Unsupervised Structure Learning of Stochastic And-Or Grammars

Kewei Tu, Maria Pavlovskaia and Song-Chun Zhu

Departments of Statistics and Computer Science, University of California, Los Angeles

Stochastic And-Or Grammar

Stochastic And-Or grammars (AOG) are a generalization of stochastic grammars of language that can be used to model other types of data. A stochastic context-free And-Or grammar contains the following elements:

- A set Σ of atomic patterns
- A set N of nonterminal patterns
  - Two disjoint subsets: And-nodes, Or-nodes
- A start symbol $S \in N$
- A set R of probabilistic production rules
  - And-rule (stochastic composition): Composition of an And-node from sub-patterns; a set of relations are specified between the sub-patterns
  - Or-rule (stochastic reconfiguration): An alternative configuration of an Or-node

	Terminal	Nonterminal	Relations in And-rules
Language	Word	Phrase	Deterministic "concatenating" relations
Image	Visual word (e.g., Gabor bases)	Image patch	Spatial relations (e.g., relative positions, rotations and scales)
Event	Atomic action (e.g., standing, drinking)	Event or sub-event	Temporal relations (e.g., co-occurring, following)

Features of stochastic And-Or grammars:

- Compact representation of a large number of patterns via hierarchical compositions and reconfigurations
- Help infer hidden structures from data and solve multiple tasks in a unified way

Unsupervised Learning

Learning a stochastic grammar involves two parts:

- Learning the grammar rules (structure)
- Learning the rule probabilities (parameters)

Unsupervised learning:

- Learning from unannotated i.i.d. data samples (e.g., natural language sentences, quantized images, action sequences)

Our objective function is the posterior probability of the grammar:

$$P(G|X) \propto P(G)P(X|G) = \frac{1}{Z}e^{-\alpha\|G\|}\prod_{x_i \in X} P(x_i|G)$$

Grammar — Unannotated training data — Prior that penalizes the grammar size — Likelihood (to be approximated by Viterbi likelihood)

Algorithm

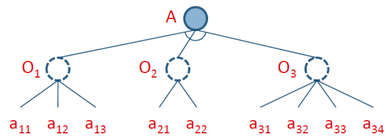
1. Start with the maximum-likelihood grammar (simply the union of all the training samples)
2. Repeat:
   Add a new grammar fragment and use it to reduce training samples s.t. the posterior is maximally increased
   Until no more fragment can be learned

A grammar fragment is a set of grammar rules that are rooted at a new nonterminal node and specify how the new nonterminal node generates one or more configurations of existing nodes.

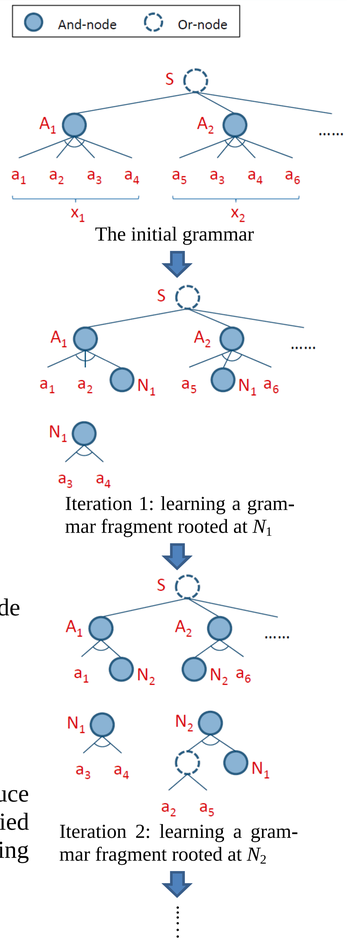
And-Or Fragment

We propose to search for And-Or fragments in the algorithm. An And-Or fragment contains:

- A new And-node as the root
- A set of new Or-nodes under the And-node
- A set of existing nodes under each new Or-node



By learning with And-Or fragments, we induce compositions and reconfigurations in a unified manner that is more efficient and robust than learning with other types of grammar fragments.



