Hierarchical Models of Vision
Cognitive Science and Machine Learning/Computer Vision

Alan Yuille
UCLA
Plan of Lecture

• (I) Beyond Mixture Models – Shared Parts.
• (II) Cognitive Models.
• (III) Compositional Models.

• Problems:

  • (A) Big deformations – hard/impossible to detect the “holistic” object, but some of the body parts (e.g., head) can be detected.
  • (B) Occlusion – detected occluded body parts is difficult or impossible.
  • (C) Low resolution -- it is possible to detect the object, but not its parts (e.g., the head).
  • Our strategy use probabilistic models on graphs, trained by machine learning techniques.
Mixture models and shared-parts.

• Learn models for individual parts and a “holistic” object model.
• Each part is modeled by Deformable Parts Model (DPM). There are several models (type) for each part to deal with different viewpoints.
• An object is represented in terms of:
  (i) the visible parts,
  (ii) the types (mixture component) of the visible parts,
  (iii) the spatial relations between the visible parts.

*Compositional* – we can construct a large number of possible object configurations from a small number of parts and part-types.

*For an input images – the algorithm must rapidly search over all the possible configurations (which parts, which types, which positions).*
Composing Parts/Types and Holistic Model.

Figure 2. (a) Our graphical model where the nodes represent the holistic object and its body parts. Their state variables are position, scale, and switch variable ($\gamma$). The holistic object is shown in yellow and some example body parts are shown in red, green, and blue. The rest of the body parts are shown with white rectangles. (b) The switch variables decouple nodes from the graph, depending on which parts are detected, and enable the model to deal with different detectability patterns. Boxes with dashed border are those decoupled from the graph.
The spatial relations.

- Spatial relations between parts.

Figure 3. The deformation and scale features used in our model.
Examples

• Examples: The red, green and blue boxes correspond to head, torso and legs, respectively. The holistic object boxes are shown in yellow, and the cyan boxes correspond to the full object.
(II) Cognitive Science versus Machine Learning

• Cognitive Science is good because it studies the Human Visual System which is:
  • (i) flexible, adaptive, robust
  • (ii) capable of learning from limited data, transferring,
  • (iii) able to perform multiple tasks,
  • (iv) closely coupled to reasoning, language, and other cognitive abilities.

• Cognitive science seeks fundamental theories capable of explaining phenomena.
Cognitive Science vrs Machine Learning

• Machine Learning/Computer Vision systems are more restricted.

• Currently concentrate on limited sets of tasks – e.g., detect object, not segment it, or detect its parts.

• Often driven by datasets and tasks: does your theory work on Pascal?

• Emphasis on technical “engineering” details. Suspicion of fundamental theories (bad experiences in the 1980’s).
Cognitive Science vrs Machine Learning

- Major Difference: (A) Data and Tasks.
- Computer Vision tests theories on very big datasets – Pascal 20,000 images, ImageNet 1,000,000 images.
- Big datasets required to deal with the complexity of images (no. images, no. objects, scenes).
- Results on restricted datasets – “toy data” – rarely scale up.
(III) Compositional Models: Complexity.

- More 10x10 images -- $256^{100} = 6.7 \times 10^{240}$ -- than the total number of images seen by all humans throughout history $3 \times 10^{21}$.
- (50 billion people, live 20 billion seconds, 30 image per second)
Hierarchical Models

• Why Hierarchies?
• Mimics the structure of the human/macaque visual system.
• Follows the low-, middle-, high-level nature of vision.
• Low-level vision is ambiguous. High-level vision exploits context and is un-ambiguous.
• Optimal design for representing, learning, and retrieving image patterns.
A Probabilistic Model is defined by four elements

• (i) **Graph Structure** – Nodes/Edges -- *Representation*

• (ii) **State Variables** – W – input I. -- *Representation*

• (ii) **Potentials** – Phi -- *Probability*

• (iii) **Parameters/Weights** – Lambda – *Probability*

• *The state variables are defined at the graph nodes.*

• *The potentials and parameters are defined over the graph edges – and relate the model to the image I.*
The Mathematics

- The mathematical formulation.
- Exponential models.

Graph: \((\mathcal{V}, \mathcal{E})\): \(\mathcal{V}\) nodes, \(\mathcal{E}\) edges. \(\mathcal{V}^l\): nodes level \(l\).

Children \(\text{ch}(\mu) \subseteq \mathcal{V}^{l-1}\), siblings \(\text{sib}(\mu) \subseteq \mathcal{V}^l\).

State variables: \(w_\mu, w_{\text{ch}(\mu)}, w_{\text{sib}(\mu)}\) states of children and siblings.

