Rates for Inductive Learning of Compositional Models

Adrian Barbu[1]**, Maria Pavlovskaia**[2]**, Song Chun Zhu**[2]
[1]Department of Statistics, Florida State University, Tallahassee, Florida 32306, USA
[2]Department of Statistics, UCLA, Los Angeles, CA 90095, USA

Abstract

Compositional Models are widely used in Computer Vision as they exhibit strong expressive power by generating a combinatorial number of configurations with a small number of components. However, the literature is still missing a theoretical understanding of why compositional models are better than flat representations, despite empirical evidence as well as strong arguments that compositional models need fewer training examples. In this paper we try to give some theoretical answers in this direction, focusing on AND/OR Graph (AOG) models used in recent literature for representing objects, scenes and events, and bringing the following contributions. First, we analyze the capacity of the space of AND/OR graphs, obtaining PAC (Probably Approximately Correct) bounds for the number of training examples sufficient to guarantee with a given certainty that the model learned has a given accuracy. Second, we propose an algorithm for supervised learning AND/OR Graphs that has theoretical performance guarantees based on the dimensionality and number of training examples. Finally, we observe that part localization, part noise tolerance and part sharing leads to a reduction in the number of training examples required.

Introduction

PAC learning concerns mathematical bounds for the number of training examples $N(\epsilon, \delta)$ sufficient to obtain a maximum error $\epsilon > 0$ in learning a concept, at a confidence level $1 - \delta$. Such PAC learning bounds have been previously derived for k-CNF (conjunctive normal form) or k-DNF (disjunctive normal form) logical expressions (Valiant 1984), finite automata (Angluin 1988; Kearns and Valiant 1994) and regular expressions (De La Higuera and Oncina 2005) with rather discouraging results requiring unrealistically large numbers of training examples that don't match our intuition.

In the Computer Vision literature people argue for the use of compositional models (Bienenstock, Geman, and Potter 1997; Jin and Geman 2006; Zhu and Mumford 2006), which are hierarchical representations of complex objects through reusable and localized parts, and observed experimentally (Tang et al. 2010) that sharing parts between objects results in better model accuracy and require fewer training examples. However, so far there has not been a mathematical analysis of these findings and no theoretical reasons why hierarchical models should be preferred to flat ones.

In this paper we will study hierarchical AND/OR graph (AOG) representations (Chen et al. 2007; Si and Zhu 2012; Wu et al. 2009; Zhu and Mumford 2006), which are general forms of compositional models. We will characterize the space of AOG graphs by certain quantities (d, b_a, b_o, n) such as maximum depth d, maximum branching number b_a, b_o at AND/OR nodes respectively and number of primitives n. As opposed to the regular grammars, the AOG graphs are not recursive and usually have small depths (e.g. 5). The AND nodes represent the composition of a part or object from its elements while the OR nodes represent alternative configurations. The primitives are the basic elements of the AOG and they are quite general: they can be filter responses, sketches or trained classifiers.

Our study leads to answers for the following claims:

1. The capacity of the space of AOGs is much smaller than the capacity of the k-CNF or k-DNF space. We will see that the capacity directly relates to the number of training examples sufficient for learning a concept.
2. Part localization, part noise tolerance and part sharing between categories have a positive effect on reducing the number of training examples for learning an object.
3. Experiments for supervised learning of parts, AOGs from parts and AOGs directly from images and comparisons between learning from images and learning from parts.

Even though the conclusions we derive may sound familiar, it is the first quantitative work that provides a theoretical foundation to the study of compositional models.

The AOG and Hypothesis Spaces

The **AOG** is a hierarchical representation, used to represent objects through intermediary concepts such as part templates. It is the basis of the generative image grammar (Zhu and Mumford 2006).

The AOG is defined on an **instance space** $\Omega = \{0, 1\}^n$. The space Ω represents all possible binary responses of n Boolean functions $t_i : I \to \{0, 1\}, i = \overline{1, n}$ called **terminal nodes** on the current image I. The terminal nodes are binary responses of some image filters. For example, a terminal node could be the response of an Adaboost classifier or a thresholded Gabor filter response (Wu et al. 2009)

