# THE CO-INFORMATION LATTICE

*Anthony J. Bell*

Redwood Neuroscience Institute
1010 El Camino Real, Suite 380
Menlo Park, CA 94025
(tbell@rni.org, tony@salk.edu)

## ABSTRACT

In 1955, McGill published a multivariate generalisation of Shannon's mutual information. Algorithms such as Independent Component Analysis use a different generalisation, the *redundancy*, or *multi-information* [13]. McGill's concept expresses the information shared by *all* of $K$ random variables, while the multi-information expresses the information shared by any two or more of them. Partly to avoid confusion with the multi-information, I call his concept here the *co-information*. Co-informations, oddly, can be negative. They form a partially ordered set, or lattice, as do the entropies. Entropies and co-informations are simply and symmetrically related by Möbius inversion [12]. The co-information lattice sheds light on the problem of approximating a joint density with a set of marginal densities, though as usual we run into the partition function.

Since the marginals correspond to higher-order edges in Bayesian hypergraphs, this approach motivates new algorithms such as Dependent Component Analysis, which we describe, and (loopy) Generalised Belief Propagation on hypergraphs, which we do not. Simulations of subspace-ICA (a tractable DCA) on natural images are presented on the web.

In neural computation theory, we identify the co-information of a group of neurons (possibly in space/time staggered patterns) with the 'degree of existence' of a corresponding cell assembly.

## 1. INTRODUCTION.

Many problems in machine learning and, we will argue, theoretical neuroscience may be reduced to the problem of estimating the probability density function, $p(\mathbf{x})$, of a random vector $\mathbf{x} = \{x_1 \dots x_N\}$. Examples in machine learning are the problem of inference in Bayesian networks (now called Generalised Belief Propagation, or GBP [16, 9]) and the problems of Independent [11, 3, 7] and Dependent [5, 6, 2] Component Analysis (ICA and DCA). Examples in theoretical neuroscience are sparse coding theories of visual perception [10, 4, 6] and theories of invariant coding and 'object' representation through cell assemblies [15].

## 2. THEORY.

In the hope of making progress along some of these fronts, we investigate, in this paper, the basic lattice structure of statistical dependency. The presentation will be very dense. In the Discussion, we will try to link some of the concepts to brain theory.

By this lattice structure, we mean the following. The indices of the $N$ variables in the random vector, $\mathbf{x}$, have a partially ordered set of subsets, called the *power set*. In a slight abuse of notation, we will denote this by $\mathcal{P}(\mathbf{x})$, where we mean the power set of the index-set of $\mathbf{x}$, rather than of $\mathbf{x}$ itself. For example, when $N = 3$:

$$\mathcal{P}(\mathbf{x}) = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$$

The lattice structure of this power set (generally a hypercube with $2^N$ vertices) is shown in Figure 1a. The sets of integers in the range $[1, N]$ are sets of indices into the vector $\mathbf{x}$, so that we can write $\mathbf{x}_{E_i}$ to sub-index into $\mathbf{x}$ with an index-set $E_i \in \mathcal{P}(\mathbf{x})$.

With each vertex, $E_i$, on the lattice of $\mathcal{P}(\mathbf{x})$, we associate

1. A marginal probability density, $p(\mathbf{x}_{E_i})$,
2. an entropy $H(\mathbf{x}_{E_i})$,
3. a co-information $I(\mathbf{x}_{E_i})$, and
4. an edge, $E_i$, (hence the '$E$'), potentially occuring in a hypergraph,
5. a sublattice of $E_i$ which we write as the power set $\mathcal{P}(E_i)$.

We will explain these terms in turn.

