Topographic Product Models Applied To Natural Scene Statistics

Simon Osindero
Department of Computer Science
University of Toronto
Toronto
Ontario, M5S 3G4
Canada
osindero@cs.toronto.edu

Max Welling
Department of Computer Science
University of California Irvine
Irvine
CA 92697-3425
USA
welling@ics.uci.edu

Geoffrey E. Hinton
Canadian Institute for Advanced Research and Department of Computer Science
University of Toronto
Toronto
Ontario, M5S 3G4
Canada
hinton@cs.toronto.edu

Abstract

We present an energy-based model that uses a product of generalised Student-t distributions to capture the statistical structure in datasets. This model is inspired by and particularly applicable to "natural" datasets such as images. We begin by providing the mathematical framework, where we discuss complete and overcomplete models, and provide algorithms for training these models from data. Using patches of natural scenes we demonstrate that our approach represents a viable alternative to "independent components analysis" as an interpretive model of biological visual systems. Although the two approaches are similar in flavor there are also important differences, particularly when the representations are overcomplete. By constraining the interactions within our model we are also able to study the topographic organization of Gabor-like receptive fields that are learned by our model. Finally, we discuss the relation of our new approach to previous work — in particular Gaussian Scale Mixture models, and variants of independent components analysis.

1 Introduction

This paper presents a general family of energy-based models, which we refer to as "Product of Student-t" (PoT) models. They are particularly well suited to modelling statistical structure in data for which linear projections are expected to result in sparse marginal distributions. Many kinds of data might be expected to have such structure, and in particular "natural" datasets such as digitised images or sounds seem to be well described in this way.

The goals of this paper are two-fold. Firstly, we wish to present the general mathematical formulation of PoT models and to describe learning algorithms for them. We hope that this part of the paper will be useful in introducing a new method to the community's toolkit for machine learning and density estimation. Secondly, we focus on applying PoT's to capturing the statistical structure of natural scenes. This is motivated from both a density estimation perspective, and also from the perspective of providing insight into information processing within the visual pathways of the brain.

PoT models were touched upon briefly in Teh et al. (2003), and in this paper we present the basic formulation in more detail, provide hierarchical and topographic extensions, and give an efficient learning algorithm employing auxiliary variables and Gibbs sampling. We also provide a discussion of the PoT model in relation to similar existing techniques.

We suggest that the PoT model could be considered as a viable alternative to the more familiar technique of ICA when constructing density models, or performing feature extraction, or when building interpretive computational models of biological visual

systems. As we shall demonstrate, we are able to reproduce many of the successes of ICA — yielding results which are comparable, but with some interesting and significant differences. Similarly, extensions of our basic model can be related to some of the hierarchical forms of ICA that have been proposed, as well as to Gaussian Scale Mixtures. Again there are interesting differences in formulation. An example of a potential advantage in our approach is that the learned representations can be inferred directly from the input, without the need for any iterative settling even in hierarchical or highly overcomplete models.

The paper is organised as follows. Section 2 describes the mathematical form of the basic PoT model along with extensions to hierarchical and topographic versions. Section 3 then describes how to learn within the PoE framework using the contrastive divergence (CD) algorithm (Hinton, 2002) (with Appendix A providing the background material for running the necessary Markov Chain Monte Carlo sampling). Then in section 4 we present results of our model when applied to natural images. We are able to recreate the success of ICA based models like, for example, Bell and Sejnowski (1995, 1997), Olshausen and Field (1996, 1997), Hoyer and Hyvarinen (2000), Hyvarinen et al. (2001), and Hyvarinen and Hoyer (2001). Our model provides computationally motivated accounts for the form of simple cell and complex cell receptive fields, as well as for the basic layout of cortical topographic maps for location, orientation, spatial frequency, and spatial phase. Additionally, we are easily able to produce such results in an overcomplete setting.

In section 5 we analyse in more detail the relationships between our PoT model, ICA models and their extensions, and Gaussian Scale Mixtures, and finally in section 6 we summarise our work.

2 Products of Student-t Models

We will begin with a brief overview of product of expert models (Hinton, 2002) in section 2.1, before presenting the basic product of Student-t model (Welling et al., 2002a) in section 2.2. Then we move on to discuss hierarchical topographic extensions in sections 2.3, 2.4 and 2.5.

2.1 Product of Expert Models

Product of expert models, or PoEs, were introduced in Hinton (2002) as an alternative method of combining expert models into one joint model. In contrast to mixture of expert models, where individual models are combined additively, PoEs combine expert

opinions multiplicatively as follows (see also Heskes (1998)),

$$P_{PoE}(\mathbf{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{i=1}^{M} p_i(\mathbf{x}|\theta_i) \tag{1}$$

where $Z(\boldsymbol{\theta})$ is the global normalization constant and $p_i(\cdot)$ are the individual expert models. Mixture models employ a "divide and conquer" strategy with different "experts" being used to model different subsets of the training data. In product models, many experts cooperate to explain each input vector and different experts specialize in different parts of the input vector or in different types of latent structure. If a scene contains n different objects that are processed in parallel, a mixture model needs a number of components that is exponential in n because each component of the mixture must model a *combination* of objects. A product model, by contrast, only requires a number of components that is linear in n because many different experts can be used at the same time.

Another benefit of product models is their ability to model sharp boundaries. In mixture models, the distribution represented by the whole mixture must be vaguer than the distribution represented by a typical component of the mixture. In product models, the product distribution is typically much sharper than the distributions of the individual experts[1], which is a major advantage for high dimensional data (Hinton, 2002; Welling et al., 2002b).

Learning PoE models has been difficult in the past, mainly due to the presence of the partition function $Z(\boldsymbol{\theta})$. However, contrastive divergence learning (Hinton, 2002) (see section 3.2) has opened the way to apply these models to large scale applications.

PoE models are related to many other models that have been proposed in the past. In particular, log-linear models[2] have a similar flavor, but are more limited in their parametrization:

$$P_{LogLin}(\mathbf{x}|\lambda) = \frac{1}{Z(\lambda)} \prod_{i=1}^{M} \exp\left(\lambda_i f_i(\mathbf{x})\right) \tag{2}$$

where $\exp[\lambda_i f_i(\cdot)]$ takes the role of an un-normalized expert. A binary product of experts model was first introduced under the name "harmonium" in Smolensky (1986). A learning algorithm based on projection pursuit was proposed in Freund and Haussler (1992). In addition to binary models (Hinton, 2002), the Gaussian case been stud-

[1]When multiplying together n equal-variance Gaussians, for example, the variance is reduced by a factor of n. It is also possible to make the entropy of the product distribution higher than the entropy of the individual experts by multiplying together two very heavy-tailed distributions whose modes are in very different places.

[2]Otherwise known as exponential family models, maximum entropy models and additive models. For example see Zhu et al. (1998)

ied (Williams et al., 2001; Marks and Movellan, 2001; Williams and Agakov, 2002; Welling et al., 2003a).

2.2 Product of Student-t (PoT) Models

The basic model we study in this paper is a form of PoE suggested by Hinton and Teh (2001) where the experts are given by generalized Student-t distributions:

$$\mathbf{y} = \mathbf{J}\mathbf{x} \tag{3}$$

$$p_i(y_i|\alpha_i) \propto \frac{1}{(1+\frac{1}{2}y_i^2)^{\alpha_i}} \tag{4}$$

The variables y_i are the responses to linearly filtered input vectors and can be thought of as latent variables that are deterministically related to the observables, $\mathbf{x}$. Through this deterministic relationship, equation 4 defines a probability density on the observables. The filters, $\{\mathbf{J}_i\}$, are learnt from the training data (typically images) by maximizing or approximately maximizing the log likelihood.

Note that due to the presence of the $\mathbf{J}$ parameters this product of Student-t (PoT) model is not log-linear. However, it is possible to introduce auxiliary variables, $\mathbf{u}$, such that the joint distribution $P(\mathbf{x},\mathbf{u})$ is log-linear[3] and the marginal distribution $P(\mathbf{x})$ reduces to that of the original PoT distribution,

$$P_{PoT}(\mathbf{x}) = \int_0^\infty d\mathbf{u}\, P(\mathbf{x},\mathbf{u}) \tag{5}$$

$$P(\mathbf{x},\mathbf{u}) \propto \exp\left[-\sum_{i=1}^{M}\left(u_i\left(1+\frac{1}{2}(\mathbf{J}_i\mathbf{x})^2\right)+(1-\alpha_i)\log u_i\right)\right] \tag{6}$$

where $\mathbf{J}_i$ denotes the row-vector corresponding to the i^{th} row of the filter matrix $\mathbf{J}$. An intuition for this form of reparametrisation with auxiliary variables can be gained by considering that a one dimensional t-distribution can be written as a continuous mixture of Gaussians, with a Gamma distribution controlling mixing proportions on components with different precisions. That is to say,

$$\frac{\Gamma\left(\alpha+\frac{1}{2}\right)}{\Gamma(\alpha)\sqrt{2\pi}}\left(1+\frac{1}{2}\tau^2\right)^{-\left(\alpha+\frac{1}{2}\right)} = \int \mathrm{d}\lambda \overbrace{\left(\frac{1}{\sqrt{2\pi}}\lambda^{\frac{1}{2}}e^{-\frac{1}{2}\tau^2\lambda}\right)}^{\text{Gaussian}}\overbrace{\left(\frac{1}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\lambda}\right)}^{\text{Gamma}} \tag{7}$$

The advantage of this reformulation using auxiliary variables is that it supports an

[3]Note that it is log-linear in the parameters $\theta_{ijk}=\mathbf{J}_{ij}\mathbf{J}_{ik}$ and α_i with features $u_i x_j x_k$ and $\log u_i$.