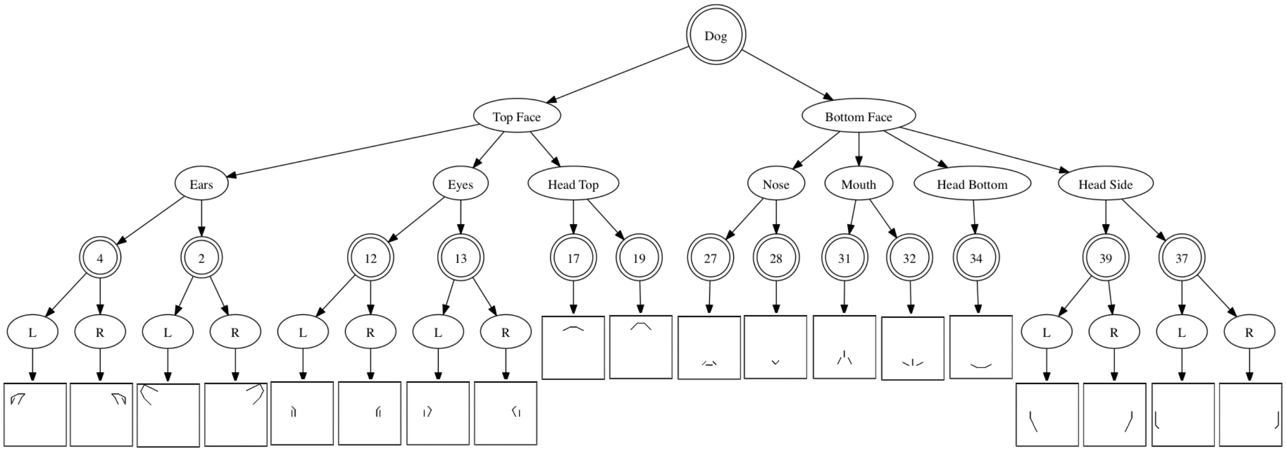

Figure 1: An example of an AND/OR graph of a dog where some parts (ears, eyes, nose, mouth) have alternative appearances.

at a certain position and orientation. The **detection process** is defined as the vector of terminal node responses $\mathbf{t} : I \to \Omega = \{0,1\}^n$.

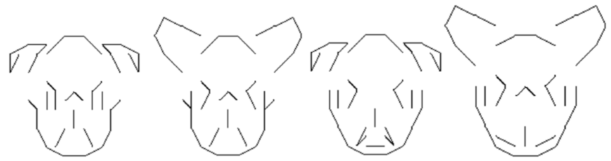

Figure 2: Samples from the dog AND/OR graph from Figure 1.

The AOG is a boolean function $g : \Omega \to \{0,1\}$ and it can be used to construct an object detector $g_0 : I \to \{0,1\}, g_0 = g \circ \mathbf{t}$ where $\mathbf{t}$ is the detection process. An example of an AOG for a dog face is shown in Figure 1. The dog face is composed of a number of part templates that are themselves represented as a combination of the 18 types of sketches (terminal node responses) shown in Figure 3 left.

The AOG boolean function $g : \Omega \to \{0,1\}$ is constructed from AND and OR nodes. An AND node represents the composition of its elements and has value 1 if and only if all its elements are present (i.e. have value 1). For example a frontal face model could consist of an AND of four parts: two eyes, a nose and a mouth. The OR nodes represent deformations or alternate configurations of the elements. For example the dog faces in Figure 1 have two types of ears, obtaining the ear concept using the OR of the two ear types.

An AOG can be extended to a Bernoulli AOG by replacing the AND/OR logical operations with noise-tolerant versions based on majority voting (e.g. a 8 child AND is replaced with a 6 out of 8 majority), or even to a real AOG by replacing the AND/OR operations with real value equivalents (e.g. the AND is replaced by a logistic classifier, the OR by a soft-max). It is easy to see that the real AOG generalizes Boosting and logistic regression, thus its PAC analysis is more complex than the bounds for linear classifiers (Schapire et al. 1998; Kakade, Sridharan, and Tewari 2008).

This is why, in this work we take a first step and restrict our attention to Boolean and Bernoulli AOG, leaving the real AOG for a later study.

Some parameters characterizing an AOG are: its depth d representing the number of AND and OR layers, the maximum branching numbers b_a for the AND nodes and b_o for the OR nodes and the maximum number n of values of the terminal nodes. The AND and OR layers must alternate because two consecutive layers of the same type can be merged into a single layer. This is why the depth d is defined as the number of pairs of AND/OR layers.

For example, the AOG from Figure 1 has parameters $d = 3, b_a = 4, b_o = 2, n = 19$. In general, object AOGs have small depth (at most 5) with branching numbers b_a on the order of 3 to 5 and b_o on the order of 5 to 7.

