabilistic population codes. In *Frontiers in Neuroscience*, 2011. Computational and Systems Neuroscience 2011.

[12] T.S. Lee and D. Mumford. Hierarchical Bayesian inference in the visual cortex. *Journal of the Optical Society of America A*, 20(7):1434–1448, 2003.

[13] PO Hoyer and A Hyvarinen. Interpreting neural response variability as Monte Carlo sampling from the posterior. In *Adv. Neur. Inf. Proc. Syst. 16*, pages 293–300. MIT Press, 2003.

[14] E Vul, ND Goodman, TL Griffiths, and JB Tenenbaum. One and done? Optimal decisions from very few samples. In *31st Annual Meeting of the Cognitive Science Society*, 2009.

[15] P Berkes, R Turner, and J Fiser. The army of one (sample): the characteristics of sampling-based probabilistic neural representations. In *Frontiers in Neuroscience*, 2011. Computational and Systems Neuroscience 2011.

[16] The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 1958.

[17] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 211(11):1019 – 1025, 1999.

[18] A. Yuille and D. Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, 10(7):301–308, 2006.

[19] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal. The 'wake-sleep' algorithm for unsupervised neural networks. *Science*, 268:1158 – 1161, 1995.

[20] E. Körner, M. O. Gewaltig, U. Körner, A. Richter, and T. Rodemann. A model of computation in neocortical architecture. *Neural Networks*, 12:989 – 1005, 1999.

[21] BA Olshausen and DJ Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[22] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. *Neural Information Processing Systems*, pages 801–808, 2007.

[23] Backpropagation applied to handwritten zip code recognition. 1989.

[24] M. Riesenhuber and T. Poggio. How visual cortex recognizes objects: The tale of the standard model. 2002.

[25] T. S. Lee and D. Mumford. Hierarchical bayesian inference in the visual cortex. *J Opt Soc Am A Opt Image Sci Vis*, 20(7):1434–1448, July 2003.

[26] J Lücke and J Eggert. Expectation Truncation And the Benefits of Preselection in Training Generative Models. *Journal of Machine Learning Research*, 2010.

[27] Gervasio Puertas, Jorg Bornschein, and Jorg Lucke. The maximal causes of natural scenes are edge filters. *Neural Information Processing Systems*, 23, 2010.

[28] Marc Henniges, Gervasio Puertas, Joerg Bornschein, Julian Eggert, and Joerg Luecke. Binary sparse coding. *Latent Variable Analysis and Signal Separation*, 2010.

[29] J Mairal, F Bach, J Ponce, and G Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11, 2010.

[30] J. Hateren and A. Schaaf. Independent Component Filters of Natural Images Compared with Simple Cells in Primary Visual Cortex. *Proc Biol Sci*, 265(1394):359–366, 1998.

[31] Dario L Ringach. Spatial Structure and Symmetry of Simple-Cell Receptive Fields in Macaque Primary Visual Cortex. *J Neurophysiol*, 88:455–463, 2002.

[32] M. Haft, R. Hofman, and V. Tresp. Generative binary codes. *Pattern Anal Appl*, 6(4):269–284, 2004.

[33] PO Hoyer. Non-negative sparse coding. *Neural Networks for Signal Processing XII: Proceedings of the IEEE Workshop*, pages 557–565, 2002.

[34] J Lücke. Receptive Field Self-Organization in a Model of the Fine Structure in V1 Cortical Columns. *Neural Computation*, 2009.