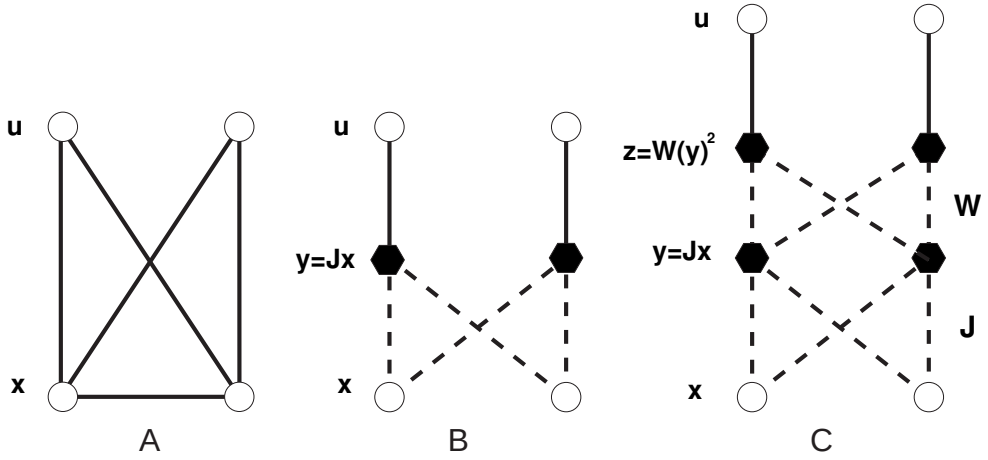

Figure 1: (A) Standard PoT model as an undirected graph or Markov random field (MRF) involving observables, $\mathbf{x}$ and auxiliary variables, $\mathbf{u}$. (B) Standard PoT MRF redrawn to show the role of deterministic filter outputs $\mathbf{y} = \mathbf{J}\mathbf{x}$. (C) Hierarchical PoT MRF drawn to show both sets of deterministic variables, $\mathbf{y}$ and $\mathbf{z} = \mathbf{W}(\mathbf{y})^2$, as well as auxiliary variables $\mathbf{u}$.

efficient, fast-mixing Gibbs sampler which is in turn beneficial for contrastive divergence learning. The Gibbs chain samples alternately from $P(\mathbf{u}|\mathbf{x})$ and $P(\mathbf{x}|\mathbf{u})$ given by,

$$P(\mathbf{u}|\mathbf{x}) = \prod_{i=1}^{M} \mathcal{G}_{u_i}\left[\alpha_i\ ;\ 1 + \frac{1}{2}(\mathbf{J}_i\mathbf{x})^2\right] \tag{8}$$

$$P(\mathbf{x}|\mathbf{u}) = \mathcal{N}_{\mathbf{x}}\left[0\ ;\ (\mathbf{J}\mathbf{V}\mathbf{J}^T)^{-1}\right] \qquad \mathbf{V} = \mathbf{Diag}[\mathbf{u}] \tag{9}$$

where $\mathcal{G}$ denotes a Gamma distribution and $\mathcal{N}$ a normal distribution. From (9) we see that the variables $\mathbf{u}$ can be interpreted as *precision* variables in the transformed space $\mathbf{y} = \mathbf{J}\mathbf{x}$.

In terms of graphical models the representation that best fits the PoT model with auxiliary variables is that of a two-layer bipartite undirected graphical model. Figure 1 (A) schematically illustrates the MRF over $\mathbf{u}$ and $\mathbf{x}$; figure 1 (B) shows the role of the deterministic filter outputs in this scheme.

A natural way to interpret the differences between directed models (and in particular ICA models) and PoE models was provided in Hinton and Teh (2001); Teh et al. (2003). Whereas directed models intuitively have a top-down interpretation (e.g. samples can be obtained by ancestral sampling starting at the top-layer units), PoE models (or more generally "energy-based models") have a more natural bottom-up interpretation. The probability of an input vector is proportional to $\exp(-E(\mathbf{x}))$ where the energy $E(\mathbf{x})$

is computed bottom-up starting at the input layer (e.g. $E(\mathbf{y}) = E(\mathbf{Jx})$). We may thus interpret the PoE model as modelling a collection of soft constraints, parameterized through deterministic mappings from the input layer to the top layer (possibly parameterized as a neural network) and where the energy serves to penalize inputs that do not satisfy these constraints (e.g. are different from zero). The costs contributed by the violated constraints are added to compute the global energy, which is equivalent to multiplying the distributions of the individual experts to compute the product distribution (since $P(\mathbf{x}) \propto \prod_i p_i(\mathbf{x}) \propto \exp(-\sum_i E_i(\mathbf{x}))$).

For a PoT, we have a two-layer model where the constraint-violations are penalized using the energy function (see equation.6),

$$E(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i \log\left(1 + \frac{1}{2}(\mathbf{J}_i\mathbf{x})^2\right) \tag{10}$$

We note that the shape of this energy function implies that, relative to a quadratic penalty, small violations are penalized more strongly whilst large violations are penalized less strongly. This results in "sparse" distributions of violations ($\mathbf{y}$-values) with many very small violations and occasional large ones.

In the case of equal number of observables, $\{x_i\}$, and latent variables, $\{y_i\}$ (the so called "complete representation"), the PoT model is formally equivalent to square, noiseless "independent components analysis" (ICA) (Bell and Sejnowski, 1995) with Student-t priors. However, in the overcomplete setting (more latent variables than observables) product of experts models are essentially different from overcomplete ICA models (Lewicki and Sejnowski, 2000). The main difference is that the PoT maintains a deterministic relationship between latent variables and observables through $\mathbf{y} = \mathbf{Jx}$, and consequently not all values of $\mathbf{y}$ are allowed. This results in important marginal dependencies between the $\mathbf{y}$-variables. In contrast, in overcomplete ICA the hidden $\mathbf{y}$-variables are marginally independent by assumption and have a stochastic relationship with the $\mathbf{x}$-variables. For more details we refer to Teh et al. (2003).

For undercomplete models (fewer latent variables than observables) there is again a discrepancy between PoT models and ICA models. In this case the reason can be traced back to the way noise is added to the models in order to force them to assign non-zero probability everywhere in input space. In contrast to undercomplete ICA models where noise is added in all directions of input space, undercomplete PoT models have noise added only in the directions orthogonal to the subspace spanned by the filter matrix $\mathbf{J}$. More details can be found in Welling et al. (2003b, 2004).

2.3 Hierarchical PoT (HPoT) Models

We now consider modifications to the basic PoT by introducing extra interactions between the activities of filter outputs, y_i, and by altering the energy function for the

model. These modifications were motivated by observations of the behaviour of 'independent' components of natural data, and inspired by similarities between our model and (hierarchical) ICA. Since the new model essentially involves adding a new layer to the standard PoT, we refer to it as a hierarchical PoT (HPoT).

As we will show in Section 4, when trained on a large collection of natural image patches the linear components $\{\mathbf{J}_i\}$ behave similarly to the learnt basis functions in ICA and grow to resemble the well-known Gabor-like receptive fields of simple cells found in the visual cortex (Bell and Sejnowski, 1997). These filters, like wavelet transforms, are known to de-correlate input images very effectively. However, it has been observed that higher order dependencies remain between the filter outputs $\{y_i\}$. In particular there are important dependencies between the "activities" or "energies" y_i^2 (or more generally $|y_i|^\beta,\ \beta > 0$) of the filter outputs. This phenomenon can be neatly demonstrated through the use of "bow-tie plots", in which the conditional histogram of one filter output is plotted given the output value of a different filter (e.g. see Simoncelli (1997)). The bow-tie shape of the plots implies that the first order dependencies have been removed by the linear filters $\{\mathbf{J}_i\}$ (since the conditional mean vanishes everywhere), but that higher order dependencies still remain; specifically, the variance of one filter output can be predicted from the activity of neighboring filter outputs.

In our modified PoT the interactions between filter outputs will be implemented by first squaring the filter outputs and subsequently introducing an extra layer of units, denoted by $\mathbf{z}$. These units will be used to capture the dependencies between these squared filter outputs: $\mathbf{z} = \mathbf{W}(\mathbf{y})^2 = \mathbf{W}(\mathbf{J}\mathbf{x})^2$, and this is illustrated in figure 1 (c). (Note that in the previous expression, and in what follows, the use of $(\cdot)^2$ with a vector argument will imply a component-wise squaring operation.) The modified energy function is

$$E(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i \log\left(1 + \frac{1}{2}\sum_{j=1}^{K} W_{ij}(\mathbf{J}_j\mathbf{x})^2\right) \qquad \mathbf{W} \geq 0 \tag{11}$$

where the non-negative parameters W_{ij} model the dependencies between the activities[4] $\{y_i^2\}$. Note that the forward mapping from $\mathbf{x}$, through $\mathbf{y}$, to $\mathbf{z}$ is completely deterministic, and can be interpreted as a bottom-up neural network. We can also view the modified PoT as modelling constraint violations, but this time in terms of $\mathbf{z}$ with violations now penalized according to the energy in Equation 11.

As with the standard PoT model, there is a reformulation of the hierarchical PoT

[4]For now, we implicitly assume that the number of first hidden-layer units (i.e. filters) is greater than or equal to the number of input dimensions. Models with fewer filters than input dimensions need some extra care, as noted in section 2.3.1. The number of top-layer units can be arbitrary, but for concreteness we will work with an equal number of first-layer and top-layer units.

model in terms of auxiliary variables, $\mathbf{u}$,

$$P(\mathbf{x},\mathbf{u}) \propto \exp\left[-\sum_{i=1}^{M}\left(u_i\left(1+\frac{1}{2}\sum_{j=1}^{K}W_{ij}(\mathbf{J}_j\mathbf{x})^2\right)+(1-\alpha_i)\log u_i\right)\right] \qquad (12)$$

with conditional distributions,

$$P(\mathbf{u}|\mathbf{x}) = \prod_{i=1}^{M}\mathcal{G}_{u_i}\left[\alpha_i\ ;\ 1+\frac{1}{2}\sum_{j=1}^{K}W_{ij}(\mathbf{J}_j\mathbf{x})^2\right] \qquad (13)$$

$$P(\mathbf{x}|\mathbf{u}) = \mathcal{N}_{\mathbf{x}}\left[0\ ;\ (\mathbf{J}\mathbf{V}\mathbf{J}^T)^{-1}\right] \qquad \mathbf{V}=\mathbf{Diag}[\mathbf{W}^T\mathbf{u}] \qquad (14)$$

Again, we note that this auxiliary variable representation supports an efficient Gibbs sampling procedure where all auxiliary variables $\mathbf{u}$ are sampled in parallel given the inputs $\mathbf{x}$ using Eqn. 13 and all input variables $\mathbf{x}$ are sampled jointly from a multivariate Gaussian distribution according to Eqn.14. As we will discuss in section 3.2, this is an important ingredient in training (H)PoT models from data using contrastive divergence.

Finally, in a somewhat speculative link to computational neuroscience, in the following discussions we will refer to units, $\mathbf{y}$, in the first hidden layer as 'simple cells' and units, $\mathbf{z}$, in the second hidden layer as 'complex cells'. For simplicity, we will assume the number of simple and complex cells to be equal. There are no obstacles to using unequal numbers, but this does not appear to lead to any qualitatively different behaviour.

2.3.1 Undercomplete HPoT Models

The HPoT models, as defined in section 2.3, were implicitly assumed to be complete or overcomplete. We may also wish to consider undercomplete models. These models can be interesting in a variety of applications where one seeks to represent the data in a lower dimensional yet informative space.

Undercomplete models need a little extra care in their definition, since in the absence of a proper noise model they are un-normalisable over input space. In Welling et al. (2003a,b, 2004) a natural solution to this dilemma was proposed where a noise model is added in directions orthogonal to all of the filters $\{\mathbf{J}\}$. We note that it is possible to generalise this procedure to HPoT models, but in the interests of parsimony we omit detailed discussion of undercomplete models in this paper.

2.4 Topographic PoT Models

The modifications described next were inspired by a similar proposal in Hyvarinen et al. (2001) named "topographic ICA". By restricting the interactions between the first and

second layers of a HPoT model we are able to induce a topographic ordering on the learnt features.

Such ordering can be useful for a number of reasons; for example it may help with data visualisation by concentrating feature activities in local regions. This restriction can also help in acting as a regulariser for the density models which we learn. Furthermore, it makes it possible to compare the topographic organisation of features in our model (and based on the statistical properties of the data) to the organisation found within cortical topographic maps.

