# Towards a Theory of Compositional Learning and Encoding of Objects

A.L. Yuille
Dept. of Statistics, University of California at Los Angeles
Dept. of Brain and Cognitive Engineering, Korea University, Seoul, Korea
yuille@stat.ucla.edu

## Abstract

*This paper develops a theory for learning compositional models of objects. It gives a theoretical basis for explaining the effectiveness of recent learning algorithms which exploit compositionality in order to perform structure induction of graphical models. It describes how compositional learning can be considered as learning either probability models or efficient codes for objects.*

## 1. Introduction

The advantages of compositional models of objects have been eloquently argued by S. Geman in papers like [3]. Such ideas have helped to motivate some successful learning algorithms [5], [2],[13],[12],[9],[11]. In particular, works by L. Zhu and his collaborators have shown that compositionality can be used to perform challenging structure induction tasks, such as learning hierarchical object models with limited supervision [13] and learning models of multiple objects with part-sharing [11].

Despite these successes the underlying theory has been less clear. What are the properties of compositional models that enable them to be learnt by the procedures in [13],[11]? How do these learning approaches differ from existing methods such as feature pursuit [6],[14] or the Expectation Maximization (EM) algorithm, see citations in [12]? From the perspective of this workshop, how do these approaches relate to learning efficient codes for objects in the sense of Minimal Description Length (MDL) [7] or rate distortion theory [4]?

This paper addresses these issues. It develops a theory for structure induction exploiting properties of compositional models. It describes how compositional learning can be considered as learning either probability models or efficient codes for objects. It sketches how theoretical results may be obtained to determine situations where compositional learning may provably converge to the optimal solution, following the analysis performed in [10].

The structure of the paper is as follows. In section (2) we describe the motivations for compositional models. Section (3) discusses the feature pursuit methods which are capable of learning graphical models if the graph structure is known and the data is aligned. In section (4) we discuss the limitations of the feature pursuit approach and how it is hard to adapt to the more challenging problems of learning object models. Section (5) introduces a simple compositional model and its basic properties which enable compositional learning to be possible. In section (6) we describe how compositional learning performs a breadth first search through the space of models. Section (7) discusses the clustering algorithm used in model search and how it relates to the encoding the data. In section (8) we discuss the limitations of compositional learning. Section (9) distinguishes compositional learning from other approaches.

## 2. Motivation: Why Learn Compositional Models for Vision?

How do we learn probability models of image or objects if the models are unknown and the correspondence between models and images are unknown (i.e. we do not know what object, or objects, appear in the images and where different parts of the objects appear)?

This is a very challenging problem. In its most ambitious form it is the task faced by an infant exploring the visual world – how to model all the image patterns that occur in images? How to discover automatically the objects that generate them?

More concrete examples include the learning of object models where the image features are interest points (i.e. the rest of the image is ignored). Work by L. Zhu *et al* specified an algorithm which performed greedy search in model space to achieve this [12]. Another example is the learning of a hierarchical probabilistic model of an object from a set of training images which include examples of the object in the presence of variable background noise [13]. Yet another example, is to learn the models for multiple objects simultaneously using the silhouettes of object examples as inputs [11].

The advantages of compositional models include the ability to share parts between different objects which has big advantages for representation, inference, and learning. In terms of representation, this means that we can encode many objects in terms of the same dictionary of parts. Inference is also more efficient because we only need to search for a part once even though it can occur in multiple objects, or multiple times in the same object. Learning is also more efficient because sharing parts between objects requires less training data (than if we had to learn each object separately).

## 3. Feature Pursuit

Feature pursuit [6],[14] is a procedure to learn probabilistic models when there are no hidden variables and the data is perfectly aligned with the model. It is assumed that the probability distribution can be expressed as an exponential model in terms of statistics $\phi(.)$ with parameterized weights $\lambda$. There is a pre-specified dictionary of statistics which are selected, and assigned weights, during learning. The learning can be expressed as a convex optimization problem because there are no hidden variables. In graphical terms, the set of graph nodes are fixed, because they are determined by the form of the data, but the edges in the graph are unknown because they will depend on the statistics which are selected. So this is not a challenging structure induction problem.

