Pattern theory for representation and inference of semantic structures in videos

Fillipe D.M. de Souza\textsuperscript{a,}\textsuperscript{*}, Sudeep Sarkar\textsuperscript{a}, Anuj Srivastava\textsuperscript{b}, Jingyong Su\textsuperscript{c}

\textsuperscript{a} Computer Science & Engineering, University of South Florida, 4202 E. Fowler Avenue, ENB 118, Tampa, FL 33620, USA
\textsuperscript{b} Department of Statistics, Florida State University, 117 N. Woodward Ave., Tallahassee, FL 32306, USA
\textsuperscript{c} Department of Mathematics & Statistics, Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA

\textbf{A R T I C L E   I N F O}

Article history:
Available online 26 February 2016

Keywords:
Pattern theory
Graphical methods
Compositional approach
Video interpretation
Activity recognition

\textbf{A B S T R A C T}

We develop a combinatorial approach to represent and infer semantic interpretations of video contents using tools from Grenander’s pattern theory. Semantic structures for video interpretation are formed using generators and bonds, the fundamental units of representation in pattern theory. Generators represent features and ontological items, such as actions and objects, whereas bonds are threads used to connect generators while respecting appropriate constraints. The resulting configurations of partially-connected generators are termed scene interpretations. Our goal is to parse a given video data set into high-probability configurations. The probabilistic models are imposed using energies that have contributions from both data (classification scores) and prior information (ontological constraints, co-occurrence frequencies, etc.). The search for optimal configurations is based on an MCMC, simulated-annealing algorithm that uses simple moves to propose configuration changes and to accept/reject them according to the posterior energy. In contrast to current graphical methods, this framework does not preselect a neighborhood structure but tries to infer it from the data. The proposed framework is able to obtain 20% higher classification rates, compared to a purely machine learning-based baseline, despite artificial insertion of low-level processing errors. In an uncontrolled scenario, video interpretation performance rates are found to be double that of the baseline.

\textcopyright 2016 Elsevier B.V. All rights reserved.

1. Introduction

An automated system for providing semantic interpretations of videos will be greatly useful in many vision-based applications, including cyber-security and video content-based search applications. However, the current solutions in the literature (see [9] for a comprehensive survey) lack domain generalization and the ability to address multiple challenges at the same time. The difficulties in developing a fully-automated systems are multiple. Some relate to low-level detection and categorization, while others relate to high-level processing and interpretation. The challenges originate from: (1) handling errors stemming from the low-level processing layer, (2) lack of a flexible representation scheme to accurately model semantic structures of interest, and their variability, and (3) need for a principled mechanism to integrate multiple data sources and efficient algorithms to estimate interpretations in new data.

Table 1 summarizes the most recent approaches on the problem of video understanding. On one side, the simplest approaches concentrate on choosing the right combination of features and machine learning algorithms to infer the semantic contents as elements belonging to a pre-defined collection of semantic labels. In these approaches, feature detectors are assumed to be noise-free and learned classification models to be sufficiently discriminative with the chosen types of features. They rely on the robustness of learning algorithms to handle low-level feature errors. Additionally, no useful information about the structures of interest are explicitly encoded in the models – it is assumed that the chosen feature representation implicitly captures useful structural information descriptive of the target collection of semantic labels. Note that these classification models do not account well for structural variabilities inherent to real semantic structures. On the other hand, explicit models (or structured models) partially solves the later issues by providing complex structured models such as Bayesian networks to represent the semantic structures of interest and small variabilities. These proposed fixed structures are not...
Table 1
Summary of most recent related works on video understanding.