We begin by choosing a topology on the space of filters. This is most conveniently done by simply considering the filters to be laid out on a grid, and considering local neighbourhoods with respect to this layout. In our experiments, we use a regular square grid and apply toroidal boundary conditions to avoid edge effects.

The complex cells receive input from the simple cells in precisely the same way as in our HPoT model: $y_i = \sum_j \mathbf{W}_{ij}(\mathbf{J}_j\mathbf{x})^2$, but now $\mathbf{W}$ is fixed and we assume it is chosen such that it computes a local *average* from the grid of filter activities. The free parameters that remain to be learnt using contrastive divergence are $\{\alpha_i, \mathbf{J}\}$. In the following we will explain why the filters $\{\mathbf{J}_i\}$ should be expected to organize themselves topographically when learnt from data.

As noted previously, there are important dependencies between the activities of wavelet coefficients of filtered images. In particular, the variance (but not the mean) of one coefficient can be predicted from the value of a "neighboring" coefficient. The topographic PoT model can be interpreted as an attempt to model these dependencies through a Markov random field on the activities of the simple cells. However, we have pre-defined the connectivity pattern and have left the filters to be determined through learning. This is the opposite strategy as the one used in, for instance, Portilla et al. (2003) where the wavelet transform is fixed and the interactions between wavelet coefficients are modelled. One possible explanation for the emergent topography is that the model will make optimal use of these pre-defined interactions if it organizes its simple cells such that dependent cells are nearby in filter space and independent ones are distant.[5]

A complementary explanation is based on the interpretation of the model as capturing complex constraints in the data. The penalty function for violations is designed such that (relative to a squared penalty) large violations are relatively mildly penalized. However, since the complex cells represent the average input from simple cells, their values would be well described by a Gaussian distribution if the corresponding simple cells were approximately independent. (This is a consequence of the central limit theorem for sums of independent random variables.) In order to avoid a mismatch between

[5]This argument assumes that the shape of the filters remains essentially unchanged (i.e. Gabor-like) by the introduction of the complex cells in the model. Empirically we see that this is indeed the case.

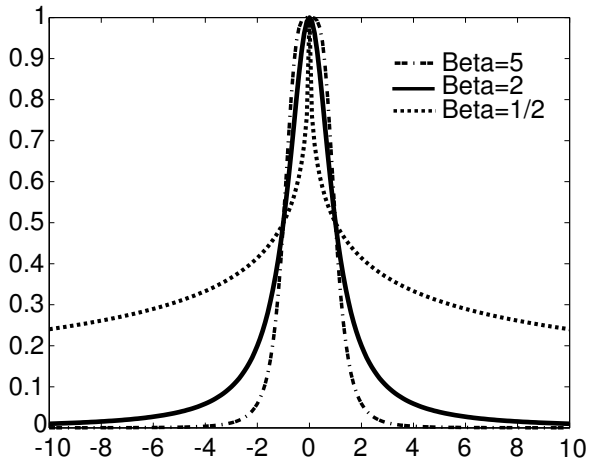

Figure 2: Functions $f(x) = 1/(1 + |x|^{\beta})$ for different values of β.

the distribution of complex cell outputs and the way they are penalized, the model ought to position simple cells that have correlated activities near to each other. In doing so, the model can escape the central limit theorem because the simple cell outputs that are being pooled are no longer independent. Consequently, the pattern of violations that arises is a better match to the pattern of violations which one would expect from the penalising energy function.

Another way to understand the pressure towards topography is to ask how an individual simple cell should be connected to the complex cells in order to minimize the total cost caused by the simple cell's outputs on real data. If the simple cell is connected to complex cells that already receive inputs from the simple cell's neighbors in position and spatial frequency, the images that cause the simple cell to make a big contribution will typically be those in which the complex cells that it excites are already active, so its additional contribution to the energy will be small because of the gentle slope in the heavy tails of the cost function. Hence, since complex cells locally pool simple cells, local similarity of filters is expected to emerge.

2.5 Further Extensions To The Basic PoT Model

The parameters $\{\alpha_i\}$ in the definition of the PoT model control the "sparseness" of the activities of the complex and simple cells. For large values of α, the PoT model will resemble more and more a Gaussian distribution, while for small values there is a very sharp peak at zero in the distribution which decays very quickly into "fat" tails.

In the HPoT model, the complex cell activities, $\mathbf{z}$, are the result of linearly combining the (squared) outputs simple cells, $\mathbf{y} = \mathbf{J}\mathbf{x}$. The squaring operation is a somewhat

arbitrary choice (albeit a computationally convenient and empirically effective one), and we may wish to process the first layer activities in other ways before we combining them in the second layer. In particular, we might consider modifications of the form: activity=$|\mathbf{J}\mathbf{x}|^\beta$ with $|\cdot|$ denoting absolute values and $\beta > 0$. Such a model defines the a density in $\mathbf{y}$-space of the form,

$$p_y(\mathbf{y}) = \frac{1}{Z(\mathbf{W}, \boldsymbol{\alpha})} \exp\left[-\sum_{i=1}^{M} \alpha_i \log\left(1 + \frac{1}{2}\sum_{j=1}^{K} W_{ij}|y_j|^\beta\right)\right] \tag{15}$$

A plot of the un-normalized distribution $f(x) = 1/(1 + |x|^\beta)$ is shown in figure 2 for three settings of the parameter β. One can observe that for smaller values of β the peak at zero become sharper and the tails become "fatter".

In section 3 we will show that sampling and hence learning with contrastive divergence can be performed efficiently for any setting of β.

3 Learning in HPoT Models

In this section we will explain how to perform maximum likelihood learning of the parameters for the models introduced in the previous section. In the case of complete and undercomplete PoT models we are able to analytically compute gradients, however in the general case of overcomplete or hierarchical PoT's we are required to employ an approximation scheme and the preferred method in this paper will be contrastive divergence (CD) (Hinton, 2002). Since CD learning is based on Markov chain Monte Carlo sampling, Appendix A provides a discussion of sampling procedures for the various models we have introduced.

3.1 Maximum Likelihood Learning in (H)PoT Models

To learn the parameters $\boldsymbol{\theta} = (\mathbf{J}, \mathbf{W}, \boldsymbol{\alpha})$ (and $\boldsymbol{\beta}$ for the extended models), we will maximize the log-likelihood of the model,

$$\boldsymbol{\theta}^{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L} = \arg\max_{\boldsymbol{\theta}} \frac{1}{N}\sum_{n=1}^{N} \log p_x(\mathbf{x}_n; \boldsymbol{\theta}) \tag{16}$$

For models which have the Boltzmann form, $p(\mathbf{x}) = \frac{1}{Z}\exp[-E(\mathbf{x}; \theta)]$, we can compute the following gradient,

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \mathbb{E}\left[\frac{\partial E(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right]_p - \frac{1}{N}\sum_{n=1}^{N} \frac{\partial E(\mathbf{x}_n; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{17}$$

where $\mathbb{E}[\cdot]_p$ denotes expectation with respect to the model's distribution over $\mathbf{x}$ (this term comes from the derivatives of the log partition function, Z). For the parameters $(\mathbf{J}, \mathbf{W}, \boldsymbol{\alpha})$ in the PoT we obtain the following derivative functions,

$$\frac{\partial E(\mathbf{x};\boldsymbol{\theta})}{\partial J_{jk}} = \sum_i \frac{\alpha_i W_{ij} (\mathbf{J}\mathbf{x})_j x_k}{1 + \frac{1}{2}\sum_{j'} W_{ij'} (\mathbf{J}\mathbf{x})_{j'}^2} \tag{18}$$

$$\frac{\partial E(\mathbf{x};\boldsymbol{\theta})}{\partial W_{ij}} = \frac{\frac{1}{2}\alpha_i (\mathbf{J}\mathbf{x})_j^2}{1 + \frac{1}{2}\sum_{j'} W_{ij'} (\mathbf{J}\mathbf{x})_{kj'}^2} \tag{19}$$

$$\frac{\partial E(\mathbf{x};\boldsymbol{\theta})}{\partial \alpha_i} = \log\left(1 + \frac{1}{2}\sum_j W_{ij} (\mathbf{J}\mathbf{x})_j^2\right) \tag{20}$$

Once we have computed the gradients of the log-likelihood, we can maximize it using any gradient-based optimization algorithm.

Elegant as the gradients in Eqn.17 may seem, in the general case they are intractable to compute. The reason is the expectation in the first term of Eqn.17 over the model distribution. One may choose to approximate this average by running a MCMC chain to equilibrium which has $p(\mathbf{x};\boldsymbol{\theta})$ as its invariant distribution. However, there are (at least) two reasons why this might not be a good idea: 1) The Markov chain has to be run to equilibrium for every gradient step of learning and 2) we need a lot of samples to reduce the variance in the estimates.

Hence, for the general case, we propose to use the contrastive divergence learning paradigm which is discussed next.

3.2 Training (H)PoT Models with Contrastive Divergence

For complete and undercomplete HPoT models we can derive the exact gradient of the log-likelihood with respect to the parameters $\mathbf{J}$. In the complete case these gradients turn out to be of the same form as the update rules proposed in Bell and Sejnowski (1995). However, the gradients for the parameters $\mathbf{W}$ and α are much harder to compute.[6] Furthermore, in overcomplete settings the exact gradients with respect to all parameters are computationally intractable.

We now describe an approximate learning paradigm to train the parameters in cases where evaluation of the exact gradients is intractable. Recall that the bottleneck in computing these gradients is the first term in the equation 17. An approximation to these expectations can be obtained by running a MCMC sampler with $p(\mathbf{x};\mathbf{J},\mathbf{W},\boldsymbol{\alpha})$ as its invariant distribution and computing Monte Carlo estimates of the averages. As mentioned in section 3.1 this is a very inefficient procedure because it needs to be repeated

[6] Although we can obtain exact derivatives for α in the special case where $\mathbf{W}$ is restricted to be the identity matrix.

for every step of learning and a fairly large number of samples may be needed to reduce the variance in the estimates[7]. Contrastive divergence (Hinton, 2002), replaces the MCMC samples in these Monte Carlo estimates with samples from brief MCMC runs, which were initialized at the data-cases. The intuition is that if the current model is not a good fit for the data, the MCMC particles will swiftly and consistently move away from the data cases. On the other hand, if the data population represents a fair sample from the model distribution, then the average energy will not change when we initialize our Markov chains at the data cases and run them forward. In general, initializing the Markov chains at the data and running them only briefly introduces bias but greatly reduces both variance and computational cost. **Algorithm 1** summarise the steps in this learning procedure.