More precisely, we assume an exponential model:

$$P(x|\lambda) = \frac{1}{Z[\lambda]} \exp\{\sum_{i=1}^{M} \lambda_i \cdot \phi_i(x)\}, \qquad (1)$$

where there is a dictionary of statistics $\{\phi_i(x) : i = 1, ..., M\}$ and parameters $\lambda = \{\lambda_i : i = 1, ..., M\}$. There is a set of training data $D = \{x^\mu : \mu = 1, ..., N\}$. The task is to estimate the parameters $\{\lambda_i : i = 1, ..., M\}$ by Maximum Likelihood (ML) from the data while restricting the number of non-zero $\lambda_i$. ML can be expressed as:

$$\{\lambda_i^*\} = \arg\max \prod_{\mu=1}^{N} P(x^\mu|\lambda). \qquad (2)$$

For exponential models, ML reduces to matching the statistics of the model to the statistics of the data:

$$\sum_x \phi_i(x) P(x|\lambda^*) = \frac{1}{N} \sum_{\mu=1}^{N} \phi_i(x^\mu). \qquad (3)$$

Feature pursuit searches through the space of possible models – i.e. the choice of feature statistics – in a greedy manner as follows. We initialize with a default model which contains no statistics (i.e. $\lambda_i = 0, \forall i$). After $t$ steps, we have chosen $t$ statistics $\{\phi_i(.) : i = 1, .., t\}$ and their associated parameters $\lambda^t = \{\lambda_i : i = 1, ..., t\}$ to have a distribution $P_t(x|\lambda^t) = \frac{1}{Z[\lambda^t]} \exp\{\sum_{i=1}^{t} \lambda_i \phi_i(x)\}$. The probability of the data using this model is expressed as $P(D|\lambda^t) = \prod_\mu P_t(x^\mu|\lambda^t)$ which, as we describe below, is directly related to the entropy of $P_t(x|\lambda^t)$. At step $t + 1$, we augment this model by adding an additional feature $\phi_{t+1}(.)$ to the exponent and perform ML estimation to estimate its parameter $\lambda_{t+1}$. The feature is chosen to maximize the probability of the data $P(D|\lambda^{t+1}) = \frac{1}{Z(\lambda^{t+1})} \exp\{\sum_{i=1}^{t+1} \lambda_i \phi_i(x)\}$. We accept the new features, provided $P(D|\lambda^{t+1}) > P(D|\lambda^t) \times \exp\{T\}$ where $T$ is a threshold, otherwise we stop and output $P_t(x|\lambda^t)$.

This is a greedy method for searching through model space. At any time $t$ we maintain a single model $P_t(x|\lambda^t)$. At time $t+1$ we consider $M$ possible expansion corresponding to augmenting $P_t(x|\lambda^t)$ by adding any of the $M$ possible statistics from the dictionary $\{\phi_i(.) : i = 1, ..., M\}$. Model selection is used to select which new feature to add – i.e. we compare $P(D|...)$ for the $M$ possible models and select the model $P_{t+1}(x|\lambda^{t+1})$ for which it is largest.

A classic example of selecting features based on this type of criterion comes from the work of Shannon on modeling the statistics of the English language [8]. In this case, the statistics are the frequencies of letters, of pairs of letters, of triples of letters, and so on.

Shannon clarified that the probability of the data $P(D|\lambda^*)$ (with $\lambda^*$ estimated by ML) is directly related to the entropy of the distribution $-\sum_x P(x|\lambda^*) \log P(x|\lambda^*)$. Hence model selection corresponds to picking the distribution which has lowest entropy and hence is best at predicting the data.

Shannon's results follow directly from the following identity which holds for all exponential distributions:

$$\begin{aligned} & \frac{1}{N} \sum_\mu \log P(x_\mu|\lambda^*) \\ &= \frac{1}{N} \lambda^* \cdot \sum_{\mu=1}^{N} \phi(x_\mu) - \log Z[\lambda^*] \\ &= \lambda^* \cdot \sum_x \phi(x) P(x|\lambda^*) - \log Z[\lambda^*], \qquad (4) \end{aligned}$$

which is the negative entropy of $P(x|\lambda^*)$. Here $\lambda \cdot \phi(x) = \sum_{i=1}^{M} \lambda_i \phi_i(x_i)$.

Hence the search through model space can be considered as searching for the best way to encode the data. Note that Shannon's theory does not specify the precise code – instead it proposes that data $x$ should be encoded by $-\log P(x|\lambda^*)$ bits. Of course, it will be natural to encode $x$ in terms of the statistics $\phi(.)$.

## 4. Beyond Feature Pursuit

The type of learning we address in this paper is more challenging than that addressed by feature pursuit. Firstly, the probability model has unknown graph structure – i.e. we do not know the nodes or the edges. Secondly, typically only a subset of the data will correspond to the object that we seek to learn while the remainder will be background clutter. Thirdly, we do not know the correspondence between the object, or objects, and the data.