We define a **concept** as any subset $G \subset \Omega$ and for any function $g : \Omega \to \{0,1\}$, we define $\Omega_g = \{x \in \Omega,\ g(x) = 1\}$. Learning the concept G means finding $g : \Omega \to \{0,1\}$ such that $\Omega_g = G$. This boolean function g can be a logical, Bernoulli or real AOG, depending on the type and level of noise contained in the concept G. For an AOG g, Ω_g is the **language** of g. Because $\Omega = \{0,1\}^n$ is finite, the concept Ω_g is a finite set. We will denote by Ω_g^I the set of all images that generate terminal node responses from Ω_g. Thus Ω_g^I can be considered the **visual language** of g.

Define the **hypothesis space** $\mathcal{H}(d, b_a, b_o, n) \subset 2^\Omega$ of the AOG as the space of all concepts that can be represented by AOGs with maximum depth d, branching numbers b_a, b_o for the AND/OR nodes respectively and instance space $\Omega = \{0,1\}^n$. To measure how far a concept G is from a learned AOG $g : \Omega \to \{0,1\}$, define for any pdf μ over Ω the **error** as

$$err_\mu(g, G) = \mu(G \Delta \Omega_g) = \mu(G - \Omega_g) + \mu(\Omega_g - G) \quad (1)$$

where $A \Delta B = (A - B) \cup (B - A)$ is the symmetric difference between sets A and B.

Capacity and Bounds for Supervised Learning AOGs

Because the instance space $\Omega = \{0,1\}^n$ is finite due to the quantization of the possible terminal node responses, the hypothesis space $\mathcal{H} = \mathcal{H}(d, b_a, b_o, n) \subset \{0,1\}^\Omega$ is also finite, so the following theorem (Blumer et al. 1987) applies

Theorem 1 *(Haussler) Let $\mathcal{H}$ be a finite hypothesis space and $\epsilon, \delta > 0$. If the number of training examples is*

$$N(\epsilon, \delta) \geq \frac{1}{\epsilon}(\ln |\mathcal{H}| + \ln \frac{1}{\delta}) \quad (2)$$

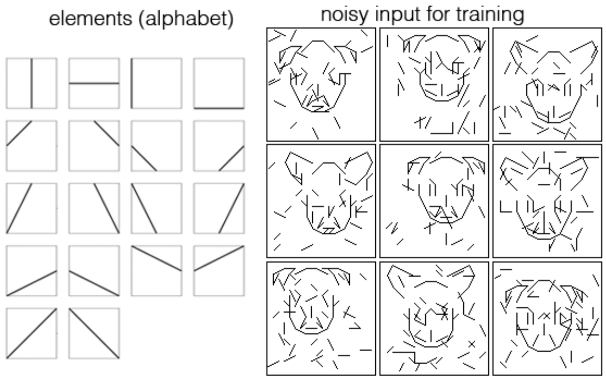

Figure 3: Left: Atomic elements (sketches) that can be placed at different positions to construct binary images. Right: Training images exhibiting occlusion, clutter and random object position.

then for any pdf μ and any hypothesis $h \in \mathcal{H}$ that is consistent with all training examples from G we have that $P(err_\mu(h, G) < \epsilon) > 1 - \delta$.

Based on this theorem, we define the **capacity** of an AOG space as $C(d, b_a, b_o, n) = \ln |\mathcal{H}(d, b_a, b_o, n)|$, where $\mathcal{H}(d, b_a, b_o, n)$ is the AOG hypothesis space of depth d, breadth b_a, b_o for the AND/OR nodes and n terminal nodes. Using eq. (2) we can directly relate the number of training examples with the capacity of the hypothesis space $\mathcal{H}$.

Proposition 1 *The capacity of the space $\mathcal{H}(d, b_a, b_o, n)$ can be bounded as*

$$C(d, b_a, b_o, n) \le (b_a b_o)^d \ln n \tag{3}$$

Proof. We compute $N(d, b_a, b_o, n) = |\mathcal{H}(d, b_a, b_o, n)|$ recursively. Consider all AOGs of depth $d + 1$. Then the top OR node has at most b_o branches. Each branch is an AND node with at most b_a children of depth at most d. As each of these children can have at most $N(d, b_a, b_o, n)$ variants, the number of variants of each branch of the top OR node is at most $N(d, b_a, b_o, n)^{b_a}$, hence we have

$$N(d + 1, b_a, b_o, n) \le N(d, b_a, b_o, n)^{b_a b_o}$$

By recurrence we get that

$$N(d, b_a, b_o, n) \le N(1, b_a, b_o, n)^{(b_a b_o)^{d-1}}$$

By the same argument we have that $N(1, b_a, b_o, n) \le n^{b_a b_o}$ so we obtain the desired result. □

A weaker bound can be obtained by treating the AOG as a feed-forward neural network. It is known (Kearns and Vazirani 1994) that the VC dimension of the space $N_{d,s}$ of neural networks with at most s nodes, each with VC dimension d, is $VC(N_{s,d}) < 2ds \log(es)$.

