

---

# Complex-Cell Models

---

**Yutian Chen**

Department of Computer Science

## Abstract

A Product of Experts (PoE) model is a probabilistic model which combines a number of individual component models by multiplying their probabilities. A typical PoE model is an exponential family Harmonious introduced by Welling et al. In this paper, we apply a hierarchical PoE model, Complex-Cell Model (CCM), to image retrieval and compare the performance with a kind of standard harmonious model, Simple-Cell Model (SCM). It models second order correlations between pixels. We find that the precision of CCM is higher than SCM when the recall is large.

## 1 Introduction

A Product of Experts (PoE) model [1] combines a number of individual component models by taking their product and normalizing the result. The probabilistic model can be expressed as

$$P(x|\{\theta_j\}) = \frac{1}{Z} \prod_{j=1}^M f_j(x|\theta_j)$$

with

$$Z = \int dx \prod_{j=1}^M f_j(x|\theta_j)$$

PoE has the advantage of modeling constraints of the data compared to mixture of models. A kind of PoE model is introduced by Welling et al, aka Exponential Family Harmonious (EFH)[2]. It can be understood as a two layers Markov Random Field shown in Figure 1. One layer is observed variables  $x$  and the other is hidden variables  $h$ . The conditional probabilities  $P(h|x)$  and  $P(x|h)$  both belong to exponential family. EFH has much faster inference speed than directed graphical models since hidden variables are conditional independent given observed data, while in directed graphical models hidden variable are conditional dependent duo to explaining away property. When applied to document retrieval and object recognition [2] [3], EFH is proved to have better performance than directed graphical models such as pLSI and LSI.

A hierarchical EFH model, called hierarchical Product of Student-t model (hPoT) [4], is introduced to analyze natural scenes. It adds another layer on top of a PoE model whose marginal probability of  $x$  in each component is student-T distribution. This model shows topographic organization of Gabor-like receptive fields.

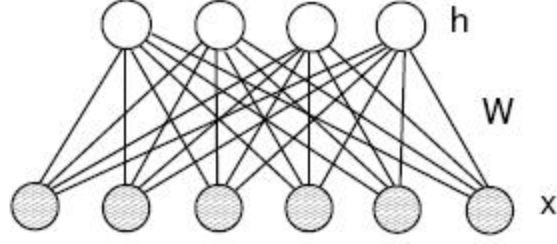


Figure 1: Exponential Family Harmonious

In this paper, We'd like to use a simplified version of hierarchical EFH model. It's called Complex-Cell Model (CCM) because its filters show similar features with complex cells in the cerebral primary visual cortex. The observed variables and hidden variables are both logical variables,  $\{0, 1\}$ . This model is trained with unlabeled digits by minimizing contrastive divergence. Then, we apply it to digit retrieval as well k-nearest neighbour classification and compare the precision-recall curve with a Simple-Cell Model (SCM) which is a standard two layers harmonious. The name also comes from a type of cells in primary visual cortex. It turns out that CCMs have Gabor-like filters modeling edges of digits and the precision is larger than SCM when recall is large.

## 2 Simple-Cell Models and Complex-Cell Models

SCMs and CCMs are both harmonium models proposed to model the priors of unlabeled data such as images and documents. We will introduce the probabilities and corresponding undirected graphs of these two models respectively in the following subsections.

### 2.1 Simple-Cell Models

The joint probability of observed and hidden variables in SCM is

$$P(x, h) = \frac{1}{Z} e^{-E(x, h)} \quad (1)$$

$$-E(x, h) = \sum_i \alpha_i x_i + \sum_j \beta_j h_j + \sum_{ij} h_j W_{ij} x_i \quad (2)$$

$$Z = \sum_{x, h} e^{(-E(x, h))} \quad (3)$$

This is a restricted Boltzmann machine with energy  $E(x, h)$ . The Z is a normalization term called partition function. The domain of observed and hidden variables is both  $\{0, 1\}$ . The conditional probabilities of  $P(x|h)$  and  $P(h|x)$  are both product of logistic functions

$$P(x|h) = \prod_i \sigma \left( (\alpha_i + \sum_j W_{ij} h_j) x_i \right) \quad (4)$$

$$P(h|x) = \prod_j \sigma \left( (\beta_j + \sum_i W_{ij} x_i) h_j \right) \quad (5)$$

$$\text{where } \sigma(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

The mean value of  $h$  is a function of the output of a linear filter  $W_j$ . This model corresponds to a two layers Markov model (Figure 1). Observed and hidden variables are conditional

independent given the other layer. This is a simplified version of EFH introduced in [2] and [3] where the conditional probabilities are more complex exponential family. It's more obvious to see that a SCM is a PoE when we marginalize hidden variables.

$$P(x) = \frac{1}{Z'} \prod_j (1 + \exp(\beta_j + \sum_i W_{ij}x_i)) \exp(\alpha^T x) \quad (7)$$

$$= \frac{1}{Z'} \exp \left[ \sum_j \log \left( 1 + \exp(\beta_j + \sum_i W_{ij}x_i) \right) + \alpha^T x \right] \quad (8)$$

where  $Z'$  is partition function.