At a minimum, we must extend the probability models to include the graph structure $S$ (i.e. the nodes and edges) and the assignment variables $A$ between parts of the object and the data. But we should also generalize the structure to AND/OR graphs so that the assignment can allow us to select different objects for different data.

Formally, we can express this in terms of searching through models of form $P(x|S, \lambda, A)$. The structure $S$ and the assignments $A$ can be thought of as missing data which need to be estimated, or summed/integrated out, while estimating the model parameters $\lambda$.

The Expectation-Maximization (EM) algorithm can, in theory, be used for such missing data problems as we will describe in detail in the next section. But EM is only suited to a limited class of problems because it relies on E- and M-steps which can be extremely difficult to compute. Even when this is practical – e.g., for probability models with known graph structure without closed loops – it is not guaranteed to converge to the global solution. Moreover, EM is very rarely applied to learning the graph structure, in addition to model parameters, and it seems impractical for the class of problems we are considering.

This motivates us to consider an alternative approach which restricts ourselves to compositional models. We will first describe their properties which explain why learning them is practical even for challenging situations.

## 5. A Simple Compositional Model

Now we introduce a simple compositional model, see figure (1). This is much simpler than the models learnt in [13],[11] but it illustrates the key ideas. The compositional model has only four leaf nodes which, in this section, are assumed to have known correspondence to the data $x^\mu = (x_1^\mu, x_2^\mu, x_3^\mu, x_4^\mu)$. The model contains hidden variables $x_{12}, x_{34}, x_{1234}$ which are related, deterministically, to the states of the leaf nodes $x_1, x_2, x_3, x_4$ [13],[11]. In the terminology which we will use later, the full model with all the nodes is a level-2 part-model while the subtrees with nodes $x_1, x_2, x_{12}$ and $x_3, x_4, x_{34}$ are level-1 part-models.

We express the compositional model in terms of the conditional distributions $P(x_1, x_2|x_{12}), P(x_3, x_4|x_{34}), P(x_{12}, x_{34}|x_{1234})$. The intuition is that $x_{12}$ represents the position of a part which is a composite of two subparts with positions $x_1$ and $x_2$.

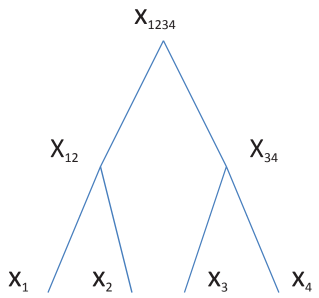

Figure 1. A simple compositional model. This is a level-2 part-model, in the terminology of section (6), and is the composition of two level-1 part-models with parent nodes $x_{12}$ and $x_{34}$ respectively.

$$P(x_1, x_2|x_{12}) = f(x_1 - x_2 \; \lambda_{12})\delta(x_{12} - \frac{x_1 + x_2}{2}),$$
$$P(x_3, x_4|x_{34}) = f(x_3 - x_4|\lambda_{34})\delta(x_{34} - \frac{x_3 + x_4}{2}),$$
$$P(x_{12}, x_{34}|x_{1234}) = f(x_{12} - x_{34}|\lambda_{1234})\delta(x_{1234} - \frac{x_{12} + x_{34}}{2}) \quad (5)$$

Here the $f(.,.)$ are probability distributions on the relative positions of parts/subparts. These distribution have unknown parameters $\lambda$. The distribution of the root node $x_{1234}$ is specified as a uniform distribution $U(x_{1234})$.

Equation (5) specifies a very simple compositional model. Its key properties, described in the next section, is that the parameters $\lambda_{12}, \lambda_{34}, \lambda_{1234}$ can be learnt independently. Other definitions of compositional models can be given (S. Geman – personal communication) which include this model as a special case.

### 5.1. Learning compositional models with known assignment

This special form of compositional models implies that the parameters of the probability distributions can be learnt separately if we know the correspondence to the data:

$$\lambda_{12}^* = \arg\max \prod_\mu f(x_1^\mu - x_2^\mu|\lambda_{12}),$$
$$\lambda_{34}^* = \arg\max \prod_\mu f(x_3^\mu - x_4^\mu|\lambda_{34})$$
$$\lambda_{1234}^* = \arg\max \prod_\mu f(\frac{x_1^\mu + x_2^\mu}{2} - \frac{x_3^\mu + x_4^\mu}{2}|\lambda_{1234}). \quad (6)$$

To verify this, observe that after summing out the hidden variables $x_{12}, x_{34}, x_{1224}$, we can express the compositional model in exponential form:

$$P(x_1, x_2, x_3, x_4|\lambda) = \frac{1}{Z[\lambda_{12}]Z[\lambda_{34}]Z[\lambda_{1234}]} \exp\{\lambda \cdot \phi(x)\}, \tag{7}$$

where $\lambda = (\lambda_{12}, \lambda_{34}, \lambda_{1234})$ and $\phi(x) = (\phi(x_1 - x_2), \phi(x_3 - x_4), \phi(\frac{x_1+x_2}{2} - \frac{x_3+x_4}{2}))$.