Example 1. Assume that we want to learn an object such as those in Figure 2, of size $15 \times 15 \times 18$. This means the object has elements on a 15×15 grid with 18 possible elements at each grid position. The model is an AOG of depth $d = 1$, which is the OR of a large number of AND templates each having at most 50 elements, thus $b_a = 50$. This is exactly the 50-DNF with $n = 4050$ literals (a literal is the presence of a sketch at any of the 225 possible positions with any of the 18 possibilities), so it has a size of at most $2^{4050^{50}}$ and a capacity of $\approx 10^{180}$.

Example 2. Suppose we want to learn the same object with an AOG of depth $d = 2$ with $b_a = b_o = 5$, with the same terminals, thus $n = 4050$. The class of such AOGs has a capacity of $C(2, 5, 5, 4050) \le 25^2 \ln 4050 \approx 5192$ which is much smaller than the 50-DNF from Example 1.

The 50-DNF from Example 1 represents the object as the union of all possible object images, which is impractical. The space of AOGs from Example 2 represents a space of pattern used to generate the object, trying to divide the object variability into a number of hidden factors.

Capacity of the AOG with Localized Parts. The capacity obtained in the previous section can be further reduced if we take into account the part locality. Usually parts are composed of a number of elements that are spatially located near each other. Thus even though the terminal nodes can in principle be any of the n elements, we can assume that the parts are localized, thus for example the terminal nodes for a part can be one of l elements close to the first terminal, which can be any of all n elements. In this case we have

Proposition 2 *The capacity with localized parts can be bounded as*

$$C(d, b_a, b_o, n, l) \le b_a^{d-1} b_o^d \ln(n l^{b_a - 1}) \tag{4}$$

Proof. The same recurrence formula holds as in Prop. 1. Assuming locality we get that $N(1, b_a, b_o, n, l) \le (n l^{b_a-1})^{b_o}$ since in an AOG of depth 1, there are b_o OR nodes, each OR node having at most $n l^{b_a-1}$ versions. □

Example 3. Assume that we want to learn the same template as in Example 2 but assuming locality for the parts with $l = 450$, thus all part terminal are in a 5×5 window that can be anywhere. In this case the capacity is $C(2, 5, 5, 4050, 450) \le 5 \cdot 5^2 \ln(4050 \cdot 450^4) \approx 4093$ which is smaller than 5192 from Example 2.

The computations from Examples 1, 2 and 3 indicate that the space of AOGs $\mathcal{H}(d, b_a, b_o, n)$ is much smaller than the space of k-DNF for practical applications, so the number of training examples required for learning is also greatly reduced.

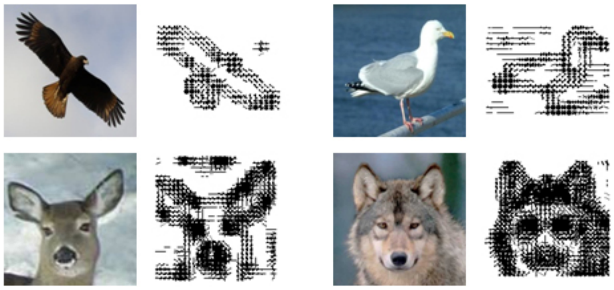

Figure 4: Real images and thresholded Gabor filter responses.

Supervised Learning of AOGs

Supervised learning of AOGs assumes that the the object and its parts have all been manually delineated. This does not completely specify the AOG since the same part could have different appearances in different images, and the learning must be able to separate them into a number of OR branches.

To formalize the learning problem, we assume we are given a number of binary images $\mathbf{x}_i \in \{0, 1\}^n, i = 1, ..., m$, where 2D or 3D images have been transformed to vectors.. To simplify the setup, we assume that in all images the objects have the same size but can be at different locations.