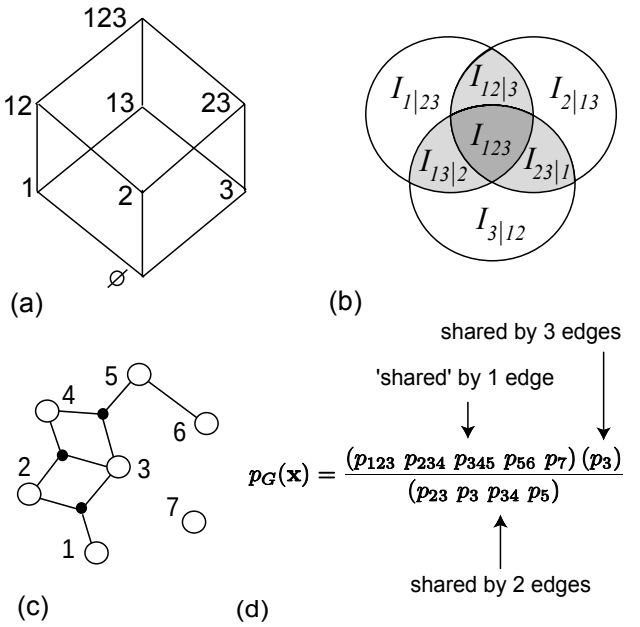


Figure 1: (a) The *lattice* of subsets of $\{1, 2, 3\}$, representing sets of indices, $E_i$, into a random vector, $\mathbf{x}$. (b) Venn diagram representation of the *co-informations* and conditional co-informations of three random variables. The three-way *redundancy* or *multi-information* [**?**] is the shaded part, and the co-information is the dark shaded part. (c) An example of a dependency *hypergraph*, $G$, over seven vertices (random variables). Black dots are used to denote 'edges' connecting more than two vertices together. (d) For an explanation of this expression for the joint density based on the hypergraph, $p_G(\mathbf{x})$, see Section 2.5.

### 2.1. Hypergraphs.

In Figure 1c, we illustrate how a hypergraph (or generalised dependency structure) is composed of 'edges' of order $K$, where $1 \leq K \leq N$. The figure shows edges of order one, two and three. In comparison, a conventional Bayesian network or a Markov Random Field has vertices ($K = 1$) and edges between pairs of those vertices ($K = 2$). A hypergraph, $G$, like an edge, $E_i$, is a member of a power set. But while an edge is a member of a power set over indices, a hypergraph is a member of a power set over edges:

$$\begin{aligned} G &\in \{\, E_i \mid E_i \in \mathcal{P}(\mathbf{x}) \,\} \text{ or equivalently} \\ G &\in \{\, E_i \mid \mathbf{x}_{E_i} \subseteq \mathbf{x} \,\} \text{ or more compactly} \\ G &\in \mathcal{P}(\mathcal{P}(\mathbf{x})) \end{aligned} \tag{1}$$

### 2.2. Entropy.

The differential entropy associated with an edge is the *total* amount of information carried by all its variables together. It is computed from its marginal probability by:

$$H(\mathbf{x}_{E_i}) = -\int_{\mathbf{x}_{E_i}} p(\mathbf{x}_{E_i}) \log p(\mathbf{x}_{E_i}) d\mathbf{x}_{E_i} \tag{2}$$

This joint entropy is the *join*, $\vee$, of the entropies directly below it on the lattice. Intuitively, it is something like set-theoretic *union* in that it collects all the information contained in the entropies one step below in the lattice, without overcounting the information redundant between them.

### 2.3. Co-information.

The co-information of an edge is the amount of information *shared* by all its variables together. It is defined as follows:

$$I(\mathbf{x}_{E_i}) = \sum_{E_j \subseteq E_i} q_j H(\mathbf{x}_{E_j}) \tag{3}$$

In this, we have introduced the *oddness*, $q_j$, of an edge (more generally, the Möbius inversion function [12]):

$$q_j = -(-1)^{|E_j|} = \begin{cases} 1 & \text{if } |E_j| \text{ is odd} \\ -1 & \text{if } |E_j| \text{ is even} \end{cases} \tag{4}$$

where $|E_j|$ is the cardinality (number of members) of the subset $E_j$. Also, since the empty set is included ($\varnothing \in E_i$), note that the empty vector contains no information: $I(\mathbf{x}_\varnothing) = H(\mathbf{x}_\varnothing) = 0$.