Vertical Potentials \(\phi^V(w_\mu, w_{\text{ch}(\mu)})\): Weights \(\lambda^V_\mu\).

Horizontal Potentials \(\phi^H(w_\mu, w_{\text{sib}(\mu)})\): Weights \(\lambda^H_\mu\).

Data Potentials \(\phi^D(w_\mu, I)\): Weights \(\lambda^D_\mu\).

\[
P(W|I) = \frac{1}{Z[\lambda^D, I]} \times \exp\left\{ \sum_{\mu \in \mathcal{G}} \lambda^D_\mu \cdot \phi^D(w_\mu, I) + \sum_{\mu \in \mathcal{V}} \lambda^V_\mu \cdot \phi^V(w_\mu, w_{\text{ch}(\mu)}) + \sum_{\mu \in \mathcal{V}} \lambda^H_\mu \cdot \phi^H(w_\mu, w_{\text{sib}(\mu)}) \right\}.
\]
Tasks:

• (I) **Inference** – estimate the state $W$ from input $I$ – assuming known Graph Structure, Potentials and Parameters.

• (II) **Learning Parameters/Potentials** – assuming known Graph Structure.

• (III) **Structure Induction** – learn the Graph Structure, Potentials and Parameters.
Grammars/Compositional Models

- Explicit Representations – ability to perform multiple tasks.
Models for Generating Images:

- Grammars (Grenander, Fu, Mjolsness, Biederman).
- Simple to Complex Grammars: Easy to hard Inference
Analysis by Synthesis

- Analyze an image by inverting image formation.
- Proposals and Verification
Can we do this for Real Images?

- Learn probabilistic models of the visual patterns that can appear in images.
- Interpret/understand an image by decomposing it into its constituent parts.
Tasks that can be performed.

- Understanding objects, scenes, and events. Reasoning about functions and roles of objects, goals and intentions of agents, predicting the outcomes of events.
Key Idea: Compositionality

- Objects and Images are constructed by compositions of parts – ANDs and ORs.
- The probability models for are built by combining elementary models by composition.
- Efficient Inference and Learning.
Why compositionality?

(1). Ability to transfer between contexts and generalize or extrapolate (e.g., from Cow to Yak).
(2). Ability to reason about the system, intervene, do diagnostics.
(3). Allows the system to answer many different questions based on the same underlying knowledge structure.
(4). Scale up to multiple objects by part-sharing.

“An embodiment of faith that the world is knowable, that one can tease things apart, comprehend them, and mentally recompose them at will.”

“The world is compositional or God exists”.
Horse Model (ANDs only).

Nodes of the Graph represents parts of the object.

Parts can move and deform.

$y$: (position, scale, orientation)
AND/OR Graphs for Horses

• Introduce OR nodes and switch variables.
• Settings of switch variables alters graph topology – *allows different parts for different viewpoints/poses*:
• Mixtures of models – with shared parts.
AND/OR Graphs for Baseball

- Enables RCMs to deal with objects with multiple poses and viewpoints (~100).
- Inference and Learning by bottom-up and top-down processing:
Results on Baseball Players:

• Performed well on benchmarked datasets.
Unsupervised Structure Learning

• Task: given 10 training images, no labeling, no alignment, highly ambiguous features.
  – Estimate Graph structure (nodes and edges)
  – Estimate the parameters.

Correspondence is unknown

Combinatorial Explosion problem
The Dictionary: From Generic Parts to Object Structures

- Unified representation (RCMs) and learning
- Bridge the gap between the generic features and specific object structures
Bottom-up Learning

Composition Clustering

Suspicious Coincidence

Competitive Exclusion
Dictionary Size, Part Sharing and Computational Complexity

More Sharing
Top-down refinement

• Fill in missing parts
• Examine every node from top to bottom
Part Sharing for multiple objects

Strategy: share parts between different objects and viewpoints.
Learning Shared Parts

- Unsupervised learning algorithm to learn parts shared between different objects.
- Zhu, Chen, Freeman, Torralba, Yuille 2010.
- Structure Induction – learning the graph structures and learning the parameters.
- Supplemented by supervised learning of masks.
Many Objects/Viewpoints

- 120 templates: 5 viewpoints & 26 classes
Learn Hierarchical Dictionary.

- Low-level to Mid-level to High-level.
- Learn by suspicious coincidences.
Part Sharing decreases with Levels
Multi-View Single Class Performance

- Comparable to State of the Art.
Relations to Neuroscience

• Theoretical Models of the Visual Cortex (e.g., Poggio et al) are also hierarchical. Similar to convolutional networks.

• Generative Models may be able to explain top-down and attentional effects.

• Behavior-to-Brain. Tests by Multi-electrodes, fMRI, and behavioral experiments.