<table>
<thead>
<tr>
<th>Work</th>
<th>Features</th>
<th>Representation</th>
<th>Learning</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit models</td>
<td>Labeling</td>
<td>STIP, HOG</td>
<td>BoVW</td>
<td>DPM, bagged REP decision,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bayes’ rules</td>
</tr>
<tr>
<td></td>
<td>[15]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OpponentSIFT, STIP</td>
<td></td>
<td>BoVW-SVM</td>
<td>SVM</td>
</tr>
<tr>
<td></td>
<td>[7]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text description</td>
<td>MPEG-Flow</td>
<td>Haar, color, EOYH,</td>
<td>BoVW-SVM</td>
<td>Probabilistic parser</td>
</tr>
<tr>
<td></td>
<td></td>
<td>COH</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[10]</td>
<td></td>
<td>High-level features</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SIFT, STIP, MFCC</td>
<td></td>
<td>BoVW-SVM, event-concept</td>
<td>Linear combination of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>relevancy matrix</td>
<td>classification scores with</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>[21]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HOG, STIP</td>
<td></td>
<td>DPM, SVM</td>
<td>Linear interpolation of</td>
</tr>
<tr>
<td></td>
<td>[4]</td>
<td></td>
<td>MMLDA, DPM</td>
<td>scores R</td>
</tr>
<tr>
<td>Explicit models</td>
<td>Labeling</td>
<td>Not mentioned</td>
<td>ADBN</td>
<td>Approx. viterbi</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STIP-BoVW</td>
<td>Variable duration</td>
<td>MAP w/dynamic programming</td>
</tr>
<tr>
<td></td>
<td>[14]</td>
<td></td>
<td>HMM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[12]</td>
<td></td>
<td>SPN</td>
<td>MPE w/graph parsing</td>
</tr>
<tr>
<td>Structured output</td>
<td>Not mentioned</td>
<td>STIP-BoVW</td>
<td>Hierarchical PCM</td>
<td>Coordinate ascent + loopy BF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GC-DBN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[13]</td>
<td></td>
<td>Not mentioned</td>
<td>Adaboost + EM</td>
</tr>
<tr>
<td></td>
<td>Not mentioned</td>
<td></td>
<td>SRG (CFG)</td>
<td>Maximum likelihood</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Not mentioned</td>
<td>FSM (Viterbi)</td>
</tr>
<tr>
<td></td>
<td>[16]</td>
<td></td>
<td>STG+BN</td>
<td>Message passing</td>
</tr>
<tr>
<td></td>
<td>[24]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[8]</td>
<td></td>
<td>SCFG</td>
<td>Graph parsing w/HTK</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Manual</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not mentioned</td>
<td>STIP, trajectories-</td>
<td>AOG</td>
<td>Earley’s-like parser</td>
</tr>
<tr>
<td></td>
<td>[17]</td>
<td>BoVW (actions as HMMs)</td>
<td></td>
<td>hierarchical cluster sampling + Earley’s parser</td>
</tr>
<tr>
<td></td>
<td>Not mentioned</td>
<td></td>
<td>AOG</td>
<td>DN Beam Search</td>
</tr>
<tr>
<td></td>
<td>RGYD HOG, Kinects 3D</td>
<td></td>
<td>4DHOI (Hierarchical</td>
<td></td>
</tr>
<tr>
<td>joint motion vectors</td>
<td></td>
<td></td>
<td>graph)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[25]</td>
<td>Pattern theory</td>
<td>Concept co-occurrence</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HOG, HOF</td>
<td></td>
<td>tables</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[19]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\* Information projection

\* Minimum description length principle

obviously scalable for larger variations of target semantic structures and may require an enormous amount of training data to estimate good parameter values that characterize those variations well.

In this paper, we present a novel way of approaching the problem of video semantic interpretation. Here, the elements of Grenander’s pattern theory [5] are used to model structures of semantic interpretations. This framework has been used in the past for applications involving computational anatomy, parsing language structures, and modeling biological growth [6]. In pattern theory (PT), generators are the most fundamental units of representation. In the context of video interpretation, they represent items pertaining to some domain-specific knowledge ontology, which are called ontological generators, and features extracted from videos, known as feature generators. Generators have bond structures that allow them to combine with each other, while satisfying ontological rules and forming connected structures that represent higher-level inferences. Thus, in our application, generators represent observable actions, imaged objects and video features. They can also potentially represent more complex concepts. The connecting bonds are termed in-bonds or out-bonds, determining the types of interactions between generators. Comparing to the terminology used in graphs, generators are nodes and bonds are edges. However, the generator structure is richer as it also allows for explicit representation of unconnected or dangling bonds, unlike in graphs where there is no concept of a dangling edge. The PT approach provides us with a framework for flexible representation of video semantic structures and a principled mechanism to infer these structures in new data. Such framework incorporates both prior knowledge and data in the form of machine learning-based classification models.