Algorithm 1 Contrastive Divergence Learning

1. Compute the gradient of the energy with respect to the parameters, $\boldsymbol{\theta}$, and average over the data cases $\mathbf{x}_n$.
2. Run MCMC samplers for k steps, starting at every data vector $\mathbf{x}_n$, keeping only the last sample $\mathbf{s}_{n,k}$ of each chain.
3. Compute the gradient of the energy with respect to the parameters, $\boldsymbol{\theta}$, and average over the samples $\mathbf{s}_{n,k}$.
4. Update the parameters using,

$$\Delta\boldsymbol{\theta} = \frac{\eta}{N}\left(\sum_{\text{samples } \mathbf{s}_{n,k}} \frac{\partial E(\mathbf{s}_{nk})}{\partial \boldsymbol{\theta}} - \sum_{\text{data } \mathbf{x}_n} \frac{\partial E(\mathbf{x}_n)}{\partial \boldsymbol{\theta}}\right) \tag{21}$$

where η is the learning rate and N the number of samples in each mini-batch.

For further details on contrastive divergence learning we refer to the literature (Hinton, 2002; Teh et al., 2003; Yuille, 2004; Carreira-Perpinan and Hinton, 2005). For highly overcomplete models it often happens that some of the $\mathbf{J}_i$-filters (rows of $\mathbf{J}$) decay to zero. To prevent this from happening we constrain the L_2-norm of these filters to be one: $\sum_j J_{ij}^2 = 1 \; \forall i$. Also, constraining the norm of the rows of the $\mathbf{W}$ matrix was helpful during learning. We choose to constrain the L_1-norm to unity $\sum_j W_{ij} = 1 \; \forall i$, which makes sense because $W_{ij} \geq 0$.

We note that the objective function is not convex and so the existence of poor local minima could be a concern. The stochastic nature of our gradient descent procedure may provide some protection against being trapped in shallow minima, although it has the concomitant price of being slower than noise-free gradient descent. We also note that the intractability of the partition function makes it difficult to obtain straightforward

[7] An additional complication is that it is hard to assess when the Markov chain has converged to the equilibrium distribution.

objective measures of model performance since log-probabilities can only be computed up to an unknown additive constant. This is not so much of a problem when one is using a trained model for, say, feature extraction, statistical image processing or classification, but it does make explicit comparison with other models rather hard. (For example there is no straightforward way to compare the densities provided by our overcomplete (H)PoT models with those from overcomplete ICA-style models.)

4 Experiments on Natural Images

There are several reasons to believe that the HPoT should be an effective model for capturing and representing the statistical structure in natural images; indeed much of its form was inspired by the dependencies that have been observed in natural images.

We have applied our model to small patches taken from digitised natural images. The motivation for this is several-fold. Firstly, it provides a useful test of the behaviour of our model on a dataset that we believe to contain sparse structure (and therefore to be well suited to our framework). Secondly, it allows us to compare our work with that from other authors and similar models, namely ICA. Thirdly, it allows us to use our model framework as a tool for interpreting results from neurobiology. Our method can complement existing approaches and also allows one to suggest alternative interpretations and descriptions of neural information processing.

Section 4.2 presents results from complete and overcomplete single layer PoT's trained on natural images. Our results are qualitatively similar to those obtained using ICA. In section 4.3 we demonstrate the higher order features learnt in our hierarchical PoT model, and in section 4.4 we present results from topographically constrained hierarchical PoT's. The findings in these two sections are qualitatively similar to the work by Hyvarinen et al. (2001), however our underlying statistical model is different and allows us to deal more easily with overcomplete, hierarchical topographic representations.

4.1 Datasets and Preprocessing

We performed experiments using standard sets of digitised natural images available on the World Wide Web from Aapo Hyvarinen[8] and Hans van Hateren[9]. The results obtained from the two different datasets were not significantly different, and for the sake of simplicity all results reported here are from the van Hateren dataset.

To produce training data of a manageable size, small square patches were extracted from randomly chosen locations in the images. As is common for unsupervised learning, these patches were filtered according to computationally well-justified versions

[8] `http://www.cis.hut.fi/projects/ica/data/images/`
[9] `http://hlab.phys.rug.nl/imlib/index.html`

of the sort of whitening transformations performed by the retina and LGN (Atick and Redlich, 1992). First we applied a log transformation to the 'raw' pixel intensities. This procedure somewhat captures the contrast transfer function of the retina. It is not critical, but for consistency with past work we incorporated it for the results presented here. The extracted patches were subsequently normalized such that mean pixel intensity for a given pixel across the data-set was zero, and also so that the mean intensity within each patch was zero — effectively removing the DC component from each input. The patches were then whitened, usually in conjunction with dimensionality reduction. This is a standard technique in many ICA approaches and speeds up learning without having much impact on the final results obtained.

4.2 Single Layer PoT Models

Figure 3 illustrates results from our basic approach, and shows for comparison results obtained using ICA. The data consisted of $150,000$ patches of size 18×18 that were reduced to vectors of dimension 256 by projection onto the leading 256 eigenvectors of the data covariance matrix, and then whitened to give unit variance along each axis.

Complete Models

We first present the results of our basic approach in a complete setting, and display a comparison of the filters learnt using our method with a set obtained from an equivalent ICA model learnt using direct gradient ascent in the likelihood. We trained both models (learning just $\mathbf{J}$, and keeping α fixed[10] at 1.5) for 200 passes through the entire dataset of $150,000$ patches. The PoT was trained using one-step contrastive divergence as outlined in section 3.2 and the ICA model was trained using the exact gradient of the log-likelihood (as in Bell and Sejnowski (1995) for instance). As expected, at the end of learning the two procedures delivered very similar results, exemplars of which are given in figure 3 (A) & (B). Furthermore, both sets of filters bear a strong resemblance to the types of simple cell receptive fields found in V1.

Overcomplete Models

We next consider our model in an overcomplete setting; this is no longer equivalent to any ICA model. In the PoT, overcomplete representations are simple generalisations of the complete case and, *unlike* causal generative approaches, the features are conditionally independent since they are just given by a deterministic mapping.

[10]This is the minimum value of α that allows us to have a well behaved density model in the complete case. As alpha gets smaller than this, the tails of the distribution get heavier and heavier and the variance and eventually mean are no longer well defined.

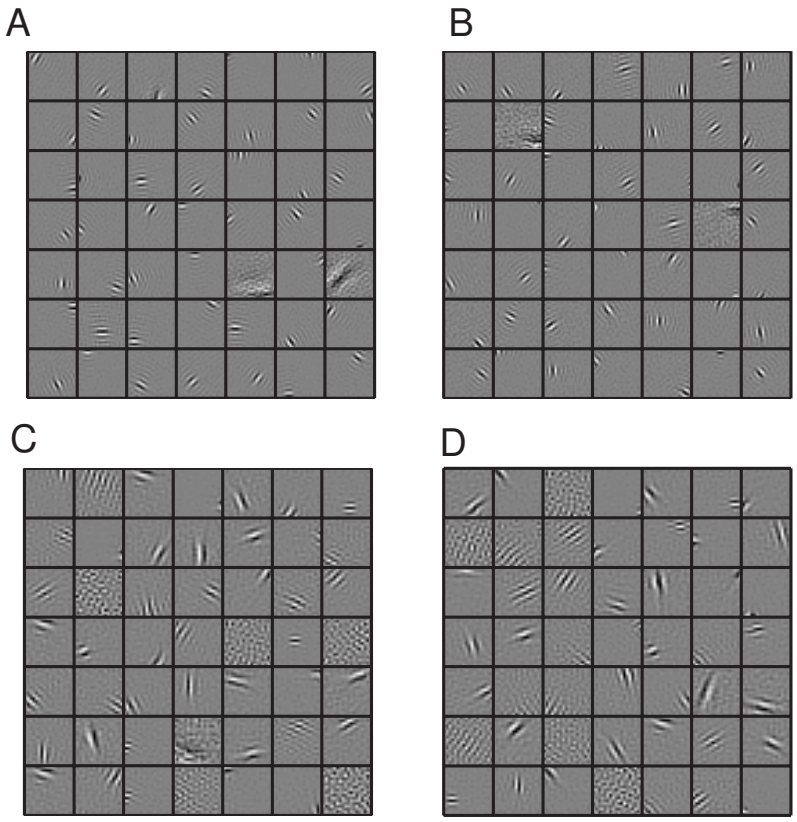

Figure 3: Learnt filters shown in the raw data space. Each small square represents a filter vector, plotted as an image. The gray scale of each filter display has been (symmetrically) scaled to saturate at the maximum absolute weight value. (A) Random subset of filters learnt in a complete PoT model. (B) Random subset of filters learnt in a complete ICA model. (C) Random subset of filters learnt in a $1.7\times$ overcomplete PoT model. (D) Random subset of filters learnt in a $2.4\times$ overcomplete PoT model.

To facilitate learning in the overcomplete setting we have found it beneficial to make two modifications to the basic set-up. Firstly, we set $\alpha_i = \alpha \, \forall i$, and make α a free parameter to be learnt from the data. The learnt value of α is typically less than 1.5 and gets smaller as we increase the degree of overcompleteness[11]. One intuitive way of understanding why this might be expected is the following. Decreasing α reduces the "energy cost" for violating the constraints specified by each individual feature, however this is counterbalanced by the fact that in the overcomplete setting we expect an input to violate more of the constraints at any given time. If α remains constant as more features are added, the mass in the tails may no longer be sufficient to model the distribution well.

The second modification that we make is to constrain the $L2$-norm of the filters to l, making l another free parameter to be learnt. If this modification is not made then there is a tendency for some of the filters to become very small during learning. Once this has happened, it is difficult for them to grow again since the magnitude of the gradient *depends* on the filter output, which in turn depends on the filter length.

The first manipulation simply extends the power of the model, but one could argue that the second manipulation is something of a fudge — if we have sufficient data, a good model and a good algorithm, it should be unnecessary to restrict ourselves in this way. There are several counter arguments to this, the principal ones being: (i) we might be interested, from a biological point of view, in representational schemes in which the representational units all receive comparable amounts of input; (ii) we can view it as approximate posterior inference under a prior belief that in an effective model, all the units should play a roughly equal part in defining the density and forming the representation. We also note that a similar manipulation is also applied by most practitioners dealing with overcomplete ICA models (eg: Olshausen and Field (1996)).

In figure 3 (C) and (D) we show example filters typical of those learnt in overcomplete simulations. As in the complete case, we note that the majority of learnt filters qualitatively match the linear receptive fields of simple cells found in V1. Like V1 spatial receptive fields, most (although not all) of the learnt filters are well fit by Gabor functions. We analysed in more detail the properties of filter sets produced by different models by fitting a Gabor function to each filter (using a least squares procedure), and then looking at the population properties in terms of Gabor parameters.[12]

Figure 4 shows the distribution of parameters obtained by fitting Gabor functions

[11] Note that in an overcomplete setting, depending on the direction of the filters, α may be less than 1.5 and still yield a normalisable distribution overall.

[12] Approximately $5-10\%$ of the filters failed to localise well in orientation or location — usually appearing somewhat like noise or checkerboard patterns — and were not well described by a Gabor function. These were detected during the parametric fitting process and were eliminated from our subsequent population analyses. It is unclear exactly what role these filters play in defining densities within our model.