The result follows by doing ML to estimate $\lambda$ from the data $D = \{x^\mu\}$.

### 5.2. Learning without assignment or structure

The simplest extension is to learn the compositional model when the assignment is unknown. We introduce an assignment variable $A$ which specifies the assignments $a(i)$ of nodes $i$ of the model.

$$\begin{aligned} P(\{x_a\}|A, \lambda) = f(x_{a(1)} - x_{a(2)}|\lambda_{12}) f(x_{a(3)} - x_{a(4)}|\lambda_{34}) \\ f(\frac{x_{a(1)} + x_{a(2)}}{2} - \frac{x_{a(3)} + x_{a(4)}}{2}|\lambda_{1234}) \\ U(\frac{x_{a(1)} + x_{a(2)} + x_{a(3)} + x_{a(4)}}{4}). \end{aligned} \tag{8}$$

The standard procedure to learn this type of model is to treat the assignment $A$ as missing variables, to assign a prior distribution $P(A)$ over them (e.g., the uniform distribution $U(A)$), and to estimate $\lambda$ by ML:

$$\lambda^* = \arg\max \prod_\mu \sum_{A^\mu} P(\{x_a^\mu\}|A^\mu, \lambda) P(A^\mu). \tag{9}$$

This estimation can, in theory, be done by the Expectation-Maximization (EM) algorithm. This requires iterating two steps – the E-step which estimates distributions $Q_\mu^t(A^\mu)$ over the hidden variables where the parameters $\lambda^t$ are fixed, and the M-step which estimates $\lambda^{t+1}$ assuming $Q_\mu^t(A^\mu)$. But there are two limitations to this algorithm. Firstly, the E- and M-steps can be computed by dynamic programming *provided* we impose the correspondence condition that each model point (i.e. leaf node) has a single correspondence in the image, *but not* if we impose one-to-one matching between the model points and the image. Secondly, and more seriously, the EM algorithm is not guaranteed to converge to the global optimum.

How can compositionality help? The *key insight* is that the compositional form of the probability distribution means that we can learn the parameters separately provided the correspondence is known. But if this is not known, then we can still learn the parameters for different parts separately which greatly simplifies the assignments. For example, to learn the distribution $f(x_{a(1)} - x_{a(2)}|\lambda_{12})$ we only need deal with the possible assignments of $x_1$ and $x_2$. *We do not need to consider the assignments* of all the leaf nodes of the model, which we would have to do if we are using the EM algorithm.

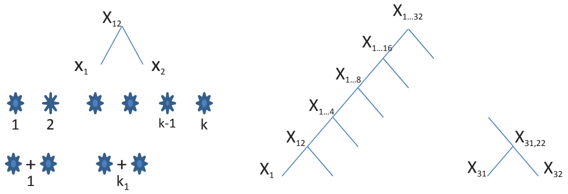

Figure 2. The compositional structure makes it practical to search over all assignments between the model and the data. For the level-1 part-models, e.g., like the one with variables $x_1, x_2, x_{12}$ (top left), we need to search over $O(k^2)$ possible assignments, where $k$ is the number of datapoints in the image represented here by stars (bottom left). But for a level-2 part-model, constructed by combining two level-1 part-models, we need only search over $O(k_1^2)$ assignments where $k_1$ is the number of level-1 parts detected in the image (bottom left) – and often $k_1 << k$. If, by contrast, we tried to search for all assignments with exploiting compositionality we would be faced with exorbitant cost of $O(k^{32})$ for a model with 32 leaf nodes. Instead of $O(16 \times k^2 + 8 \times k_1^2 + 4 \times k_2^2 + 2 \times k_3^2 + k_4)$.

This argument extends to the more challenging situations when the structure of the model is unknown, when the data may contain one of several different models, and if the data also contains random data generated by a background process. We do assume, however, that the probability models can be expressed in compositional form, or in terms of a mixture of distributions of hierarchical form.