The contribution of this framework in advancing the state-of-the-art tools for video understanding is four-fold. First, it provides a principled and mathematically-grounded mechanism to model and infer semantic interpretations for video content descriptions. Second, it provides a very flexible and comprehensive structural representation scheme for describing complex semantic structures. Third, it is capable of overcoming errors generated by low-level classifiers, with the help of ontological constraints encoded in the representation; we shall demonstrate this capability using experiments. Fourth, the space of feasible interpretations does not grow exponentially in the number of features/concepts in our representation, yet it is rich and comprehensive. Consequently, inferences in our representation space are generated using polynomial time algorithms, built on a combination of MCMC (Markov Chain Monte Carlo) sampling and simulated annealing.

A short version of this work was presented in [19]; thus, this paper is an extended account presenting a more elaborate description of the theory, about the mapping of its conceptual elements to the practical scenario of modeling video semantic interpretations (so that it easily accessible for implementation) as well as more experimental studies for different scenarios of degradation (in terms of error rates from the low-level processing layer). We also published an extension of this work in [18] that studies how to handle inference for long sequences of videos by considering temporal information.
2. Related work

Most works treat video interpretation as a pattern classification problem. Their typical data representation scheme is based on bag-of-visual-words and the classification models are learned using traditional machine learning algorithms such as Support Vector Machines (SVM) [7, 10, 15]. These approaches attempt to bridge the gap between feature representation and video classification in one step. Variations of these exist [12, 21], in which more steps and data sources are added to the recognition framework, but all encode context, logical and temporal dependencies implicitly. In addition, outputs form different steps are combined with each other to generate natural language description of videos [4, 11, 12, 21]. We call all of them implicit models. In summary, for these models, the video interpretation task may consist of either video labeling or video description by natural language sentences. These approaches are easy to implement but they suffer limitations in handling low-level processing errors, structure representation and scalability.

Another direction of works on video interpretation offers solutions that overcome some of these limitations only partially. These solutions encode context, logical and temporal dependencies explicitly using graphical models and semantic knowledge. Such models are in general parametric, such as Hidden Markov Models (HMMs) and dynamic Bayesian Networks (DBNs), that require full estimation of many parameters or probability distributions in high dimensional spaces. We classify these approaches as explicit models because they model the structure of events explicitly. With explicit models, video interpretation is commonly addressed as video labeling by maximizing some posterior probability or equivalently minimizing an energy function for input video features, given example graphical models of interest, e.g., [8, 16, 20, 22, 24]. Structured interpretations of videos are also possible using graphical models; examples of these are methods based on And-Or graphs (AOG) and context-free grammars (CFG) [17, 23, 25].

Table 1 pinpoints the variations among works in both major categories of video interpretation approaches, i.e. explicit and implicit models. The difference is in the abstraction model used to represent the semantic structure of events and in the chosen learning and inference methods. Histograms of optical flow and gradient orientations are among the most commonly used features, typically extracted from local neighborhood of space-time interest points. Bag of Visual Words (BoVW) appear to be a widely common statistical representations based on these features to summarize structural information. These works also differ in other aspects not shown in Table 1, for instance, application domain and context-dependent assumptions.

The major limitations current state of the art are low flexibility in representing the structural variability of events, in the scalability of the structure representation models, in handling clutter of information and in the non-uniformity in which domain knowledge constraints are encoded.

3. Pattern theory based computational structure

The proposed pattern-theory framework uses both prior knowledge about the target domain and observed data (features extracted from videos) to construct semantic interpretations of an observed scene. An example of a semantic interpretation of the video in Fig. 1 (left) is stirring carrots in a bowl using spatula. Note that the goal is not to label or classify the video into a particular category but, more generally, to generate a description about the occurring atomic actions and objects. This involves identifying key semantic elements from the video data, such as actions and objects and their relationships, to form interpretations of ongoing events in the video.