In words, Eq.(3) says that the co-information of the random vector associated with edge $E_i$ is calculated from the joint entropies in its sublattice. This is done by adding the entropies of odd-dimensional subedges and subtracting the entropies of even-dimensional ones.

A symmetrical formula exists defining the joint entropy in terms of the co-informations:

$$H(\mathbf{x}_{E_i}) = \sum_{E_j \subseteq E_i} q_j I(\mathbf{x}_{E_j}) \tag{5}$$

Low-dimensional examples of co-informations are the information shared by one variable (univariate entropy), two variables (mutual information) and three variables (3-way co-information). These special cases of Eq.(3) are:

$$\begin{aligned} I_1 &= H_1 \\ I_{12} &= H_1 + H_2 - H_{12} \\ I_{123} &= H_1 + H_2 + H_3 - H_{12} - H_{13} - H_{23} + H_{123} \end{aligned} \tag{6}$$

Here we have used a compact index-based notation for $I(\mathbf{x}_{E_i})$ and $H(\mathbf{x}_{E_i})$.

The co-information is the *meet*, $\wedge$, of the co-informations directly below it on the lattice. Intuitively, it is something like a set-theoretic *intersection*, in that it discards the information not shared amongst all of the subsets one step below in the lattice.

More explicitly, the co-information of $\mathbf{x} = \{x_1, x_2, x_3\}$ reads:

$$I(\mathbf{x}) = \int_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(x_1,x_2)p(x_1,x_3)p(x_2,x_3)}{p(x_1,x_2,x_3)p(x_1)p(x_2)p(x_3)} d\mathbf{x} \tag{7}$$

and this pattern of even-sized subsets on the numerator, and odd-sized subsets on the denominator continues for $N > 3$ so that we can rewrite Eq.(3) as:

$$I(\mathbf{x}_{E_i}) = -\int_{\mathbf{x}_{E_i}} p(\mathbf{x}_{E_i}) \log \prod_{E_j \subseteq E_i} p(\mathbf{x}_{E_j})^{q_i} d\mathbf{x}_{E_i} \tag{8}$$

where $p(\mathbf{x}_\varnothing) = 1$. Just as Eq.(3) is, in a sense, a dual equation to Eq.(5), so is Eq.(8) a dual to Eq.(2), which we can see by rewriting Eq.(5), plugging in all the marginals from Eq.(8), and watching them cancel to give Eq.(2).

### 2.4. Negative co-information.

Although the co-information is a correct measure of the information shared together by $K$ variables, equalling zero when there is no such dependency, it does have the odd feature that for odd $K > 1$, it can be negative.

We will calculate the simplest case of this, which involves $XOR$, or 2-parity. In this case, we have three binary variables, $\mathbf{x} = \{x_1, x_2, x_3\}$, where $x_3 = (x_1 + x_2) \mod 2$, and $x_1$ and $x_2$ assume each of their four possible combinations equiprobably. Using Eqs.(7), we can calculate that $I_1 = I_2 = I_3 = 1$ bit, while $I_{12} = I_{13} = I_{23} = 0$ bits. Now $H_{123} = 2$ bits because $x_3$ is a deterministic function of $x_1$ and $x_2$. Therefore, from Eqs.(7), we can calculate that $I_{123} = -1$.

The case of $N$-parity is also interesting. In this case, $\mathbf{x} = \{x_1 \dots x_N\}$, $x_N = \left(\sum_{i=1}^{N-1} x_i\right) \mod 2$, and all combinations of $\{x_1 \dots x_{N-1}\}$ are equiprobable. Then $I_i = 1$ for all $i$, and $I(\mathbf{x}_{E_i}) = 0$ for all $E_i \in \mathcal{P}(\mathbf{x})$ such that $1 < |E_i| < N$. In other words, only the univariate entropies and the 'top' co-information $I(\mathbf{x})$ are non-zero. In order that $H(\mathbf{x}) = N - 1$, as before, it is necessary, by Eq.(5), that when $N > 1$ and is odd, $I(\mathbf{x}) = -1$ while when $N$ is even, $I(\mathbf{x}) = 1$.