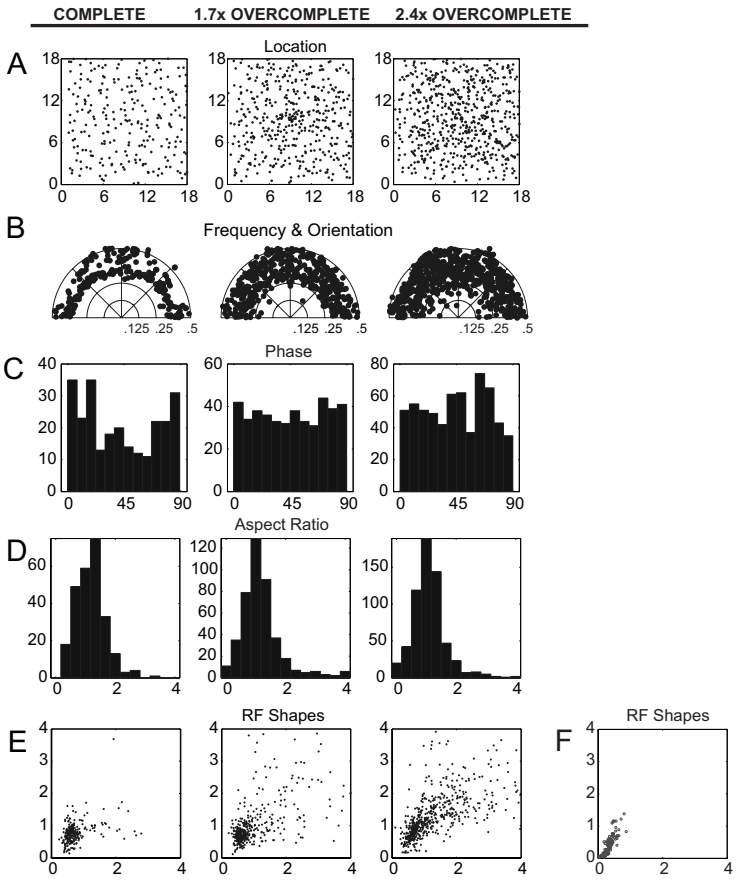

Figure 4: A summary of the distribution of some parameters derived by fitting Gabor functions to receptive fields of three models with different degrees of overcompleteness in the representation size. The leftmost column (A-E) is a complete representation, the middle column is $1.7\times$ overcomplete and the rightmost column is $2.4\times$ overcomplete. (A) Each dot represents the center location, in pixel coordinates within a patch, of a fitted Gabor. (B) Scatterplots showing the joint distribution of orientation (azimuthally) and spatial frequency in cycles per pixel (radially). (C) Histograms of Gabor fit phase (mapped to range $0°$–$90°$ since we ignore the envelope sign.) (D) Histograms of the aspect ratio of the Gabor envelope (Length/Width) (E) A plot of "normalized width" versus "normalized length", c.f. Ringach (2002). (F) For comparison, we include data from real macaque experiments Ringach (2002).

to complete and overcomplete filters. For reference, similar plots for linear spatial receptive fields measured *in vivo* are given in Ringach (2002) and van Hateren and van der Schaaf (1998).

The plots are all reasonable qualitative matches to those shown for the "real" V1 receptive fields as shown for instance in Ringach (2002). They also help to indicate the effects of representational overcompleteness. With increasing overcompleteness the coverage in the spaces of location, spatial frequency and orientation becomes denser and more uniform whilst at the same time the distribution of receptive fields shapes remains unchanged. Further, the more overcomplete models give better coverage in lower spatial frequencies that are not directly represented in complete models.

Ringach (2002) reports that the distribution of shapes from ICA/sparse coding can be a poor fit to the data from real cells — the main problem being that there are too few cells near the origin of the plot, which corresponds roughly to cells with smaller aspect ratios and small numbers of cycles in their receptive fields. The results which we present here appear to be a slightly better fit. (One source of the differences might be Ringach's choice of ICA prior.) A large proportion of our fitted receptive fields are in the vicinity of the macaque results, although as we become more overcomplete we see a spread further away from the origin.

In summary, our results from these single layer PoT models can account for many of the properties of simple cell linear spatial receptive fields in V1.

4.3 Hierarchical PoT Models

We now present results from the hierarchical extension of the basic PoT model. In principle we are able to learn both sets of weights, the top level connections $\mathbf{W}$ and the lower level connections $\mathbf{J}$, simultaneously. However, effective learning in this full system has proved difficult when starting from random initial conditions. The results which we present in this section were obtained by initialising $\mathbf{W}$ to the identity matrix and first learning $\mathbf{J}$, before subsequently releasing the $\mathbf{W}$ weights and then letting the system learn freely. This is therefore equivalent to initially training a single layer PoT and then subsequently introducing a second layer.

When models are trained in this way, the form of the first layer filters remains essentially unchanged from the Gabor receptive fields shown previously. Moreover, we see interesting structure being learnt in the $\mathbf{W}$ weights as illustrated by figure 5. The figure is organised to display the filters connected most strongly to a top layer unit. There is a strong organisation by what might be termed "themes" based upon location, orientation and spatial frequency. An intuition for this grouping behaviour is as follows: there will be correlations between the squared outputs of some pairs of filters, and by having them feed into the same top-level unit the model is able to capture this regularity. For most input images all members of the group will have small combined activity, but for a few images they will have significant combined activity. This is exactly what the

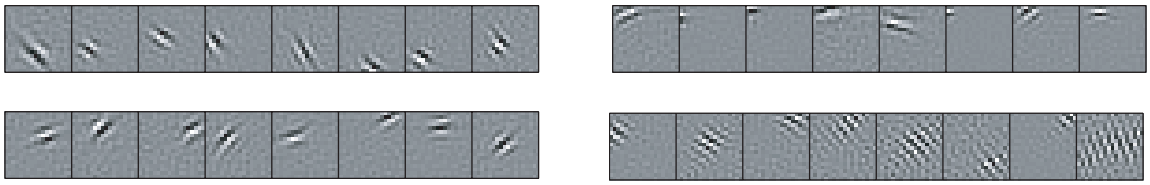

Figure 5: Each panel in this figure illustrates the "theme" represented by a different top level unit. The filters in each row are arranged in descending order, from left to right, of the strength W_{ij} with which they connect to the particular top layer unit.

energy function favours, as opposed to a grouping of very different filters which would lead to a rather Gaussian distribution of activity in the top layer.

Interestingly, these themes lead to responses in the top layer (if we examine the outputs $z_i = \mathbf{W}_i(\mathbf{J}\mathbf{x})^2$) that resemble complex cell receptive fields. It can be difficult to accurately describe the response of non-linear units in a network, and we choose a simplification in which we consider the response of the top layer units to test stimuli that are gratings or Gabor patches. The test stimuli were created by finding the grating or Gabor stimulus that was most effective at driving a unit and then perturbing various parameters about this maximum. Representative results from such a characterisation are shown are shown in figure 6.

In comparison to the first layer units, the top layer units are considerably more invariant to phase, and somewhat more invariant to position. However, both the sharpness of tuning to orientation and spatial frequency remain roughly unchanged. These results typify the properties that we see when we consider the responses of the second layer in our hierarchical model and are a striking match to the response properties of complex cells.

4.4 Topographic Hierarchical PoT Models

We next consider the topographically constrained form of the hierarchical PoT which we proposed in an attempt to induce spatial organisation upon the representations learnt. The $\mathbf{W}$ weights are fixed and define local, overlapping neighbourhoods on a square grid with toroidal boundary conditions. The $\mathbf{J}$ weights are free to learn, and the model is trained as usual.

Representative results from such a simulation are given in figure 7. The inputs were patches of size 25×25, whitened and dimensionality reduced to vectors of size 256; the representation is $1.7\times$ overcomplete. By simple inspection of the filters in figure 7 (A) we see that there is strong local continuity in the receptive field properties of orientation and spatial frequency and location, with little continuity of spatial phase.

With notable similarity to experimentally observed cortical topography, we see pinwheel singularities in the orientation map and a low frequency cluster in the spatial

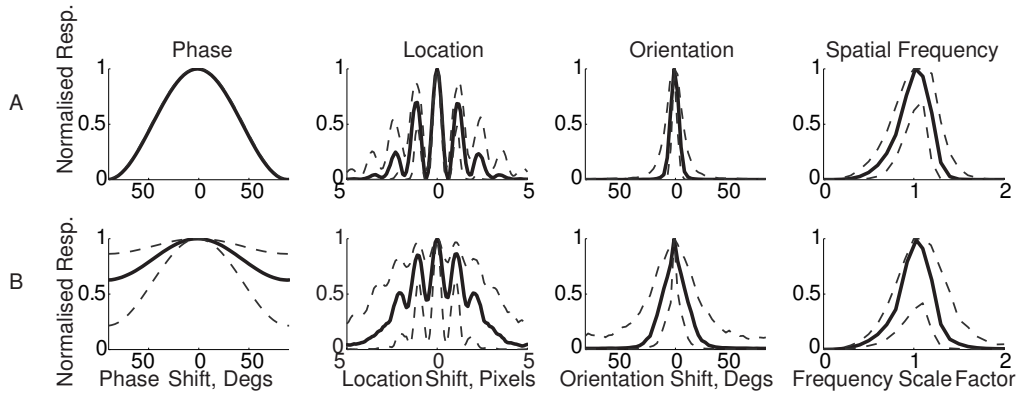

Figure 6: (A) Tuning curves for "simple cells", i.e. first layer units. (B) Tuning curves for "complex cells", i.e. second layer units. The tuning curves for Phase, Orientation and Spatial Frequency were obtained by probing responses using grating stimuli, the curve for location was obtained by probing using a localised Gabor patch stimulus. The optimal stimulus was estimated for each unit, and then one parameter (Phase, Location, Orientation or Spatial Frequency) was varied and the changes in responses were recorded. The response for each unit was normalized such that the maximum output was 1, before combining the data over the population. The solid line shows the population average (median of 441 units in a $1.7\times$ overcomplete model), whilst the lower and upper dotted lines show the 10% and 90% centiles respectively. We use a style of display as used in Hyvarinen et al. (2001)