To exploit this insight, we describe a breadth first search through the space of probability models. This search decreases an overall fitness function although it is not guaranteed to converge to the optimal solution.

## 6. Breadth-First Model Search in Compositional Learning

The compositional strategy is to search through the space of all models that have compositional form – and hence obey the property that we can learn different parts of the model separately. In this section, for simplicity, we will drop the assignment variable $A$ with the convention that it is always used (i.e. that we will have to search over all assignments).

We start with a default model which generates the data by a uniform distribution $U(x) = \prod_i U(x_i)$. This gives a default encoding for the image which we can evaluate by computing $P(D) = \prod_{\mu=1}^N \prod_i U(x_i^\mu)$. We proceed by the following strategy, see figure (3).

Next we search for compositions of pairs of points that happen frequently together in the image by a clustering technique described in the next section (we use distributions defined over pairs of points for simplicity, but note that L. Zhu *et al* [13],[11] built compositions out of triples). This outputs a set of $N_1$ level-1 part-models of

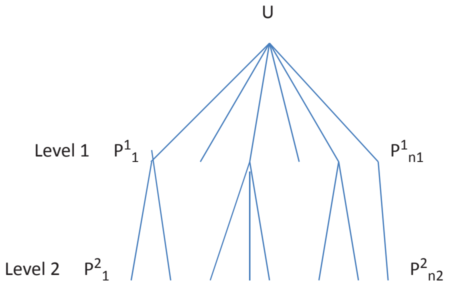

Figure 3. The breadth-first model selection strategy. We start with a default model $U$. Then we generate a set of level-1 part-models $P^1_1, ..., P^1_{N_1}$ each of which are converted into level-1 image-models and hence give $N_1$ alternative ways to encode the images. These level-1 part-models are expanded by composition to give a set of $N_2$ level-2 part models $P^2_1, ..., P^2_{N_2}$ which, in turn, are converted into level-2 image-models. Each level-1 and level-2 image-models are required to encode the image better than their parents. The procedure stops automatically when it fails to find higher level part-models with better encodings of the data.

form $P(x_1, x_2|x_{12}) = f(x_1 - x_2|\lambda_{12})\delta(x_{12} - \frac{x_1+x_2}{2})$. We convert each level-1 part-model into a level-1 image-model for the entire data as follows (see the next section for more details). We assume that each image consists of a number of examples of the part – i.e. generated by sampling from $P(x_1, x_2|x_{12})$ – while the remaining data points are generated by the uniform distribution $U(x)$. It follows, from the clustering technique, that each level-1 image-model gives a better encoding of the data than the default model. Hence the output of the first stage is a family of $N_1$ level-1 image-models each associated with a single level-1 part-model. Note these level-1 part models correspond to the level-1 dictionaries described in [13],[11].

Next we expand these level-1 image models to level-2 image models as follows. For each level-1 image model we search for compositions of its level-1 part model with other level-1 parts, using the same clustering technique as before. I.e. to find compositions of level-1 models $P(x_1, x_2|x_{12})$ and $P(x_3, x_4|x_{34})$ we perform clustering over $1/2(x_1 + x_2) - 1/2(x_3 + x_4)$, which will yield a level-2 part-model $P(x_{12}, x_{34}|x_{1234})$. As before, we supplement these level-2 part-models with a uniform distribution to obtain a level-2 image model. The total number of level-2 image-models is bounded by $N_1^2/2$ but will be smaller in practice because we fail to find compositions for some part-models. Each level-2 image-model is associated to one level-2 part-model and one, or two, level-1 part-models. As before, the clustering technique will ensure that all these level-2 image models give a better of encoding of the data than the level-1 image-models that they are based on.

This procedure repeats until we fail to find any new clusters. The output is a set of image-models and their associated part-models.

For some applications, for example learning models of multiple objects [11], we simply output the part-models. The part-models at the highest levels correspond to models of objects. The part-models at lower levels correspond to parts which are typically shared between several objects.

For other applications, such as learning an object model for a single object [13] we select the highest level part-model and combine it with additional part-models provided this improves the overall fit of the model to the data.

## 7. Clustering in Compositional Learning as Encoding of Data

The compositional learning strategy is to detect part-models starting with simple compositions, at level-1, and proceeding to more complex compositions at higher levels. These part-models are associated with image-models as described in the previous section. These compositions are obtained by a clustering process which will be described in this section. New part-models are created provided that their associated image-model gives a lower cost of encoding the data.