This case is interesting to compare with the case of an $N$-dimensional binary vector $\mathbf{x}$ which is equiprobably all zeros or all ones. Then the shared informations of all orders are each one bit, and therefore $I(\mathbf{x}_{E_i}) = 1$ for all $E_i \in \mathcal{P}(\mathbf{x})$. Plugging these values into Eq.(5), we can then see that $H(\mathbf{x}) = 1$, as required.

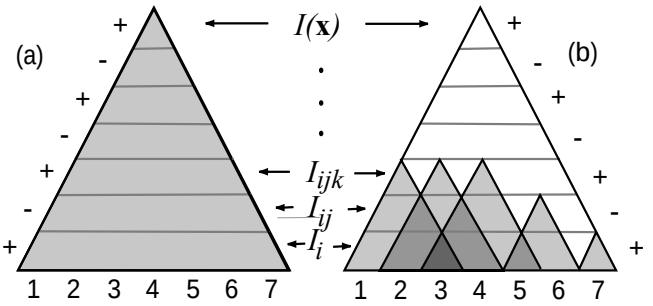


Figure 2: (a) A schematic view of the co-information lattice for $\mathbf{x} = \{x_1 \dots x_7\}$. The combinations of co-informations of various orders gives the entropy $H(\mathbf{x})$. (b) A schematic view of the partial co-information lattice for the graph entropy $H_G(\mathbf{x})$, where $G$ is the graph in Figure 1c. The equation $D[p\|p_G] = H_G(\mathbf{x}) - H(\mathbf{x})$ is represented pictorially as:

Thus, we have shown that for odd $K > 1$, the co-information can be positive or negative. It is not quite clear yet what it means to have a negative co-information, but it is clear that a non-zero value signals the existence of a $K$th-order dependency.

Here we have considered datasets determined by one logical function. It is interesting to generalise this to datasets determined (or over- or under-determined) by more than one logical function. Then we may discover uses for the co-information lattice in solving Boolean satisfiability problems.

### 2.5. Density Estimation

Density estimation is performed by minimising the Kullback-Leibler, or $KL$, divergence between the true density, $p(\mathbf{x})$, of $\mathbf{x}$, and a density calculated according to a model. By model, we mean here a hypergraph $G \in \mathcal{P}(\mathcal{P}(\mathbf{x}))$. The density according to the model, we denote by $p_G(\mathbf{x})$. How do we calculate $p_G(\mathbf{x})$ from a graph, $G$, and a (presumed accurate) set of marginals, $p(\mathbf{x}_{E_i})$ over its edges?

To see this, consider Figure 2. The left side of this figure is a 'cartoon' depiction of Eqs.(3,5), showing how the 7 dimensional entropy/co-information is calculated by adding and subtracting all the co-informations/ entropies of even and odd orders in the power set, $\mathcal{P}(x)$. (It is a cartoon because it represents this lattice as a continuous $2D$ triangle, rather than as a discrete $7D$ boolean hypercube.)

We compare the case in Figure 2a where we have all

the co-informations needed to 'define' $H(\mathbf{x})$, with the case in Figure 2b, where we only have the shaded area. This area corresponds to an (actually larger) quantity called the *graph (cross) entropy*:

$$H_G(\mathbf{x}) = -\int_{\mathbf{x}} p(\mathbf{x}) \log p_G(\mathbf{x}) d\mathbf{x} \quad (9)$$

$$= Z_G(\mathbf{x}) + \sum_{E_i \in \cap G} q_i H(\mathbf{x}_{E_i}) \quad (10)$$

where

$$p_G(\mathbf{x}) = \frac{1}{z_G(\mathbf{x})} \prod_{E_i \in \cap G} p(\mathbf{x}_{E_i})^{q_i} \quad (11)$$

and $Z_G(\mathbf{x}) = \int_{\mathbf{x}} p(\mathbf{x}) \log z_G(\mathbf{x}) d\mathbf{x}$ is an entropy-style functional of $p(\mathbf{x})$, derived from the *partition function*, $z_G(\mathbf{x}) = \int_{\mathbf{x}} \prod_{E_i \in \cap G} p(\mathbf{x}_{E_i})^{q_i} d\mathbf{x}$, required to normalise Eq.(11) to be a probability distribution.[1]