The ideal image will have zeros except at the object locations where it is 1. However, due to noise we assume that each bit can switch to the other value with probability q (Bernoulli noise). Examples of such noisy binary images are shown in Figure 3, right. This setup mimics the binary images obtained by thresholding Gabor filter responses such as those shown in Figure 4. Even though this binary setup seems like a simplification, it keeps most of the challenges existent in real images (occlusion, clutter) and is very similar to the setup in the Active Basis framework (Wu et al. 2009).

Part Learning using Two-Step EM

Each image can serve as a training example for many parts. In the AOG each node represents a concept and in the supervised case they can be learned recursively starting from the bottom. Since the parts have been delineated (e.g. by bounding boxes), each part can learned separately from the training examples as the OR of a number of possible appearances, each being the AND of a number of terminal elements.

In the rest of this section we will show how to learn one part with performance guarantees. Based on the manual annotation, the part examples are cropped from the training images and need to be clustered based on their appearances into a number k of clusters, specified by the user.

Depending on the problem, the cropped image dimensions might be naturally clustered into one or more of groups. Working with each group separately, we can assume that all part images to be clustered have the same dimension d. Thus we have a number of training examples $\mathbf{x}_1, ..., \mathbf{x}_m \in \{0,1\}^d$.

Let $D(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1$ be the Hamming distance. We assume that the part images are generated as a mixture of k true part templates $\mathbf{P}_1, ..., \mathbf{P}_k$ corrupted by Bernoulli noise with level q. Let $c = \min_{i \neq j} D(\mathbf{P}_i, \mathbf{P}_j)/d$ be the separation between the true part templates.

In this case learning can be done with the two-step EM algorithm 1. The pruning step 6 picks the first template $T_i^{(1)}$ randomly, the second one furthest from the first one, the third template as the one at largest distance from the set containing the first two templates, and so on until there are k templates.

Assume that the following conditions hold:

C1: $c > \max(\frac{16q}{3+2q}, \frac{8q}{3(1-2q)})$

C2: $d(1-2q) > \max(24, \frac{2}{cB} \ln \frac{8l}{c})$

C3: $dc^2(1-2q)^2 > 3456q \ln 8el$

C4: $mc^2(1-2q)^2 > 27648ql \left(\frac{1}{2} + \frac{\ln 2}{d} \right)$

where $B = \frac{1}{2}(1-2q) \ln \frac{1}{6q} > 0$.

Let $E = \min \left(\frac{1}{2}, \frac{\frac{3}{4}c(1-2q) - 2q}{c(1-2q) + 2q} \right)$. It was proved in (Barbu, Wu, and Zhu 2013) that:

Theorem 2 *Let m examples be generated from a mixture of k binary templates under Bernoulli noise of level q and $w_i > w_{min}$ for all i. Let $\epsilon, \delta \in (0,1)$. If conditions $C1 - C4$ hold and in addition the following conditions hold*

1. *The initial number of clusters is $l = \frac{12}{w_{min}} \ln \frac{2}{\delta w_{min}}$.*
2. *The number of examples is $m \geq \frac{8}{w_{min}} \ln \frac{12k}{\delta}$.*
3. *The separation is $c > \frac{8}{dB} \ln \frac{5n}{\epsilon w_{min}}$.*
4. *The dimension is $d > \max \left(\frac{3}{qE^2} \ln \frac{18m^2}{\delta}, 2 \ln \frac{12k}{\delta} \right)$.*

Then with probability at least $1 - \delta$, the estimated templates after the round 2 of EM satisfy:

$$D(\mathbf{T}_i^{(2)}, \mathbf{P}_i) \leq D(mean(S_i), \mathbf{P}_i) + \epsilon q$$

where S_i are the training examples coming from the mixture component $\mathbf{P}_i$.

This means the the two step EM does almost as well as if we knew the cluster assignments for all training examples.