The initial grammar

Iteration 1: learning a grammar fragment rooted at N_1

Iteration 2: learning a grammar fragment rooted at N_2

Posterior gain computation

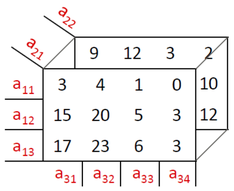
The posterior gain of adding an And-Or fragment can be efficiently computed based on a set of sufficient statistics.

Likelihood gain:

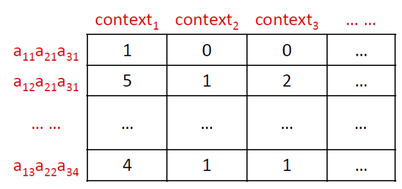
The product of the coherence of the n-gram tensor and the coherence of the context matrix.

Prior gain:

Determined by the change of the grammar size: increased by the size of the And-Or fragment; decreased by the reduction (the sum of elements of the n-gram tensor)



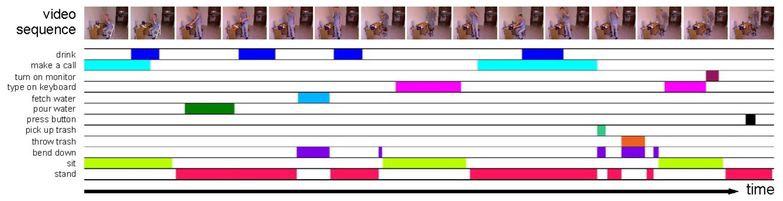
n-gram tensor



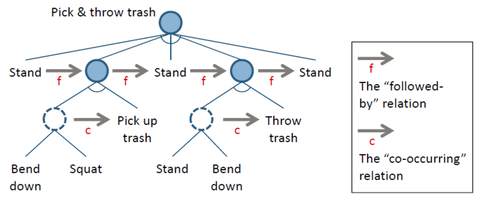
Context matrix

Experiments on Event Data

We applied our approach to learn event grammars from human activity data. Each input video was preprocessed into a sequence of binary vectors; each element in the vector represents an atomic action. The learned event grammar is evaluated by comparing the events identified by the grammar against the manual annotations on the test data.



An example input from the human activity dataset. Each colored bar denotes the start/end time of an occurrence of an action.



Part of the learned event And-Or grammar

	Data 1	Data 2
ADIOS [15]	0.810	0.204
SPYZ [18]	0.756	0.582
Ours (f)	**0.831**	0.702
Ours (c+f)	0.768	0.624
Ours (cf)	0.767	**0.813**

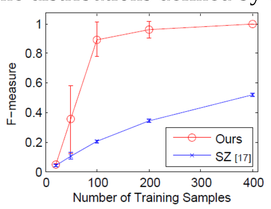
Evaluation results (F-measure) of our approach and two previous approaches

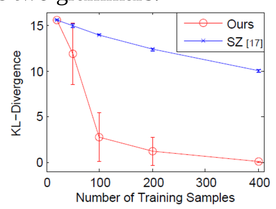
Experiments on Image Data

We tested our approach in learning image grammars from images.

On a synthetic dataset of animal face sketches, we compared the learned grammar against the true grammar by measuring the precision and recall of the sets of images generated from the two grammars, as well as the KL-divergence between the distributions defined by the two grammars.

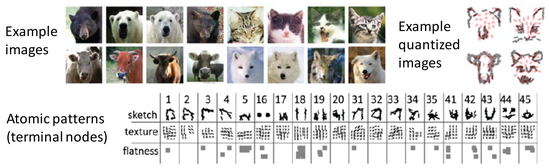
Example synthetic images





Evaluation results of our approach and a previous approach.

On a real dataset of animal faces, we first quantized the images with a set of learned atomic patterns; we then applied our approach and evaluated the perplexity of the learned image grammar.



Example images and the atomic patterns

	Perplexity
Ours	67.5
SZ [17]	129.4

Evaluation results of our approach and a previous approach (lower is better).