The basic ideas can be illustrated by learning a single level-1 part-model and its associated image-part model. Recall that a level-1 image part-model is of form $f(x_1, x_2|\lambda_{12})$. Intuitively a good part-model is one for which we can identify clusters of pairs of points $(x^\mu_{a(1)}, x^\mu_{a(2)})$ and parameters $\lambda$ such that $f(x^\mu_{a(1)}, x^\mu_{a(2)}|\lambda)$ is high for all $\mu$. In other words, we can find instances of this part-model in all images. Later we learn a probability distribution $P(\alpha)$ for the number of instances $\alpha^\mu$ in image $\mu$.

We claim that identifying these clusters can be done by standard clustering algorithms. Moreover, the criterion for a good cluster is precisely the condition that the associated image-model gives a lower cost for encoding the data than the default background model.

To make this concrete, we restrict $f(x_1, x_2|\lambda)$ to be a Gaussian model in the relative position $x_1 - x_2$. In this case $\lambda = (m, \Sigma)$, where $m$ and $\Sigma$ are the mean and covariance.

Standard clustering algorithms, will output sets of pixel pairs $(x^\mu_{a(1)}, x^\mu_{a(2)})$ such that:

$$(x^\mu_{a(1)} - x^\mu_{a(2)})^T \Sigma^{-1} (x^\mu_{a(1)} - x^\mu_{a(2)}) < T, \qquad (10)$$

where $T$ is a threshold. This corresponds precisely to finding sets of points such that $f(x^\mu_{a(1)}, x^\mu_{a(2)}|\lambda)$ is large. In other words to finding examples of the part-models.

The output is a set of clusters. Each cluster corresponds to a part-model – i.e. a value $\lambda$ – together with the set of data points that are assigned to it. This set of data points specifies the set of assignments of the part-model to the data. At this stage, we make no restriction on the possible assignments – for example, data points may be assigned to several different part-models. This relates to our earlier claim that finding assignments is easier for sub-parts of the object. Consistency between these assignments will be imposed at higher levels when we seek composite part-models which combine more elementary parts. This is similar to dynamic programming on trees for detecting objects after learning, see [13], which starts by searching for positions of object sub-parts at lower levels and then imposes consistency at higher levels, similar to constraint satisfaction.

For the level-2 models – we cluster to obtain models which have sets of points $(x^\mu_{a(1)}, x^\mu_{a(2)}, x^\mu_{a(3)}, x^\mu_{a(4)})$ and parameters $\lambda_{1234}$ such that $f(x^\mu_{a(1)}, x^\mu_{a(2)}, x^\mu_{a(3)}, x^\mu_{a(4)}|\lambda_{1234})$ is high. The pairs of points $(x^\mu_{a(1)}, x^\mu_{a(2)})$ and $(x^\mu_{a(3)}, x^\mu_{a(4)})$ are those data points assigned to the level-1 part-models $P(x_1, x_2|\lambda_{12})$ and $P(x_3, x_4|\lambda_{34})$ respectively. When composing these points we check for consistency of assignments.

Now we present an alternative perspective that shows the connection between clustering and encoding. This starts by relating part-models to image-models. It also helps illustrate the relationship to standard EM approaches.

We associate a part-model to an image-model as follows. Recall that the default image model is a uniform distribution $U(x) = \prod_i U(x^\mu_i)$. For each part model, we allow a probability $P(\alpha)$ for the number of occurrences of the part in each image. This gives a generative model:

$$P(\alpha)\prod_{j=1}^{\alpha} U(x_j) f(x'_j - x_j|\lambda) \prod_{i\neq\{j:j=1,\dots,\alpha\}} U(x_i). \quad (11)$$

Intuitively, we select a number $\alpha$ of part instances in each image. These correspond to pairs of points $\{(x_j, x'_j) : j = 1, \dots, \alpha\}$. We assume that the remaining points $\{x_i\}$ (and the point $x'_j$ for each pair $(x_j, x'_j)$) are generated by the uniform distribution.

Now imagine introducing the assignment variables. We can use a probability model of form equation (11) to generate the data $D = \{x^\mu\}$. The task is to estimate $\lambda$, the distribution $P(\alpha)$ of instances, and the assignments.

Applying the EM algorithm, however, will only give us one solution and moreover will have multiple minima corresponding to all the part-models that happen in the data. Instead we want to find a large set of possible solutions which include all the minima, but which can have some 'false positives' (which can be removed later). Clustering enables us to achieve this.