Algorithm 1 Two-step EM for Learning Bernoulli Templates

```
   Input: Examples $S = \{\mathbf{x}_1, ..., \mathbf{x}_m\} \subset \{0,1\}^d$
   Output: Part Templates $\mathbf{T}_i, i = 1, .., k$
 1 Initialize $\mathbf{T}_i^{(0)}$ as $l$ random training examples
 2 Initialize $w_i^{(0)} = 1/l$ and $q_0 \leq 1/2$ such that
       $q_0(1-q_0) = \frac{1}{2d} \min_{i,j} D(\mathbf{T}_i^{(0)}, \mathbf{T}_j^{(0)})$.
 3 E-Step: Compute for each $i = 1, ..., l$
       $f_i(\mathbf{x}_j) = q_0^{D(\mathbf{x}_j, \mathbf{T}_i^{(0)})}(1-q_0)^{d - D(\mathbf{x}_j, \mathbf{T}_i^{(0)})}, j = 1, ..., m,$
       $p_i^{(1)}(\mathbf{x}_j) = \frac{w_i^{(0)} f_i(\mathbf{x}_j)}{\sum_{i'} w_{i'}^{(0)} f_{i'}(\mathbf{x}_j)}, j = 1, ..., m$
 4 M-Step: Update $w_i^{(1)} = \sum_{j=1}^m p_i^{(1)}(\mathbf{x}_j)/m$ and
       $\mathbf{T}_i^{(1)} = \frac{1}{mw_i^{(1)}} \sum_{j=1}^m p_i^{(1)}(\mathbf{x}_j)\mathbf{x}_j$
 5 Pruning: Remove all $\mathbf{T}_i^{(1)}$ with $w_i^{(1)} < w_T = \frac{1}{4l}$
 6 Pruning: Keep only $k$ templates $\mathbf{T}_i^{(1)}$ far apart.
 7 Initialize $w_i^{(1)} = 1/k$ and $q_1 = q_0$.
 8 E-Step: Compute
       $f_i(\mathbf{x}_j) = q_1^{D(\mathbf{x}_j, \mathbf{T}_i)}(1-q_1)^{d - D(\mathbf{x}_j, \mathbf{T}_i)}, j = 1, ..., m$
       $p_i^{(2)}(\mathbf{x}_j) = \frac{w_i^{(1)} f_i(\mathbf{x}_j)}{\sum_{i'} w_{i'}^{(1)} f_{i'}(\mathbf{x}_j)}, j = 1, ..., m$
 9 M-Step: Update $w_i^{(2)} = \sum_{j=1}^m p_i^{(2)}(\mathbf{x}_j)/m$ and
       $\mathbf{T}_i^{(2)} = \frac{1}{mw_i^{(2)}} \sum_{j=1}^m p_i^{(2)}(\mathbf{x}_j)\mathbf{x}_j$
10 Round the elements of $\mathbf{T}_i^{(2)}, i = 1, ..., k$ to the nearest
   integer.
```

Noise Tolerant Parts

The part learned in the previous section consist of the mixture centers $\{\mathbf{T}_i\}_{\overline{1,k}}$, weights $\{w_i\}_{\overline{1,k}}$ and estimated noise $\hat{q}$. These components can be used to obtain the part probability from the mixture model:

$$p(\mathbf{x}) = \sum_{i=1}^k w_i \hat{q}^{D(\mathbf{x}, \mathbf{T}_i)}(1-\hat{q})^{d - D(\mathbf{x}, \mathbf{T}_i)}$$

$$p(\mathbf{x}) = (1-\hat{q})^d \sum_{i=1}^{k} w_i (\hat{q}/(1-\hat{q}))^{D(\mathbf{x},\mathbf{T}_i)}$$

Part detection in an image is obtained by restricting to all windows $\mathbf{x}$ of the same size d as the training examples, and comparing the mixture probability $p(\mathbf{x})$ with a threshold. This way the part representation is one AND/OR layer in a Bernoulli AOG. When there is a single mixture component ($k = 1$), comparing the probability with a threshold is equivalent to comparing the Hamming distance $D(\mathbf{x}, \mathbf{T})$ with a threshold.

The output of the part detection is a binary image with 1 at the locations where the part was detected and the rest 0. The probability threshold can be tuned to obtain detection images satisfying the Bernoulli noise assumptions that the probability q_1 of a switch from 1 (part) to 0 (no part) is the same as the probability of a switch from 0 to 1.

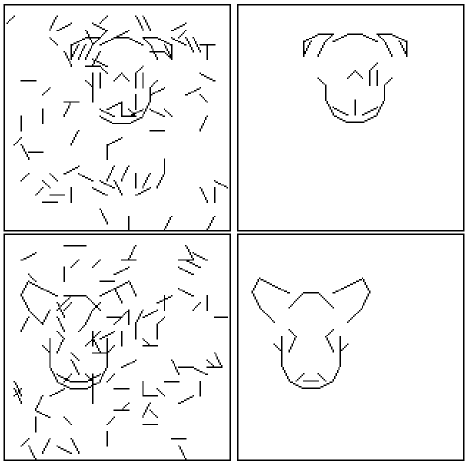

Figure 5: Large parts are more noise tolerant and can help reduce the noise level for object learning. Shown are two noisy binary images with $q = 0.1$ (left) and the part detection results (right) using noise tolerant parts.