## 2.2 Complex-Cell Model

A hierarchical PoE is introduced in [4]. The undirected model is shown in Figure 2. It adds a second layer on top of PoE including a nonlinear transition  $y \implies y^2$ . In this paper, instead of using student-t distribution, we use a binary model.

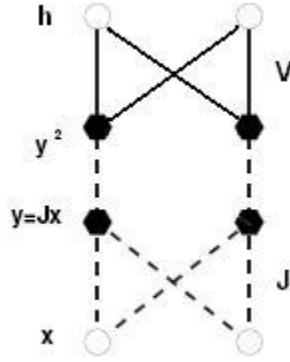


Figure 2: hPoE. Dash line means deterministic function

The energy of CCM model is

$$-E(x, h) = \sum_i \alpha_i x_i + \sum_k \gamma_k h_k + \sum_{kj} h_k V_{jk} \left( \sum_i J_{ji} x_i \right)^2 \quad (9)$$

The hidden variables  $h$  are still conditional independent given the observed variables  $x$ , but  $x$  is no longer conditional independent because of the nonlinear operation.

$$P(x_i | x_{-i}, h) = \sigma \left( \left[ \alpha_i + \sum_{kj} h_k V_{jk} J_{ji} \left( 2 \sum_l J_{j,l \neq i} x_l + J_{ji} \right) \right] x_i \right) \quad (10)$$

$$P(h|x) = \prod_k \sigma \left( \left[ \gamma_k + \sum_j V_{jk} \left( \sum_i J_{ji} x_i \right)^2 \right] h_k \right) \quad (11)$$

CCM puts constraints on second order relationship between observed variables. Given different  $h$ , it gives high probability to  $x$  with different covariance matrix.

### 3 Training Algorithm

Parameter learning for harmonium models is performed by stochastic gradient ascent on the log-likelihood of the data [2]. For large redundant dataset it is more efficient to estimate the required gradients on small batches rather than on the entire dataset. We also include a momentum term to speed up convergence and a decay parameter to reduce unneeded weights.

The derivatives of log-likelihood wrt. parameter  $w$  is in the form

$$\frac{\partial \log(P(x, h))}{\partial w} \propto \langle \frac{-\partial E(x, h)}{\partial w} \rangle_{\bar{p}} - \langle \frac{-\partial E(x, h)}{\partial w} \rangle_p \quad (12)$$

where  $\langle \cdot \rangle_{\bar{p}}$  denotes expectation over empirical distribution, i.e. image samples, while  $\langle \cdot \rangle_p$  denotes expectation over the model distribution given by the current parameters. The first term is easy to compute by averaging over samples, while the second one is computational intractable. One approach is to run the Gibbs sampling defined by equations [4] [5] [10] [11]. However, it takes too long to run this sampler to equilibrium for every iteration. An alternative way is to initialize the Gibbs sampler on each data point, run only a few or even one steps of sampling, and then use the unconverged points to estimate the model expectation. This is known as contrastive divergence learning [5]. It reduces variance at the expense of a bias for the parameter estimates. Moreover, the bias is usually very small in practice. When computing the derivative of  $-E(x, h)$ , it is useful to reduce the variance of estimates by replacing  $h$  with  $P(h|x)$ .

### 4 Experiments

We use the MNIST digit dataset to run unsupervised learning and test the performance of digit retrieval of these two model. The training dataset and testing dataset both consist of 500 randomly chosen samples for each digit, that is, totally 5000 digits for each set, from the entire training and testing dataset.

#### 4.1 Preprocessing

Considering the limited time, a bicubic subsampling is performed to transform  $28 \times 28$  images to  $14 \times 14$  images. Then, each pixel is quantized to 0 or 1 with a threshold 50, since the input of our models is logic numbers. Figure 3 shows some digit samples in the dataset.



Figure 3: samples

#### 4.2 Training

We use 49 hidden units in both models. In each training iteration, we use a minibatch of 100 digits. The order of samples are randomly permuted after every round. In each iteration, Gibbs sampling runs only one step before computing the model expectation. Each model is

trained for 300,000 iterations for convergence. We then finely tune the CCM by continuing to train for 10,000 iterations with 5 step Gibbs sampling and newly sampled images are used as the initial point for the next round. It runs much slower with large Gibbs sampling steps. Similar tuning methods for SCM don't improve the performance.

### 4.3 Retrieval

In the retrieval phase, we assume the label of training digits are known and each testing digit retrieves similar training digits according to their distances, which are the cosine of angles between their latent vectors. Then, the precision-recall curve (PRC) is plotted. In SCM, we map the image onto the latent space by  $f = Wx$  [2]. In the CCM, the latent vector is computed as  $f = (V\sigma(V^T * ((Jx)^2 + \beta)) \times |Jx|$ , where  $\times$  means the component-wise multiplication. This formula is explained as follows.  $\bar{h} = \sigma(V^T * ((Jx)^2 + \beta))$  is the mean of hidden variables. Multiplying  $V$  with  $\bar{h}$  maps the influence of hidden variables onto the space of filter output as a weight. Then the amplitude of filter output  $Jx$  is adjusted by its weight. Also, we compare the model with PCA. The number of features,  $d$ , in PCA, which is 29 in this experiment, is chosen so that the PRC is optimized on the testing dataset. A more suitable way to choose  $d$  is using cross-validation. This is not done because of it is too hard to judge which  $d$  is better automatically and that the performance of PCA with cross-validation won't be better than choosing according to testing set.