Where did these equations come from and what is $\cap G$? We give an intuitive account here. The shaded area in Figure 2b corresponds to the hypergraph, $G$, in Figure 1b, each peak corresponding to one of the five 'edges' in $G$, each of the five overlapping triangles corresponding to one of the edges' sublattices, each sublattice representing a $H(\mathbf{x}_{E_i})$ by combinations of co-informations as in Eq.(5). If the shaded area represents $H_G(\mathbf{x})$, what does the white space above it represent? It represents the negative of the (necessarily positive) KL divergence between the true density and the hypergraph model density, as shown in the following equation:

$$D[p \| p_G] = H_G(\mathbf{x}) - H(\mathbf{x}) = \int_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{p_G(\mathbf{x})} d\mathbf{x} \quad (12)$$

Plugging Eq.(10) and Eq.(5) into Eq.(12), yields:

$$D[p \| p_G] = Z_G(\mathbf{x}) - \sum_{\substack{E_j \not\subseteq E_i, \\ \forall E_i \in G}} q_j I(\mathbf{x}_{E_j}) \quad (13)$$

The second term here is a sum of co-informations which are higher in the lattice than those in the graph.

We calculate $p_G(\mathbf{x})$ from its marginals, the $p(\mathbf{x}_{E_i})$, by first calculating $H_G(\mathbf{x})$, the shaded area in Figure 2b. This consists of overlapping marginal entropies, $H(\mathbf{x}_{E_i})$. We could sum up these overlapping triangles but we would overcount the overlaps between the edges (the darker shades). We could then subtract each overlap once but we would then undercount the overlap between the overlaps (the darkest shade). Although this example terminates with overlaps of overlaps, the general case leads to Eq.(10), a Möbius inversion [12] very similar in style to Eqs.(3,5).

[1] Thanks to NIPS*2002 for pointing this term out to me.

But Eq.(10) still needs some explaining. By the obscure notation, $\cap G$, we mean to define the *intersection list* of a graph, $G$:

$$\cap G = \mathrm{Map}(\cap, \mathcal{P}(G)) \quad (14)$$

where 'Map' applies the intersection operator, $\cap$, to all sets of subsets of the graph $G$'s edges. In pictures, $\cap G$ is a list of all ten triangles that can be discerned in the shaded area of Figure 2b (bearing in mind that the darkest triangle appears twice, as both a two- and as a three-way intersection). In Figure 1d, $\cap G$ is a list, with duplication, of all the index-sets appearing in the expression for $p_G(\mathbf{x})$.

By adding and subtracting correctly the triangles corresponding to these index-sets, we can assemble the whole shaded area, $H_G(\mathbf{x})$, except for the average log partition function. This combination process is exactly what is expressed in Eq.(10). Combining Equations (3,5,9,10) we get our graph-based density estimate, expressed in Eq.(11).

Often, as in Figure 1c, there will cancellations in the marginal terms. It will be interesting to see which classes of hypergraphs lead to various levels of complexity of $Z_G(\mathbf{x})$, and how GBP may be incorporated.

### 2.6. Bayesian Estimation.

In Bayesian analysis, and presumably also in the brain to some degree, we are concerned with the probabilities of different interpretations of the data, rather than just the data probabilities. Thus, instead of considering the true and modelled data densities, $p(\mathbf{x})$ and $p(\mathbf{x}|G)$ (as we now rewrite $p_G(\mathbf{x})$), we now consider the prior and posterior densities over the models, $p(G)$ and $p(G|\mathbf{x})$.