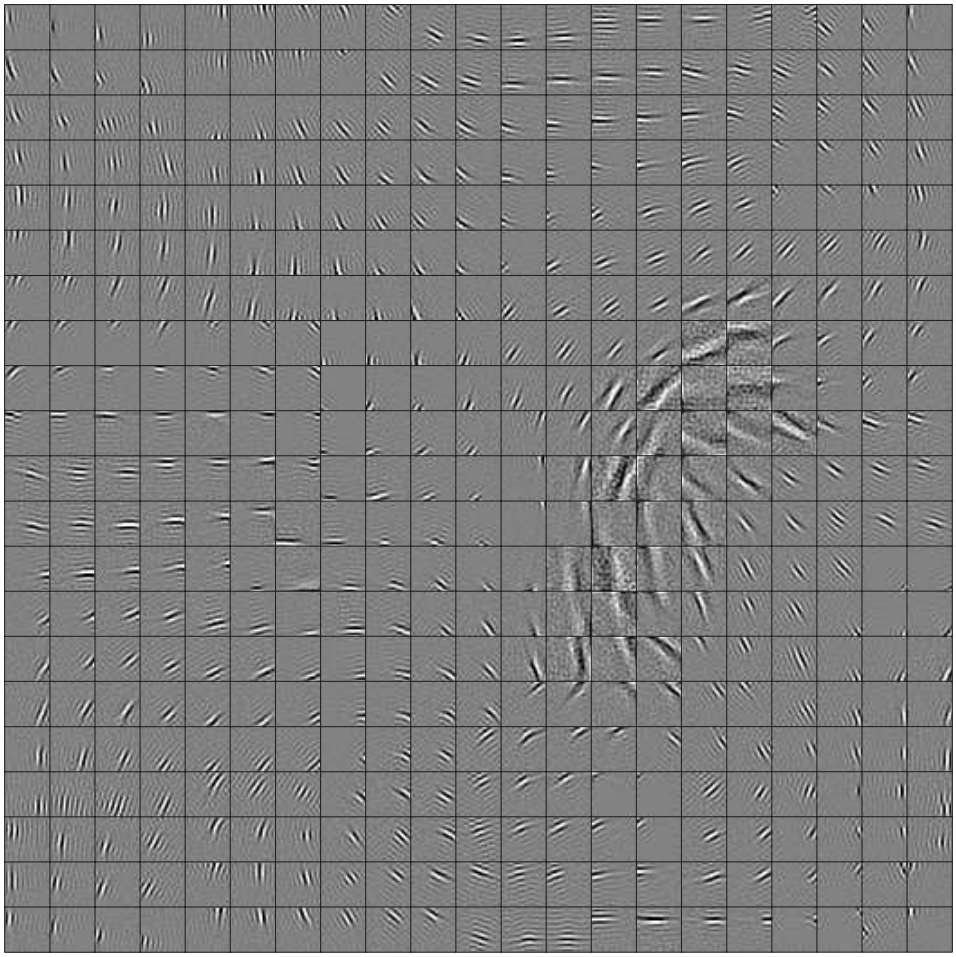

Figure 7: An example of a filter map. (The grayscale is saturating in each cell independently.) This model was trained on 25×25 patches that had been whitened and dimensionality reduced to 256 dimensions, and the representation layer is $1.7\times$overcomplete in terms of the inputs. The neighbourhood size was a 3×3 square (i.e. 8 nearest neighbours.) We see a topographically ordered array of learnt filters with local continuity of orientation, spacial frequency, and location. The local variations in phase seem to be random. Considering the map for orientation we see evidence for pinwheels, in the map for spacial frequency there is a distinct low-frequency cluster.

frequency map which seems to be somewhat aligned with one of the pinwheels. Whilst the map of location (retinotopy) shows good local structure, there is poor global structure. We suggest that this may be due to the relatively small scale of the model and the use of toroidal boundary conditions (which eliminated the need to deal with edge effects.)

5 Relation to Earlier Work

5.1 Gaussian Scale Mixtures

We can consider the complete version of our model as a Gaussian scale mixture (Andrews and Mallows, 1974; Wainwright and Simoncelli, 2000; Wainwright et al., 2000) with a particular (complicated) form of scaling function.[13]

The basic form for a GSM density on a variable, $\mathbf{g}$, can be given as follows (Wainwright and Simoncelli, 2000),

$$p_{\text{GSM}}(\mathbf{g}) = \int_{-\infty}^{\infty} \frac{1}{(2\pi)^{\frac{N}{2}} |c\mathbf{Q}|^{\frac{1}{2}}} \exp\left(-\frac{\mathbf{g}^T (c\mathbf{Q})^{-1} \mathbf{g}}{2}\right) \phi_c(c) dc \tag{22}$$

where c is a non-negative scalar variate and $\mathbf{Q}$ is a positive definite covariance matrix. This is the distribution that results if we draw c from $\phi_c(c)$ and a variable $\mathbf{v}$ from a multi-dimensional Gaussian $\mathcal{N}_{\mathbf{V}}(0, \mathbf{Q})$ and then take $\mathbf{g} = \sqrt{c}\mathbf{v}$.

Wainwright et al. (2000) discuss a more sophisticated model in which the distributions of coefficients in a wavelet decomposition for images are described by a GSM which has a separate scaling variable, c_i, for each coefficient. The c_i have a Markov dependency structure based on the multi-resolution tree which underlies the wavelet decomposition.

In the complete setting, where the $\mathbf{y}$ variables are in linear one-to-one correspondence with the input variables, $\mathbf{x}$, we can interpret the distribution $p(\mathbf{y})$ as a Gaussian scale mixture. To see this we first rewrite $p(\mathbf{y}, \mathbf{u}) = p(\mathbf{y}|\mathbf{u})p(\mathbf{u})$, where the conditional $p(\mathbf{y}|\mathbf{u}) = \prod_j \mathcal{N}_{y_j}[0, (\sum_i W_{ij} u_i)^{-1}]$ is Gaussian (see Eqn.14). The distribution $p(\mathbf{u})$ needs to be computed by marginalizing $p(\mathbf{x}, \mathbf{u})$ in Eqn. 12 over $\mathbf{x}$ resulting in,

$$p(\mathbf{u}) = \frac{1}{Z_u} \prod_i e^{-u_i} {u_i}^{\alpha_i - 1} \prod_k \left(\sum_j W_{jk} u_j\right)^{-\frac{1}{2}} \tag{23}$$

where the partition function Z_u ensures normalisation. We see that the marginal dis-

[13] In simple terms a GSM density is one that can be written as a (possibly infinite) mixture of Gaussians that differ only in the scale of their covariance structure. A wide range of distributions can be expressed in this manner.

tribution of each y_i is a Gaussian scale mixture in which the scaling variate for y_i is given by $c_i(\mathbf{u}) = (\sum_j W_{ji} u_j)^{-1}$. The neighbourhoods defined by $\mathbf{W}$ in our model play an analogous role to the tree-structured cascade process in Wainwright et al. (2000), and determine the correlations between the different scaling coefficients. However, a notable difference in this respect is that Wainwright et al. (2000) assume a fixed tree structure for the dependencies whereas our model is more flexible in that the interactions through the $\mathbf{W}$ parameters can be learned.

The overcomplete version of our PoT is not so easily interpreted as a GSM because the $\{y_i\}$ are no longer independent given $\mathbf{u}$, nor is the distribution over $\mathbf{x}$ a simple GSM due to the way in which $\mathbf{u}$ is incorporated into the covariance matrix (see equation 9). However, much of the flavour of a GSM remains.

5.2 Relationship to tICA

In this section we show that, in the complete case, the topographic PoT model is isomorphic to the model optimised (but not the one initially proposed) by Hyvarinen et al. (2001) in their work on topographic ICA (tICA). These authors define an ICA generative model in which the components/sources are not completely independent but have a dependency that is defined with relation to some topology, such as a toroidal grid — components close to one another in this topology have greater co-dependence than those that are distantly separated.

Their generative model is shown schematically in figure 8. The first layer takes a linear combination of "variance-generating" variables, $\mathbf{t}$, and then passes them through some non-linearity, $\phi(\cdot)$, to give positive scaling variates, σ. These are then used to set the variance of the sources, $\mathbf{s}$, and conditioned on these scaling variates, the components in the second layer are independent. These sources are then linearly mixed to give the observables, $\mathbf{x}$.

The joint density for $(\mathbf{s}, \mathbf{t})$ is given by

$$p(\mathbf{s}, \mathbf{t}) = \prod_i p_{s_i}\left(\frac{s_i}{\phi\left(\mathbf{H}_i^T \mathbf{t}\right)}\right) \frac{p_{t_i}(t_i)}{\phi\left(\mathbf{H}_i^T \mathbf{t}\right)} \tag{24}$$

and the log-likelihood of the data given the parameters is

$$\mathcal{L}(\mathbf{B}) = \sum_{\text{data } \mathbf{x}} \int \prod_i p_{s_i}\left(\frac{\mathbf{B}_i^T \mathbf{x}}{\phi\left(\mathbf{H}_i^T \mathbf{t}\right)}\right) \frac{p_{t_i}(t_i)}{\phi\left(\mathbf{H}_i^T \mathbf{t}\right)} |\det \mathbf{B}| d\mathbf{t} \tag{25}$$

where $\mathbf{B} = \mathbf{A}^{-1}$.

As noted in their paper, this likelihood is intractable to compute because of the integral over possible states of $\mathbf{t}$. This prompts the authors to derive an approach that makes various simplifications and approximations to give a lower bound on the likelihood.

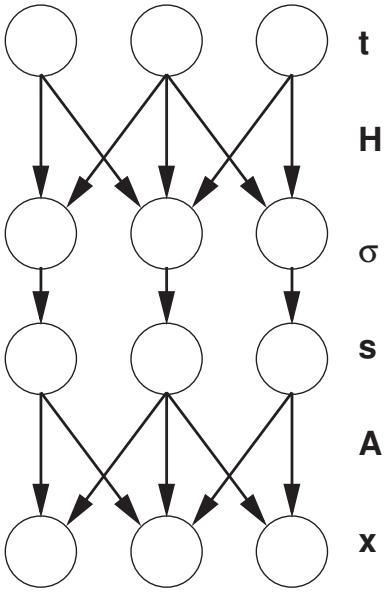

Figure 8: Graphical model for topographic ICA (Hyvarinen et al., 2001). First the variance "generating variables", t_i, are generated independently from their prior. They are then linearly mixed through the matrix $\mathbf{H}$, before being non-linearly transformed using function $\phi(\cdot)$ to give the variances, $\sigma_i = \phi(\mathbf{H}_i^T \mathbf{t})$, for each of the sources, i. Values for these sources, s_i, are then generated from independent zero mean distributions with variances σ_i, before being linearly mixed through matrix $\mathbf{A}$ to give observables x_i.

Firstly, they restrict the form of the base density for s to be gaussian[14], both $\mathbf{t}$ and $\mathbf{H}$ are constrained to be non-negative, and $\phi(\cdot)$ is taken to be $(\cdot)^{-\frac{1}{2}}$. This yields the following expression for the marginal density of $\mathbf{s}$,

$$p(\mathbf{s}) = \int \frac{1}{(2\pi)^{\frac{d}{2}}} \exp\left(-\frac{1}{2}\sum_k t_k \left[\sum_i H_{ik} s_i^2\right]\right) \prod_i p_{t_i}(t_i)\sqrt{\mathbf{H}_i^T \mathbf{t}} d\mathbf{t} \tag{26}$$

This expression is then simplified by the approximation,

$$\sqrt{\mathbf{H}_i^T \mathbf{t}} \approx \sqrt{H_{ii} t_i} \tag{27}$$

Whilst this approximation may not always be a good one, it is a strict lower bound on the true quantity and thus allows for a lower bound on the likelihood as well. Their final approximate likelihood objective, $\widetilde{\mathcal{L}(\mathbf{B})}$, is then given by,

$$\widetilde{\mathcal{L}(\mathbf{B})} = \sum_{data}\left(\sum_{j=1}^{d} G\left(\sum_{i=1}^{d} H_{ij}(\mathbf{B}_i^T \mathbf{x})^2\right) + \log|\det(\mathbf{B})|\right) \tag{28}$$

where the form of the scalar function G is given by,

$$G(\tau) = \log \int \frac{1}{\sqrt{2\pi}} \exp\left(\frac{1}{2} t\tau\right) p_t(t)\sqrt{H_{ii}} dt \tag{29}$$

The results obtained by Hyvarinen and Hoyer (2001); Hyvarinen et al. (2001) are very similar to those presented here in section 4. These authors also noted the similarity between elements of their model and the response properties of simple and complex cells in V1.