### 4.4 Result

Figure 4, 5 show filters  $W$  and  $J$  respectively in SCM and CCM. In Figure 4, we see that most filters have a black or white hole with different positions in the flat background. Each filter detects the value at a specific location. In Figure 5, it has more complicated structures. Some images show short lines accompanied by two lines in the opposite color on both sides which is similar to Gabor filters. It has the ability to detect an edge at a certain location.

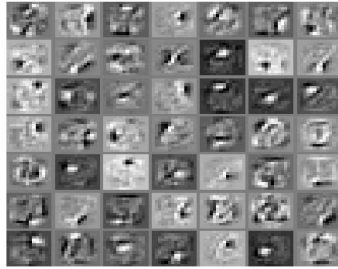


Figure 4: parameter  $W$  in Simple-Cell Model

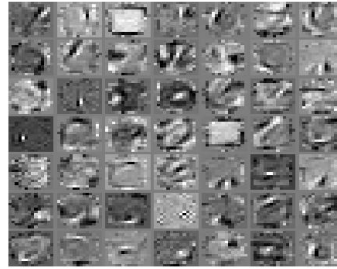


Figure 5: parameter  $J$  in Complex-Cell Model

The Precision-Recall curves (PRC) are shown in Figure 6. We find that the curve of CCM is slightly lower than that of SCM when recall is very small ( $< 0.003$ ). It means if we run a K-nearest neighbour classification especially when  $K = 1$ , the performance of complex-cell model is a little inferior to simplex-cell model. The curve of PCA is almost the same as SCM. Figure 7 is the result of k-NN classification. The 1-Nearest Neighbour precision of SCM is better than CCM, but the best k-NN classification is still obtained by CCM when  $k = 4$  though the difference is really negligible. In most range of recall, the precision of retrieval in CCM is the largest among these three models. This shows that CCM can keep some common features in the same class better than SCM even when two images of the

same digit are quite different.

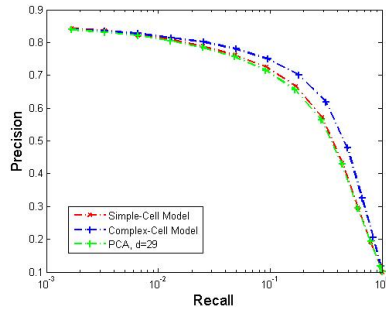


Figure 6: Precision-Recall Curve

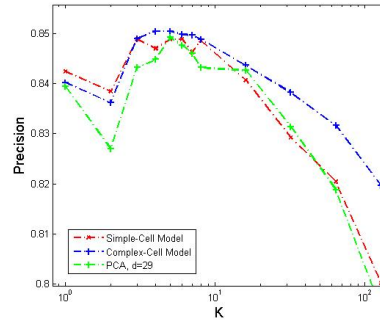


Figure 7: k-Nearest Neighbour classification

The ability of CCM to model correlation between pixels is illustrated by comparing Figure 8 and Figure 9. We runs Gibbs sampling in CCM and SCM for 1000 steps starting from the same 100 images. The samples are shown in these two figures. Most images are still easy to be recognized as a digit though the numbers they present are different from the original images. With a careful observation, we can see the images obtained by CCM are more clear and have less noise than by SCM. Strokes in digits are well presented in CCM, while they're less continuous in SCM. This can be ascribed to the conditional dependency between pixels induced by nonlinear transition.



Figure 8: Sampling in CCM



Figure 9: Sampling in SCM

## 5 Discussion

In this paper, we show that when adding another layer to the PoE, nonlinearity is introduced so that it models not only the value at each position but also the correlation between pixels. This changes the patch of filters and put constants on pixels of a stroke. Although the retrieval performance of CSM is better than a two layers harmonium, SCM for most recall values, the improvement is not very marked and the latent vector used here is not so intuitive. The following effort can be placed to improve the retrieval performance especially when recall is small. Also, other applications such as retrieval for documents and denoising can be implemented to see if this model has better properties than SCM in a more general scope.

## References

- [1] Max Welling. (2007) Products of Experts. *NIPS*
- [2] Max Welling, Michal Rosen-Zvi and Geoffrey Hinton. (2004) Exponential Family Harmoniums with an Application to Information Retrieval. *NIPS*
- [3] Peter Gehler, Alex Holub and Max Welling. (2006) The Rate Adapting Poisson (RAP) model for Information Retrieval and Object Recognition. *ICML*
- [4] Simon Osindero, Max Welling and Geoffrey Hinton. (2005) Topographic Product Models Applied to Natural Scene Statistics. *Neural Computation*, 18, pp 381-344.
- [5] Hinton, G. E. (2002) Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14, pp 1771-1800.