These distributions are related, via Bayes' theorem: $p(G|\mathbf{x}) = p(\mathbf{x}|G)p(G)/p(\mathbf{x})$. The graphs themselves form a lattice, and we move a single step on this lattice by altering a edge, $E_j$, through the addition or removal of a single variable, $x_i$. The difference in $\log p(G|\mathbf{x})$ for an addition, we write as follows

$$\underset{x_i \to E_j}{\Delta} \log p(G|\mathbf{x}) = \underset{x_i \to E_j}{\Delta} \log p(\mathbf{x}|G) + \underset{x_i \to E_j}{\Delta} \log p(G) \quad (15)$$

where the $p(\mathbf{x})$ term in Bayes' theorem has cancelled out. The prior over the graphs, $p(G)$, thus acts as a constraint, enabling us to avoid 'maximum likelihood overfitting', the choosing of the 'one big edge' graph, $G = \{\{1 \dots N\}\}$, for which $H(\mathbf{x}|G) = H(\mathbf{x})$.

The question of what is the most natural prior over the hypergraphs is one that probably has a quite elegant answer.

We could move on the lattice of graphs to find the maximum *a posteriori* (MAP) graph, the most probable interpretation, but it would be preferable to find

some method for implicitly coding the probabilities of the different graphs. For a suggestion on how this may be done in the brain, see the Discussion.

## 3. ALGORITHMS.

A number of algorithms may be explored using our expression for $p_G(\mathbf{x})$ in Eq.(11). An intriguing connection is with energy-based Belief Propagation algorithms for performing inference over hidden variables. In fact, the *region-graph approximations* explored in this context in [16] are related to our method. They define regions of a graph and calculate the over- and under-counted intersections between the regions just as we do in Figure 2b. However, while their regions are selected heuristically, ours are determined by the marginals involved. Stronger connections can be made with [9], where beliefs propagate explicitly on the lattice. Space does not permit a full discussion of these connections. We proceed instead to a discussion of DCA.

### 3.1. Dependent Component Analysis.

In Independent Component Analysis (ICA) and its emerging generalisation, Dependent Component Analysis (DCA), we are looking for a linear transform, $\mathbf{u} = \mathbf{W}\mathbf{x}$, of the random vector, $\mathbf{x}$, such that, again, the KL-divergence, $D[p(\mathbf{x})||p_G(\mathbf{x})] = H_G(\mathbf{x}) - H(\mathbf{x})$, is minimised.

The varying term, the graph cross entropy, in DCA, is written (from Eq.(10) and $H_G(\mathbf{u}) = H_G(\mathbf{x}) + \log|\mathbf{W}|$):

$$H_G(\mathbf{x}) = Z_G(\mathbf{x}) - \log|\mathbf{W}| + \sum_{E_j \in \cap G} q_j \hat{H}(\mathbf{u}_{E_j}) \quad (16)$$

where $|\cdot|$ denotes absolute determinant, and estimated (or parameterised) marginals, $\hat{p}(\mathbf{u}_{E_i})$, have been introduced, yielding *edge (cross) entropies*:

$$\hat{H}(\mathbf{u}_{E_i}) = -\int_{\mathbf{u}_{E_i}} p(\mathbf{u}_{E_i}) \log \hat{p}(\mathbf{u}_{E_i})\, d\mathbf{u}_{E_i} \quad (17)$$

We perform natural gradient descent [1] in Eq.(16), or, equivalently, maximise the likelihood, by adjusting the weights with $\Delta\mathbf{W} \propto -(\nabla_{\mathbf{W}} H_G(\mathbf{x}))\, \mathbf{W}^T\mathbf{W}$ giving a stochastic gradient learning rule:

$$\Delta\mathbf{W} \propto \left(\mathbf{I} + \mathbf{f}(\mathbf{u})\mathbf{u}^T\right)\mathbf{W} - (\nabla_{\mathbf{W}} Z_G(\mathbf{x}))\, \mathbf{W}^T\mathbf{W} \quad (18)$$

The familiar first term is as in ICA, except that now the non-linearities (the *score functions*), crucial to the working of the ICA, are generalised for multi-dimensional marginals. The $i$th element of the vector function $\mathbf{f}(\mathbf{u})$ is now:

$$f_i(\mathbf{u}) = \sum_{\substack{E_j \in \cap G, \\ u_i \in E_j}} q_j \frac{\partial}{\partial u_i} \log \hat{p}(\mathbf{u}_{E_j}) \quad (19)$$

containing terms from every $E_j$ containing $u_i$. For ICA, this just reduces to $f_i(\mathbf{u}) = \partial/\partial u_i \log \hat{p}(u_i)$.