Interestingly, the optimisation problem that they *actually* solve (i.e. maximisation of equation 28), rather than the one they originally propose, can be mapped directly onto the optimisation problem for a square, topographic PoT model if we take: $\mathbf{B} \equiv \mathbf{J}^{\text{PoT}}$, $\mathbf{H} \equiv \mathbf{W}^{\text{PoT}}$ and $G(\tau) = \log(1 + \frac{1}{2}\tau)$. More generally, we can construct an equivalent, square energy-based model whose likelihood optimisation corresponds exactly to the optimisation of their "approximate" objective function. In this sense, we feel that our perspective has some advantages. Firstly, in that we have a more accurate picture of what model we *actually* (try to) optimise. Secondly, in that we are able to move more easily to overcomplete representations. If Hyvarinen et al. (2001) were to make their model overcomplete there would no longer be a deterministic relationship between their sources $\mathbf{s}$ and $\mathbf{x}$ — this additional complication would make the already difficult prob-

[14]Their model can therefore be considered as type of GSM, although the authors do not comment on this.

lems of inference and learning significantly harder. Thirdly, in the HPoT framework we are able to learn the top-level weights $\mathbf{W}$ in a principled way using the techniques discussed in section 3.2, whereas current tICA approaches have treated only fixed local connectivity.

5.3 Relationship to other ICA extensions

Karklin and Lewicki (2003, 2005) also propose a hierarchical extension to ICA that involves a second hidden layer of marginally independent sparsely active units. Their model is of the general form proposed in Hyvarinen et al. (2001) but uses a different functional dependency between the first and second hidden layers to that employed in the topographic ICA model which Hyvarinen et al. (2001) fully develop.

In the generative pass from Karklin and Lewicki's model, linear combinations of the second layer activities are fed through an exponential function to specify scaling or variance parameters for the first hidden layer. Conditioned upon these variances, the units in the first hidden layer are independent and behave like the hidden variables in a standard ICA model. This model can be described by the graph in figure 8 where the transfer function $\phi(\cdot)$ is given by an exponential. Using the notation of this figure, the relevant distributions are ,

$$p(t_i) = \frac{q_i}{2\Gamma(q_i^{-1})} \exp\left(|t_i|^{q_i}\right) \tag{30}$$

$$\sigma_j = ce^{[\mathbf{Ht}]_j} \tag{31}$$

$$p(s_j|\sigma_j) = \frac{q_j}{2\sigma_j\Gamma(q_j^{-1})} \exp\left(\left|\frac{s_j}{\sigma_j}\right|^{q_j}\right) \tag{32}$$

$$x_k = [\mathbf{As}]_k \tag{33}$$

The authors have so far only considered complete models and in this case, as with tICA, the first layer of hidden variables are deterministically related to the observables.[15]

To link this model to our energy-based PoT framework first we consider the following change of variables,

$$\mathbf{B} = \mathbf{A}^{-1} \tag{34}$$

$$\mathbf{K} = \mathbf{H}^{-1} \tag{35}$$

$$\nu_j = \sigma_j^{-1} \tag{36}$$

Then, considering the q variables to be fixed, can write the energy function of their

[15]Furthermore, as well as focussing their attention on the complete case, the authors assume the first level weights are fixed to a set of filters obtained using regular ICA.

model as

$$E(\mathbf{x}, \boldsymbol{\nu}) = \sum_i \left| \sum_k K_{ik} \log(c\nu_k) \right|^{q_i} + \sum_j \left(\log \nu_j + |\nu_j|^{q_j} |\left[\mathbf{B}\mathbf{x}\right]_j|^{q_j} \right) \qquad (37)$$

I.e. when we take the Boltzmann distribution with the energies defined in equation 37 we recover the joints and marginals specified by Karklin and Lewicki (2003, 2005).

Whilst the overall models are different, there are some similarities between this formulation and the auxiliary variable formulation of extended HPoT models (i.e. equation 12 with generalised exponent β from section 2.5). Viewed from an energy based perspective, they both have the property that an energy 'penalty' is applied to (a magnitude function of) a linear filtering of the data. The 'scale' of this energy penalty is given by a supplementary set of random variables which themselves are subject to an additional energy function.

As with standard ICA, in overcomplete extensions of this model the similarities to an energy based perspective would be further reduced. We note as an aside that it might be interesting to consider the 'energy-based' overcomplete extension of Karklin and Lewicki's model, in addition to the standard causal overcomplete extension. In the overcomplete version of the causal model inference would likely be much more difficult because of posterior dependencies both within and between the two hidden layers. For the overcomplete energy-based model, the necessary energy function appears not to be amenable to efficient Gibbs sampling, but parameters could still be learned using Contrastive Divergence and Monte Carlo methods such as Hybrid Monte Carlo.

5.4 Representational differences between causal models and energy-based models

As well as specifying different probabilistic models, overcomplete energy-based models (EBM's) such as the PoT differ from overcomplete causal models in the types of representation they (implicitly) entail. This has interesting consequences when we consider the "population codes" suggested by the two types of model. We focus on the representation in the first layer ("simple cells"), although similar arguments might be for deeper layers as well.

In an overcomplete causal model, many configurations of the sources are compatible with a configuration of the input.[16] For a given input, a posterior distribution is induced over the sources in which the inferred values for different sources are conditionally dependent. As a result, even for models which are linear in the generative direction, the formation of a posterior representation in overcomplete causal models is essentially *non-linear* and moreover it is *non-local* due to the lack of conditional in-

[16] In fact, strictly speaking there is a subspace of compatible source configurations.

dependence. This implies that unlike EBM's inference in overcomplete causal models is typically iterative, often intractable, and therefore time consuming. Also, although we can specify the basis functions associated with a unit, it is much harder to specify any kind of feed-forward receptive field in causal models. The issue of how such a posterior distribution could be encoded in a representation remains open; a common postulate (made on the grounds of efficient coding) is that a maximum a posteriori (MAP) representation should be used, but we note that even computing the MAP value is usually iterative and slow.

Conversely, in overcomplete EBM's with deterministic hidden units such as we have presented in this paper, the mapping from inputs to representations remains simple and non-iterative and requires only local information.

In figure 9 we use a somewhat absurd example to schematically illustrate a salient consequence of this difference between EBM's and causal models that have sparse priors. The array of vectors in figure 9 (A) should be understood to be either a subset of the *basis functions* in an overcomplete causal model, or a subset of the *filters* in overcomplete PoT model. In panel (B) we show four example input images. These have been chosen to be identical to four of the vectors shown in panel (A). The left-hand column of panel (C) shows the representation responses of the units in an EBM-style model for these four inputs; the right-hand column shows the MAP responses from an overcomplete causal model with a sparse source prior.

This is admittedly an extreme case, but it provides a good illustration of the point we wish to make. More generally, although representations in an overcomplete PoT are sparse there is also some redundancy; the PoT population response is typically less sparse (Willmore and Tolhurst, 2001) than an causal model with an "equivalent" prior.

Interpreting the two models as a description of neural coding, one might expect the EBM representation to be more robust to the influences of neural noise as compared with the representation suggested from a causal approach. Furthermore, the EBM style representation is shiftable — it has the property that for small changes in the input there are small changes in the representation. This property would not necessarily hold for a highly overcomplete causal model. Such a discontinuous representation might make subsequent computations difficult and non-robust, and it also seems somewhat at odds with the neurobiological data — however proper comparison is difficult since there is no real account of dynamic stimuli or spiking in either model. At present, it remains unclear which type of model — causal or energy-based — provides the more appropriate description of coding in the visual system, especially since there are many aspects that neither approach captures.

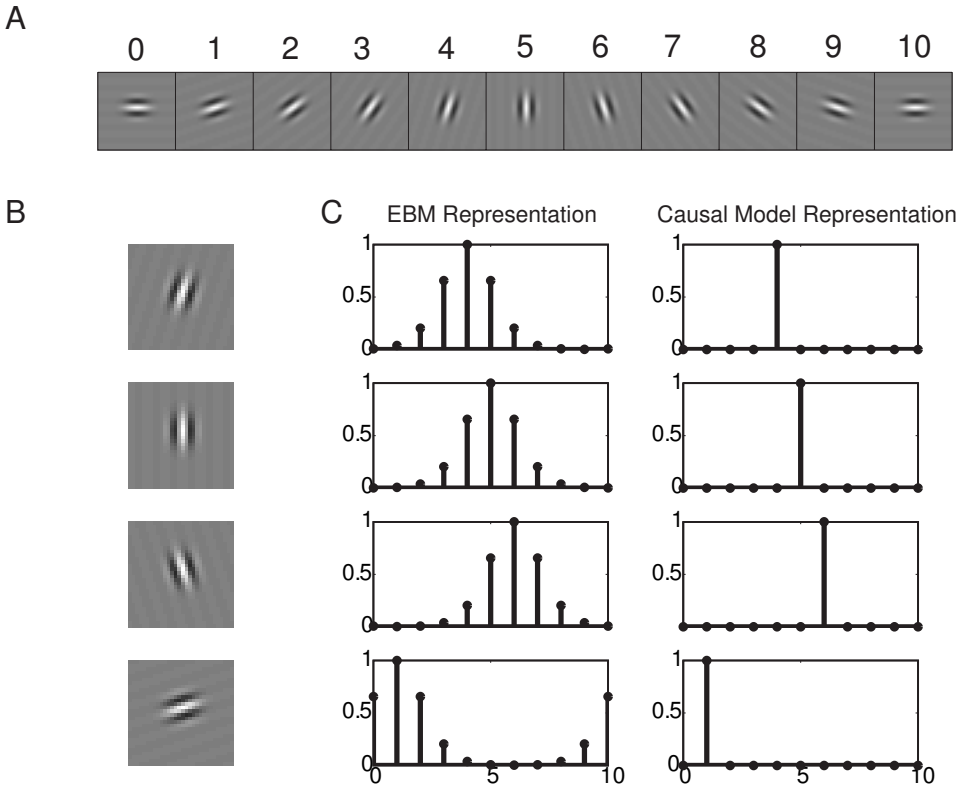

Figure 9: Representational differences between overcomplete causal models and overcomplete deterministic EBM's. (A) The 11 vectors in this panel should be considered as the vectors associated with a subset of representational units in either an overcomplete EBM or an overcomplete causal model. In the EBM they would be the feed-forward filter vectors; in the causal model they would be basis functions. (B) Probe stimuli — these images exactly match the vectors as those associated with units 4,5,6, & 1. (C) The left-hand column shows the normalized responses in an EBM model of the 11 units assuming they are filters. The right-hand column shows the normalized response from the units assuming that they are basis functions in a causal model with a sparse prior, and that we have formed a representation by taking the MAP configuration for the source units.