Moreover, we can compute the cost of encoding the data by the default background model and by image-models of the form given in equation (11). After some algebra, we find that the condition that the image-model encodes the data better than the default model is precisely the clustering threshold condition. This follows from the requirement that the image-model predicts the data better than the default model – and hence has lower encoding cost – is given by:

$$\prod_\mu P(\alpha^\mu) \prod_{j=1}^{\alpha^\mu} f(x_j - x_{j'}|\lambda) \prod_{j\neq\{i=1,\dots,\alpha^\mu\}} U(x^\mu_i) > \prod_\mu \prod_i U(x^\mu_i), \quad (12)$$

which reduces to the condition:

$$\sum_\mu \sum_{j=1}^{\alpha^\mu} \frac{(x^\mu_j - x^\mu_{j'} - m)^2}{2\sigma^2} \leq \sum_\mu \log P(\alpha^\mu) + \sum_\mu \alpha_\mu \log K - \frac{1}{2}\sum_\mu \alpha_\mu \log(2\pi\sigma^2), \quad (13)$$

where $K$ is a normalization constant for the uniform distribution. Note if we estimate $P(\alpha)$ from the data, then the terms $\sum_\mu \log P(\alpha^\mu)$ will tend to $-N$ times the entropy of the distribution.

Observe that equation (13) reduces to the clustering condition.

To summarize, clustering enables us to find image models which describe the data better than the default background model. We proceed by performing higher order clustering to determine level-2 part models. At this stage we check for consistency of assignments of the part models. As before, we augment these part-models with uniform distributions to describe the rest of the data. These level-2 image-models are selected only if they give a better encoding of the image than the level-1 image-model that they are grown from.

## 8. Limitations of Compositional Learning

Compositional learning exploits the fact that different parts of the object model can be learnt independently. It proposes to learn these models by clustering so that we deal with the assignment variables first for the simple low-level parts and only later for the more complex high-level parts. We do a breadth first search in the space of models so we do not need to impose consistency between the assignments of the low-level part models. Consistency is imposed later when we construct new models by composing low-level parts to make more complex parts.

The learning procedure assumes that we can use clustering to determine the part-models. These assumptions will only hold in certain conditions and could be analyzed by the same analysis used in [10] to determine the errors which result from using an approximate model to perform inference. For example, it may be possible to prove that compositional learning will work for certain distributions of the object structures and the background but will be impossible otherwise. This may be similar to the order parameter results obtained in [10] which specified whether it was possible to detect a target in the presence of background clutter.

Here we list some of the problems which may arise and discuss their potential severity. (I) The clusters may be contaminated. But limited contamination may not matter and may only cause mild bias. (II) Fake clusters caused by accidental coincidences of random points in the background. These should not cause any problem because they can be removed later because they will not to be composed to form higher level parts. (III) Clusters caused by higher-order regularities, such as correlations induced between $x_1$ and $x_3$, which should also be removed at higher levels. (IV) Merged clusters. These are more serious because we cannot easily separate them into the correct clusters. (V) Superimposed clusters – i.e. two parts which are very similar may overlap. This is probably okay because it means we reduce our vocabulary of models by treating them as the same.

Observe that the approach in this paper has formulated computational learning in terms of parameter estimation. An alternative, but related approach, is to treat it as encoding in the sense of rate-distortion theory [4].

## 9. Discussion

It is important to distinguish our compositional approach from more standard methods such as Minimum Description Length (MDL). Consider learning the level-1 part-models as described in section (6). An MDL approach would attempt to encode the data in terms of a mixture of Gaussians and solve this task using one of the new generation of efficient algorithms for fitting mixtures of Gaussians to data such as [1].

But we choose not to do this. Instead we follow Mao's exhortation to ”let one hundred flowers bloom”. To be more specific, when we learn our level-1 part-models we do not want to find all clusters simultaneously in order to optimize a global criterion. We do not want to impose optimality at this stage. Instead we seek evidence for object parts and are willing to pay the price for redundancy and allow data to be assigned to more than one cluster. Rather than finding a single encoding of the data in terms of level-1 models we seek instead to find a *family of different ways to encode the data*, see figure (3). Recall that each level-1 model gives a way to encode of the data – the corresponding level-1 image-model – which represents the data in terms of the level-1 part-model and the background model.

There are several reasons why we wish to avoid optimality at this stage. But, most importantly, recall that we are clustering on *relationships between data points rather than on the data points themselves*. Hence a specific data point may be related to several different clusters or to none. We want our algorithm to be able to explore several possibilities without being constrained by premature optimality. They can be resolved later as we proceed up the hierarchy by searching for higher level part-models.