The second term in Eq.(18) is the natural gradient of the troublesome average log partition function. For ICA, and also for subspace-ICA [6], it is zero. Topographic ICA [6], which has edges corresponding to overlapping neighbourhoods in a 'visual cortical map', is an unnormalised model in that it ignores the partition function.

To make things more concrete, we will consider the sparse exponential model in which each edge, $E_j \in G$, has associated with it an estimated marginal which is a radially symmetric laplacian, $\hat{p}(\mathbf{u}_{E_j}) = a\exp(-b\|\mathbf{u}_{E_j}\|)$, where $\|\mathbf{u}_{E_j}\|$ denotes the euclidean length of the projection of $\mathbf{u}$ on the subspace $E_j$, and $a$ and $b$ are constants to be determined.

It would be nice if $\hat{p}(\mathbf{u}_{E_j})$ was normalised and if the average length of $\mathbf{u}_{E_j}$ was 1. Define $r = \|\mathbf{u}_{E_j}\|$ and $n = |E_j|$. Then we find (by dividing one of the corresponding hyperspherical integrals by the other) that $b = n$, and our two constraints are met by:

$$\hat{p}(\mathbf{u}_{E_j}) = \frac{n^n}{(n-1)!\, S_n^1} e^{-nr} \quad (20)$$

where $S_n^r = 2\pi^{n/2} r^{n-1}/\Gamma(n/2)$ is the surface area of an $n$-dimensional hypersphere of radius $r$. The distribution of $r$ under the model is Eq.(20) times $S_n^r$:

$$\hat{p}(r) = \frac{n^n r^{n-1}}{(n-1)!} e^{-nr} \quad (21)$$

Finally, we can calculate the necessary terms in Eq.(19) for the radial laplacian learning model. Simply:

$$\frac{\partial}{\partial u_i} \log \hat{p}(\mathbf{u}_{E_j}) = -\frac{n}{r} u_i \quad (22)$$

Radially symmetric distributions have already shown their use in the most successful DCA algorithm to date, Hyvärinen & Hoyer's subspace-ICA, which, building on [4, 10], accounts for most of the main features of brain area V1 organisation. These include simple and complex orientation selective cells, and topography, including the correct behaviour of orientation and spatial frequency selectivity around singularities in the map.

To accompany this paper, we present on the web[2] results and commentary from our implementation, Eq. (22), of subspace-ICA trained on 16x16 natural images. We used subspaces of size $2^n$ for $n$ from 0 to 8.

For DCA with graphs giving rise to an $\mathbf{x}$-dependent partition function, we may have to resort to methods used in other 'free energy-based models' in order to estimate the gradient of the log partition function. For example, [14] use *contrastive divergence*, a sampling technique, to learn overcomplete ICA mappings.

[2] `http://www.cnl.salk.edu/~tony/RNI.html`

## 4. DISCUSSION.

The concept of co-information is a natural one to use in attempting to build a theory of machine learning, just as the concept of 'cell assembly' (a structure containing associative memories) is natural for brain theory. Both represent degrees of "hanging togetherness" amongst elements. We argue that the two concepts are the same.

They arise from intuitions (both from mathematics and from neuroscience) that patterns (regularities, symmetries, dependencies, redundancies) in the world, can be captured in structured groupings of variables or neural events. In this context, the memories embedded in a group of neurons are the patterns that are statistically more likely to appear in this group. The overall propensity of a group to produce patterns together is what we mean by the co-information: it is a measure of 'groupness'.