6 Summary

We have presented a hierarchical energy-based density model that we suggest is generally applicable to data-sets that have a sparse structure, or that can be well characterised by constraints that are often well-satisfied, but occasionally violated by a large amount.

By applying our model to natural scene images we are able to provide an interpretational account for many aspects of receptive field and topographic map structure within primary visual cortex, and which also develops sensible high-dimensional population codes. Deterministic features (i.e. the first- and second-layer filter outputs) within our model play a key role in defining the density of a given image patch, and we are able to make a close relationship between these features and the responses of simple cells and complex cells in V1. Furthermore, by constraining our model to interact locally we are able to provide some computational motivation for the forms of the cortical maps for retinotopy, phase, spatial frequency and orientation.

Whilst our model is closely related to some previous work, most prominently Hyvarinen et al. (2001), it bestows a different interpretation on the learnt features, is different in its formulation and describes rather different statistical relations in the overcomplete case.

We present our model as both a general alternative tool to ICA for describing sparse data distributions and also as an alternative interpretive account for some of the neurobiological observations from the mammalian visual system. Finally we suggest that the models outlined here could be used as a starting point for image processing applications such as denoising or deblurring, and that it might also be adapted to time series data such as natural audio sequences.

Acknowledgements

We thank Peter Dayan and Yee Whye Teh for important intellectual contributions to this work and many other researchers for helpful discussions. The work was funded by the Gatsby Charitable Foundation, the Wellcome Trust, NSERC, CFI and OIT. GEH holds a Canada Research Chair.

Appendices

A Sampling in HPoT models

Complete Models

We start our discussion with sampling in complete HPoT models. In this case there is a simple invertible relationship between $\mathbf{x}$ and $\mathbf{y}$, implying that we may focus on sampling $\mathbf{y}$ and subsequently transforming these samples back to $\mathbf{x}$-space through $\mathbf{x} = \mathbf{J}^{-1}\mathbf{y}$. Unfortunately, unless $\mathbf{W}$ is diagonal, all $\mathbf{y}$ variables are coupled through $\mathbf{W}$, which makes it difficult to devise an exact sampling procedure. Hence, we resort to Gibbs sampling using Eqn.13 where we replace $y_j = \mathbf{J}_j\mathbf{x}$ to acquire sample $\mathbf{u}|\mathbf{y}$. To obtain a sample $\mathbf{y}|\mathbf{u}$ we convert Eqn.9 into

$$P(\mathbf{y}|\mathbf{u}) = \mathcal{N}_{\mathbf{y}}\left[\mathbf{y};\ \mathbf{0}, \mathbf{Diag}[\mathbf{W}^T\mathbf{u}]^{-1}\right] \tag{38}$$

We iterate this process (alternatingly sampling $\mathbf{u} \sim P(\mathbf{u}|\mathbf{y})$ and $\mathbf{y} \sim P(\mathbf{y}|\mathbf{u})$) until the Gibbs sampler has converged. Note that both $P(\mathbf{u}|\mathbf{y})$ and $P(\mathbf{y}|\mathbf{u})$ are factorized distributions implying that both $\mathbf{u}$ and $\mathbf{y}$ variables can be sampled in parallel.

Overcomplete Models

In the overcomplete case we are no longer allowed to first sample the $\mathbf{y}$ variables, and subsequently transform them into $\mathbf{x}$ space. The reason is that the deterministic relation $\mathbf{y} = \mathbf{J}\mathbf{x}$ means that when there are more $\mathbf{y}$ variables than $\mathbf{x}$ variables, some $\mathbf{y}$ configurations are not allowed, i.e. they are not in the range of the mapping $\mathbf{x} \rightarrow \mathbf{J}\mathbf{x}$ with $\mathbf{x} \in \mathbb{R}$. If we sample $\mathbf{y}$, all these samples (with probability one) will have some components in these forbidden dimensions, and it is unclear how to transform them correctly into $\mathbf{x}$-space. An approximation is obtained by projecting the $\mathbf{y}$-samples into $\mathbf{x}$-space using $\tilde{\mathbf{x}} = \mathbf{J}^{\#}\mathbf{y}$. We have often used this approximation in our experiments and have obtained good results, but we note that its accuracy is expected to decrease as we increase the degree of overcompleteness.

A more expensive but correct sampling procedure for the overcomplete case is to use a Gibbs chain in the variables $\mathbf{u}$ and $\mathbf{x}$ (instead of $\mathbf{u}$ and $\mathbf{y}$) by using Eqns.13 and 14 directly. In order to sample $\mathbf{x}|\mathbf{u}$ we need to compute a Cholesky factorization of the inverse-covariance matrix of the Gaussian distribution $P(\mathbf{x}|\mathbf{u})$,

$$\mathbf{R}^T\mathbf{R} = \mathbf{J}\mathbf{V}\mathbf{J}^T \qquad \mathbf{V} = \mathbf{Diag}[\mathbf{W}^T\mathbf{u}] \tag{39}$$

The samples $\mathbf{x}|\mathbf{u}$ are now obtained by first sampling from a multivariate standard normal distribution, $\mathbf{n} \sim \mathcal{N}_{\mathbf{n}}[\mathbf{n}; \mathbf{0}, \mathbf{I}]$, and subsequently setting: $\mathbf{x} = \mathbf{R}^{-1}\mathbf{n}$. The reason this procedure is expensive is that $\mathbf{R}$ depends on $\mathbf{u}$ which changes at each iteration.

Hence, the expensive Cholesky factorization and inverse have to be computed at each iteration of Gibbs sampling.

Extended PoT Models

The sampling procedures for the complete and undercomplete *extended* models discussed in section 2.5 are very similar, apart from the fact that the conditional distribution $P(\mathbf{y}|\mathbf{u})$ is now given by,

$$P_{ext}(\mathbf{x}|\mathbf{u}) \propto \prod_{i=1}^{M} \exp(-\frac{1}{2}\mathbf{V}_{ii}|y_i|^{\beta}) \qquad \mathbf{V} = \mathbf{Diag}[\mathbf{W}^T\mathbf{u}] \tag{40}$$

Efficient sampling procedures exist for this generalized Gaussian-Laplace probability distribution. In the overcomplete case it has proven more difficult to devise an efficient Gibbs chain (the Cholesky factorization is no longer applicable), but the approximate projection method using the pseudo-inverse, $\mathbf{J}^{\#}$ still seems to work well.

References

Andrews, D. and Mallows, C. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society*, 36:99–102.

Atick, J. J. and Redlich, A. N. (1992). What does the retina know about natural scenes. *Neural Computation*, 4(2):196–210.

Bell, A. J. and Sejnowski, T. J. (1995). An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.

Bell, A. J. and Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338.

Carreira-Perpinan, M. and Hinton, G. (2005). On contrastive divergence learning. In *Artificial Intelligence and Statistics*.

Freund, Y. and Haussler, D. (1992). Unsupervised learning of distributions of binary vectors using 2-layer networks. In *Advances in Neural Information Processing Systems*, volume 4, pages 912–919.

Heskes, T. (1998). Selecting weighting factors in logarithmic opinion pools. In Jordan, M. I., Kearns, M. J., and Solla, S. A., editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

Hinton, G. and Teh, Y. (2001). Discovering multiple constraints that are frequently approximately satisfied. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 227–234, Seattle, Washington.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Hoyer, P. O. and Hyvarinen, A. (2000). Independent component analysis applied to feature extraction from colour and stereo images. *Network-Computation in Neural Systems*, 11(3):191–210.

Hyvarinen, A. and Hoyer, P. O. (2001). A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41(18):2413–2423.

Hyvarinen, A., Hoyer, P. O., and Inki, M. (2001). Topographic independent component analysis. *Neural Computation*, 13(7):1527–1558.

Karklin, Y. and Lewicki, M. S. (2003). Learning higher-order structures in natural images. *Network-Computation in Neural Systems*, 14:483–399.

Karklin, Y. and Lewicki, M. S. (2005). A hierarchical bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. *Neural Computation*, 17:397–423.

Lewicki, M. and Sejnowski, T. (2000). Learning overcomplete representations. *Neural Computation*, 12:p.337–365.

Marks, T. K. and Movellan, J. R. (2001). Diffusion networks, products of experts, and factor analysis. Technical Report UCSD MPLab TR 2001.02, University of California San Diego.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–610.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311–3325.

Portilla, J., Strela, V., Wainwright, M., and Simoncelli, E. P. (2003). Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans Image Processing*, 12(11):1338–1351.

Ringach, D. L. (2002). Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *J Neurophysiol*, 88(1):455–63.

Simoncelli, E. (1997). Statistical models for images: Compression, restoration and synthesis. In *31st Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA.

Smolensky, P. (1986). Information processing in dynamical systems: foundations of harmony theory. In Rumehart, D. and McClelland, J., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. McGraw-Hill, New York.

Teh, Y., Welling, M., Osindero, S., and Hinton, G. (2003). Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research - Special Issue on ICA*, 4:1235–1260.

van Hateren, J. H. and van der Schaaf, A. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc R Soc Lond B Biol Sci*, 265(1394):359–66.

Wainwright, M. J. and Simoncelli, E. P. (2000). Scale mixtures of gaussians and the statistics of natural images. In *Advances in Neural Information Processing Systems 12*, volume 12, pages 855–861.

Wainwright, M. J., Simoncelli, E. P., and Willsky, A. S. (2000). Random cascades of gaussian scale mixtures and their use in modeling natural images with application to denoising. In *7th International Conference on Image Processing*, Vancouver, BC, Canada. IEEE Computer Society.

Welling, M., Agakov, F., and Williams, C. (2003a). Extreme components analysis. In *Advances in Neural Information Processing Systems*, volume 16, Vancouver, Canada.

Welling, M., Hinton, G., and Osindero, S. (2002a). Learning sparse topographic representations with products of student-t distributions. In *Advances in Neural Information Processing Systems*, volume 15, Vancouver, Canada.

Welling, M., Zemel, R., and Hinton, G. (2002b). Self-supervised boosting. In *Advances in Neural Information Processing Systems*, volume 15, Vancouver, Canada.

Welling, M., Zemel, R., and Hinton, G. (2003b). A tractable probabilistic model for projection pursuit. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Welling, M., Zemel, R., and Hinton, G. (2004). Probabilistic sequential independent components analysis. *IEEE-Transactions in Neural Networks , Special Issue on Information Theory*.

Williams, C., Agakov, F., and Felderhof, S. (2001). Products of gaussians. In *Advances in Neural Information Processing Systems*, volume 14, Vancouver, CA.

Williams, C. K. I. and Agakov, F. (2002). An analysis of contrastive divergence learning in gaussian boltzmann machines. Technical Report EDI-INF-RR-0120, School of Informatives.

Willmore, B. and Tolhurst, D. J. (2001). Characterizing the sparseness of neural codes. *Network-Computation in Neural Systems*, 12(3):255–270.

Yuille, A. (2004). A comment on contrastive divergence. Technical report, Department Statistics and Psychology UCLA. Technical Report.

Zhu, S. C., Wu, Y. N., and Mumford, D. (1998). Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126.