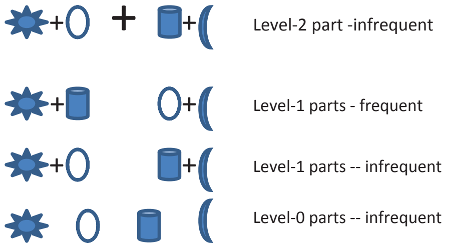

Figure 4. Why we need breadth-first model search. Consider a set of images that contain instances of: (i) a level-2 part-model (first row) containing level-1 submodels represented by a star-ellipse and a cylinder-crescent (third row) which occurs infrequently, (ii) two level-1 part models star-cylinder and ellipse-crescent (second row) that occur frequently, and (iii) isolated data points star, ellipse, cylinder and crescent which occur infrequently (or frequently). If we select the level-1 part-models using an MDL principle then we risk failing to detect the star-ellipse and cylinder-crescent because they occur infrequently and their instances may be better explained by the more frequent star-cylinder and ellipse-crescent part-models supplemented by isolated examples of star, ellipse, cylinder and crescent (i.e. from the default background model used to fill-in the datapoints which are not generated by the level-1 and level-2 part-models. It is only when we search for the level-2 part-models that we realize the advantages of encoding the star-ellipse and cylinder-crescent as part-models because they can be composed to form a level-2 part-model which occurs in the images.

We re-emphasize that an important advantage of our compositional learning approach is that it avoids the need to search over all matching assignments for large object models (e.g., ones with many nodes) which is not feasible for large numbers of nodes. Instead we only have to search over assignments for level-1 part models, then over the assignments of these part models to level-2 part models, and so on. This exploits the compositional structure and avoids the combinatorial explosion in assignments which would occur for more traditional models and learning algorithms.

## 10. Conclusion

This paper has sketched a theory of compositional learning which provides theoretical underpinnings for successful experimental works such as [13],[11]. The theory clarifies and exploits the basic property of compositional models – that their different parts can be learnt independently provided the correspondence is known. Compositional learning exploits this property by proposing a breadth first search through the space of models which starts by finding models of sub-parts and proceeds by combining them together to create more complex parts. This search is performed using clustering techniques to identity part-models. The clustering conditions ensure that these part-models, together with default models for the background, give good encoding of the data.

## Acknowledgements

I would like to thank George Papandreou, Xingyao Ye, Roozbeh Mottaghi, Xiaobai Liu and an anonymous referee for helpful feedback. I would also like to thank Stuart Geman and YingNian Wu for discussions about compositional models. Nam-Gyu Cho helped with the figures. This work was supported by the U.S. Office of Naval Research under the MURI grant N000141010933, by the National Science Foundation with grant NSF NSF0917141 and by the Korean Ministry of Education, Science, and Technology, under the National Research Foundation WCU program R31-10008.

## References

[1] A. P. Benavent, F. E. Ruiz, and J. M. Sáez. Learning gaussian mixture models with entropy-based criteria. *IEEE Transactions on Neural Networks*, 20(11):1756–1771, 2009.

[2] S. Fidler, M. Boben, and A. Leonardis. Similarity-based cross-layered hierarchical representation for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[3] Y. Jin and S. Geman. Context and hierarchy in a probabilistic image model. In *CVPR (2)*, pages 2145–2152, 2006.

[4] Y. Ma, S. Member, H. Derksen, W. Hong, J. Wright, and S. Member. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3, 2007.

[5] B. Ommer and J. M. Buhmann. Learning compositional categorization models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[6] S. D. Pietra, V. J. D. Pietra, and J. D. Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):380–393, 1997.

[7] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[8] C. Shannon. Prediction and entropy of printed english. *The Bell System Technical Journal*, 30:50–64, 1951.

[9] Y. Wu, Z. Si, H. Gong, and S. Zhu. Learning active basis model for object detection and recognition. *Intl. Journal of Computer Vision (IJCV)*, 2009.

[10] A. L. Yuille, J. Coughlan, Y. Wu, and S. Zhu. Order parameters for detecting target curves in images: When does high-level knowledge help? *International Journal of Computer Vision*, 41(1/2):9–33, 2001.

[11] L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille. Part and appearance sharing: Recursive compositional models for multi-view multi-object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] L. Zhu, Y. Chen, and A. L. Yuille. Unsupervised learning of probabilistic grammar-markov models for object categories. In *Transactions on Pattern Analysis and Machine Intelligence*, pages 114–128, 2009.

[13] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[14] S. C. Zhu, Y. N. Wu, and D. Mumford. Minimax entropy principle and its application to texture modeling. *Neural Computation*, 9(8):1627–1660, 1997.