In fact, if the part is large (e.g. $d > 5$), this new Bernoulli noise level q_1 is much lower than q, as illustrated in Figure 5. Assuming there is only one mixture component $\mathbf{T}$, the probability to detect the part using the threshold $D(\mathbf{x}, \mathbf{T}) \le k_0$ is $p_{11} = \sum_{i=0}^{k_0} \binom{d}{i} q^i (1-q)^{d-i}$ while the probability of a false detection is $p_{01} = \sum_{i=0}^{k_0} \binom{d}{i} q^{d-i}(1-q)^i$. In this case taking $k_0 = \lfloor d/2 \rfloor$ we obtain $q_1 = 1 - p_{11} = p_{01} = \sum_{i=0}^{\lfloor d/2 \rfloor} \binom{d}{i} q^{d-i}(1-q)^i$, the left tail of the Binomial distribution. For example if $d = 9$ and $q = 0.1$ we obtain $q_1 < 10^{-3}$.

Parts can be obtained in other ways than using the two-step EM, for example they can be borrowed from other objects or they can be learned in an unsupervised way as in (Fidler and Leonardis 2007; Zhu, Chen, and Yuille 2009). When the parts are borrowed from other objects at the same time the OR relationships between alternate part appearances can be borrowed. The parts, with or without the OR relationships can be transformed into noise tolerant versions using the Hamming distance and the mixture model, as described above.

Recursive Graph Learning

From the part detection we obtain a detection binary image for each part. These part detection images can be stacked together into a 3D binary image. Learning the next AND/OR layer of the AOG proceeds again using the two-step EM on these 3D binary images. The noise level can be considered as the largest noise level among the part detections. In general, learning in from the parts is easier because of noise level is smaller than the input level of noise q of the terminal elements.

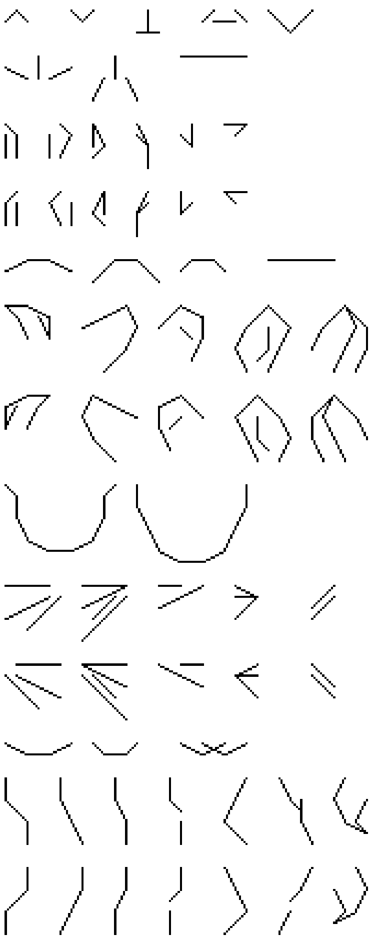

Figure 6: Each row shows one of the 13 noise tolerant parts used to train the AOG in scenario 1. Each part is the OR mixture of the AND templates shown in that row.

Experiments

We conduct an experiment that compares learning an object from parts vs. learning it directly from images.

Part Sharing Vs. Learning from Scratch

In this experiment we compare learning an object AOG from parts vs. learning it from images. Our goal is to show that part sharing across categories (transfer learning) can reduce the number of training examples $N(\delta)$.

We consider learning an AOG dog model from training examples under two scenarios.

1. The input noisy image is parsed into parts obtained from other objects. There are totally 13 noise tolerant parts

shown in Figure 6, among which the 8 parts that compose the dog model (ears, eyes, nose, mouth, head), borrowed from other animals. Part detection is run for each of these parts obtaining 13 detection images for each input image. Then the dog AOG is learned from the 3D binary image obtained from stacking the 13 binary detection images.

2. The dog model is learned directly from the training images by learning the parts first using the two-step EM algorithm described in the previous section.

The dog AOG from Figure 1 was learned from different numbers of training examples using both scenarios.

In the first scenario, the AOG was learned from the parts as a mixture with only one cluster ($k = 1$). The mixture center is obtained as the average of the binary images, as in step 4 of Algorithm 1. The mixture center was then made binary by rounding to the nearest integer.

In the second scenario, the dog parts were learned first using two-step EM, then the AOG was learned from the dog parts in a similar way to scenario 1.