• Vision – adapted to the statistics of the environment. Computational theories offer hypotheses for testing.

• Machine Learning also suited to “big data” – prediction and knowledge extraction.
Conclusions

• Cognitive Science and Machine Learning.
• Hierarchical Models: probabilities on graphs.
• Generative Models.
• Relations to Neuroscience,
What types of Probability Models?

• We assume that the models can be expressed in terms of an AND/OR graph.
• But this is an enormous space to search over. We propose a method to search through the space of models.
• Our strategy – compositional – learn small parts first, then proceeds to learn bigger parts.
• Evaluate the models by their probability of generating the data (model selection).
Generative Models and Images

• Learning Generative Models of entire images is too hard at present – cf. special cases.
• Structure Induction is very hard.
• To simplify: use generalize models for simple features.
  • (i) Interest Points (IPs). Described by SIFT.
  • (ii) Edgelets.
• Learn models for objects (not images).
Unsupervised Structure Induction.

• To Summarize:
• We do not know the graph structure.
• We do not know if an object is present in the image.
• We do not know how many types of objects can be present in the image.
• We do not know what IPs are `object’ or `background’.
• We do not know the correspondence between image IP’s and the graphical model.
PGMM 1:

- Dataset – Caltech.

The input data is a set of natural images; 
The output of the model is a structure like following.
PGMM 2.

• The object has a cluttered/noisy background. We do not know what is object and what is background.

The cocktail party effect describes the ability to focus one's listening attention on a single talker among a mixture of conversations and background noises, ignoring other conversations.

• A single talker: Interest Points
• Other conversations: background
PGMM 3.

- This method is based on Interest Points (IPs).
- Why? Because there are few IP’s (sparse).
- They capture important (interesting) parts of the object.
PGMM 4:

• Correspondence problem.
• Some interest-points (IPs) are background
• Others are from the object,
• But from which part of the object?

Why we need inference to solve correspondence problem?

- Data used by Orban is clean and vocabulary known;
- PGMMs extract IP’s and clustering them into a vocabulary from natural images;
- PGMMs needs to extract IP’s and match them to words already known from a new image.
PGMM 5:

• The Basic Idea:

- The basic idea of PGMMs is to search over model structure to find optimal structure.
- The whole procedure is a greedy search,

1. Initially, all of the data are assumed to be generated by a background model, without any spatial relationship between them;
2. Expand the structure by using AND/OR graph grammar, and the grammar will be demonstrated below;
3. For each extension, use the model evaluation method to evaluate it and get a score. Accept the extension with the highest score and update the structure;
4. Repeat 2 & 3 until the score almost doesn't change, exit with graph structure then.
PGMM 6:

- The model is built by a Grammars.
- The basic elements are triplets of IP’s.

**Triplets** is based on the three related nodes’ position $z_i$, scale $l_i$ and orientation $\theta_i$. $r_i = \{z_i, l_i, \theta_i\}$ denote the position feature for a point, and $l = \{r_a, r_b, r_c\}$ denotes a triplet.

**The graph grammars are,**

1. **AND extension.** Combine the new triplet and old one.
2. **OR extension.** Connect the new triplet with an old one.
PGMM 7:

• Grammars – and how to grow them.
• Start with a triplet – and another triplet – if the resulting model fits the data better.
• Model selection – choose between models.
PGMM 8:

- Model section is performed by evaluating the probability that they model generates the data.
- In practice, we make a standard approximation (Laplace).

\[
P(D|I) = \prod_i \sum_H P(D^i|H, \theta^*, I) P(H|\theta^*, I) P(\theta^*|I) P(I)
\]  

Using Laplace approximation.
PGMM 9:

- Some experimental results on Caltech 101:
- Could only use a limited number of model because this approach needs a lot of data.
- Unusual to do unsupervised learning for Caltech.

The classification performance for 26 classes that have at least 80 images. The average classification rate is 87.6 percent.
Summary:

Could learn one, two, three or more models if the dataset required it (e.g. plane, face, bike).

Could learn object models even when half the data was random background.

Performance of models was as good as alternative (supervised methods) for the set of objects with sufficient data (in 2006).
• **Limitations of PGMM**: this model only uses image features defined at interest points.

• How to improve?

• Use this model to learn a ‘skeleton structure” of the object.

• **Then use the skeleton to train a model which uses more cues – edges and appearance.**

• *Eureka Moment?* – *when the simple IP model is powerful enough to train a model with more cues.*

Key Point

• We are trying to learn models for subparts of the models – while ignoring the rest.

• Danger – this makes sense only for a limited class of probability distributions.

• Compositional. Parameters of the models can be learnt independently – provided we know the correspondence. But if we do not, then we can learn them by clustering (contamination of the data).