More complex than a memory is what we refer to as an 'interpretation'. This is a grouping of memories that also 'hang together'. An overall propensity of a group of memories to activate together is what we mean by a hypergraph, which is just a list of groupings. We could continue taking power sets, talking about groupings of interpretations (graphs of graphs), and so on. But for reasons of combinatorial complexity, it is better to search for ways to implicitly represent the hierarchy of such structures. For example, rather than comparing all the (hyper-exponential number of) interpretations in an area of cortex, through Eq.(15), we can evaluate a set of high co-information 'edge marginals', subsets of which, via Eq.(11), combine to give different interpretations, $G$.

The roles of synapses and synchrony are most interesting in this context. Perhaps high co-information groups may be most easily spotted as groups of frequently synchronised neurons [15], and that, correspondingly, the easiest way to spot them in multi-unit neuronal recordings, is through calculating co-informations, across space, and if necessary, time (synfire chains).

In summary, however, the main results: I described the co-information lattice, used it to show how to express the probability density under a general hypergraphical model, and then used this to derive the lattice of Dependent Component Analysis algorithms.

**Acknowledgements.** Thanks for help from Bruno Olshausen, Barak Pearlmutter, Martin Wainwright, Ilya Temenman, Jonathan Yedidia, Juan Lin, Bill Softky and Glen Brown. Thanks for inspiration to Paul Bush, Zach Mainen, Rita Venturini and Terry Sejnowski, and also to my RNI colleagues Pentti Kanerva, Fritz Sommer, and Jeff Hawkins.

## 5. REFERENCES

[1] Amari S.-I., Cichocki A. & Yang H. 1996. A new learning algorithm for blind signal separation. *NIPS 8*.

[2] Bach F.R. & Jordan M.I. 2002. Finding Clusters in Independent Component Analysis, *UCB/CSD-02-1209*, University of California, Berkeley.

[3] Bell A.J. & Sejnowski T.J. 1995. An information-maximization approach to blind separation and blind deconvolution. *Neural Comp.*, 7,6:1129-1159.

[4] Bell A.J. & Sejnowski T.J. 1997. The 'independent' components of natural images are edge-filters. *Vision Research*, 37:3327-3338, 1997

[5] Cardoso J.-F. 1998. Multidimensional independent component analysis. In *Proc. ICASSP, 1998*

[6] Hyvärinen A. & Hoyer P. 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.

[7] Hyvärinen A., Karhunen J. & Oja E. 2001. *Independent Component Analysis*, John Wiley.

[8] McGill W.J. 1955. Multivariate information transmission, *IEEE Trans. Inf. Theory*, 4, 4, 93-111

[9] McEliece R.J. & Yildirim M. 2003. Belief propagation on partially-ordered sets. Available from www.ee.caltech.edu/EE/faculty/rjm

[10] Olshausen B. & Field D.F. 1997. Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images. *Nature*, 381: 607-609.

[11] Pham D.T., Garat P. & Jutten C. 1992. Separation of mixture of independent sources through a maximum likelihood approach. *Proc. EUSIPCO '92*

[12] Stanley R.P. 1999. *Enumerative Combinatorics, vol. I & II*, Camb. Univ. Press

[13] Studenty M. & Vejnarova J. 1998. The multiinformation function as a tool for measuring stochastic dependence. In Jordan M.I. (ed.) 1998. *Learning in Graphical Models*, Dordrecht: Kluwer, 1998.

[14] Teh Y-W., Welling M., Osindero S. & Hinton G.E. 2003. Energy-based models for sparse overcomplete representations. *J. Machine Learn. Res.*, submitted.

[15] von der Malsburg, C. 1999. The What and Why of Binding: The Modeler's Perspective, *Neuron* 24, 1, 95-104.

[16] Yedidia J.S., Freeman W.T. & Weiss Y. 2002. Constructing free energy approximations and Generalised Belief Propagation algorithms. *TR-2002-35* Mitubishi Electronic Reseach Labs (www.merl.com)

2.4.03