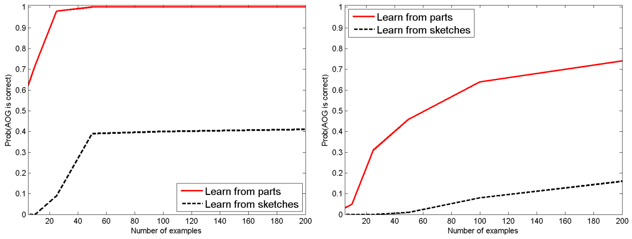

Figure 7: The percentage of times the dog AOG was learned correctly vs number of training examples for two noise levels. Left: $q = 0.1$, right: $q = 0.2$

In the two images of Figure 7 are shown the percentage of times the entire dog AOG was learned correctly (thus the error is $\epsilon = 0$) vs the number of training examples for two noise levels $q = 0.1$ (left) and $q = 0.2$ (right). Each data point was obtained as the average of 100 independent runs.

For both noise levels, the AOG was learned correctly more often when using borrowed parts than when learning the parts directly from the training examples. Also we observe that given sufficient training examples, the dog AOG can be learned with high probability.

Conclusion

In this paper we made the following contributions.

1. We computed the capacity of the space of AOGs of certain maximum depth, breadth and with a given number of terminal nodes and relate the capacity with the number of training examples that guarantee a certain success rate.
2. We showed that the capacity of an AOG is much smaller than the capacity of the k-DNF or k-CNF expressions, so fewer training examples are needed to train a hierarchical representation by parts than a flat representation.
3. We observed that part localization reduces the capacity and the number of training examples.
4. We obtained an algorithm with theoretical guarantees for learning the AOG recursively in a supervised manner.
5. We presented empirical evidence that part noise tolerance and part sharing between object categories (transfer learning) results in a reduction in the number of training examples.

In the future we plan to extend this work to real AOGs that define concepts using score functions obtained by weighted linear aggregation similar to Boosting/SVM and derive learning bounds and algorithms in these more general settings.

Acknowledgment. We acknowledge support from DARPA MSEE grant FA 8650-11-1-7149.

References

Angluin, D. 1988. Queries and concept learning. *Machine learning* 2(4):319–342.

Barbu, A.; Wu, Y. N.; and Zhu, S. C. 2013. Learning Mixtures of Bernoulli Templates by Two-Round EM with Performance Guarantee. *ArXiv*.

Bienenstock, E.; Geman, S.; and Potter, D. 1997. Compositionality, mdl priors, and object recognition. *Advances in Neural Information Processing Systems* 838–844.

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmuth, M. 1987. Occam's razor. *Information Processing Letters* 24(6):380.

Chen, Y.; Zhu, L.; Lin, C.; Yuille, A.; and Zhang, H. 2007. Rapid inference on a novel and/or graph for object detection, segmentation and parsing. *Advances in Neural Information Processing Systems* 20:289–296.

De La Higuera, C., and Oncina, J. 2005. Learning context-free languages.

Fidler, S., and Leonardis, A. 2007. Towards scalable representations of object categories: Learning a hierarchy of parts. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. IEEE.

Jin, Y., and Geman, S. 2006. Context and hierarchy in a probabilistic image model. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, 2145–2152. IEEE.

Kakade, S.; Sridharan, K.; and Tewari, A. 2008. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. *NIPS* 22.

Kearns, M., and Valiant, L. 1994. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM (JACM)* 41(1):67–95.

Kearns, M. J., and Vazirani, U. V. 1994. An introduction to computational learning theory.

Schapire, R.; Freund, Y.; Bartlett, P.; and Lee, W. 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics* 26(5):1651–1686.

Si, Z., and Zhu, S. 2012. Learning hybrid image templates (hit) by information projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34(7):1354–1367.

Tang, K.; Tappen, M.; Sukthankar, R.; and Lampert, C. 2010. Optimizing one-shot recognition with micro-set learning. In *CVPR*, 3027–3034. IEEE.

Valiant, L. 1984. A theory of the learnable. *Communications of the ACM* 27(11):1134–1142.

Wu, Y.; Si, Z.; Gong, H.; and Zhu, S. 2009. Learning active basis model for object detection and recognition. *International journal of computer vision* 1–38.

Zhu, S., and Mumford, D. 2006. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision* 2(4):259–362.

Zhu, L.; Chen, Y.; and Yuille, A. 2009. Learning a hierarchical deformable template for rapid deformable object parsing. *IEEE Trans. PAMI*.