We express semantic video interpretations as connected structures that are built using elements of pattern theory. Specifically, items such as actions, objects and features appear as generators (graph nodes) and their interactions as closed bonds (graph edges); see Fig. 1 (right) for an illustration. Observed features provide direct evidence of the occurrence of actions and objects in a scene. Priors in the form of co-occurrence frequency tables help handle uncertainty/noise in observed features. Priors will favor interpretations that are semantically more coherent by constraining certain items to occur together more often than others, for instance, spatula is likely to be in the scene if the observed action is stir. The representation also allows for incomplete bonds to capture cases such as missing feature evidence for predicted actions or objects. In this respect, it differs radically from traditional graph structures where dangling edges are not allowed.

Low-level features can be extracted from spatiotemporal regions of interest, using a pre-processing stage such as segmentation, object detection and motion salience extraction. An example of a set of bounding boxes corresponding to such regions is shown in Fig. 1 (center). It shows a video depicted on the left going through the pre-processing phase where regions of interest containing information about the events and objects in the scene are detected and described using histograms of optical flow (HOF) and histograms of oriented gradients (HOG). In the pattern theory framework, these features form feature generators that connect and support the action and object generators in an interpretation structure. The complete collection of generators formed by action generators (e.g., stir, pick up, pour) and object generators (e.g., bowl, knife, spoon) are derived from the target ontological domain; thus, they are broadly referred to as ontological generators.

As mentioned earlier, generators form nodes in the pattern-theoretic interpretative structures, as the one shown in Fig. 1 (right). A zoom-in of Fig. 1 (right) is shown in Fig. 2 to emphasize the point that generators are connected by compatible bonds. For example, in Fig. 2, the stir generator connects to the bowl generator by their bonds carrying bond value receptacle. Connections between ontological generators such as stir, spatula and bowl, carrot are valid given prior knowledge about the target domain. Bond interactions formed by a feature generator and an ontological generator are based classification scores associated with corresponding feature observations. In short, a collection of connected action and object generators forms a configuration that is interpreted as a semantic description of a target video scene.

Thus far, we have only informally introduced the elements of representation in pattern theory, namely, generators, bonds and configurations. As follows, we present their formal definitions accompanied with concrete examples. Illustrations are provided to support better understanding of the concepts.

3.1. Generators, bond structures and modality sets

Definition 1. Generators are structured units of representation, formally denoted by \( g \), used as parts to build complex patterns.

In the cuisine domain, generators are actions and objects such as stir, pour, pickup, putdown, bowl, spoon, plate, tomato, lettuce, etc. We refer to them as stir generator, pour generator, etc. The complete collection of generators forms the generator space \( G \). Each generator \( g \) has a bond structure \( B(g) \) that allows them to combine with other generators, as determined by the domain ontology. For example, in Fig. 2, the stir generator has a bond structure formed by the arrangement of bond values (receptacle, stirrer, food, feature).
The out-bond (non-shaded half-circles) carrying bond value receptacle in the stir generator can connect to any other generator having a compatible in-bond (black-shaded half-circles). In Fig. 2, this meant to connect it to the in-bond of bond value receptacle of the bowl generator.

The bond structure $B(g_i)$ of a generator $g_i$ can be represented by a tuple $(B_i(g_i), B_o(g_i))$. $B_i(g_i)$ is the structural arrangement of in-bonds and out-bonds, each identified by a coordinate $j = 1, 2, \ldots, w(g_i)$, where $w(g_i)$ is the cardinality of bonds in $g_i$. $B_o(g_i)$ is the collection of bond values $\{b_i(g_i), \ldots, b_{w(g_i)}(g_i)\}$ associated with each bond $j$ in $B_i(g_i)$. In-bonds indicate incoming connections and out-bonds are outgoing connections; thus, out-bonds connect to in-bonds, which results in bond interactions.

Bond structures account for ontological constraints that conform with semantic dependencies of the target domain. They ensure that the constructed interpretations have coherent pattern structures. There is no restriction on what the set of bond value $\beta$ can be and how it should be defined. The only rule is that they are used to construct the local structure of generators such that, when generators are connected to each other through their bonds, the constructed structure respects the rules of combination. We let the set of bond values be composed of the names of the modalities (concept that will defined next). Later in this section, the rules of combination will be formalized in terms of a bond relation function $\rho$.

**Definition 2.** A modality set $\mathcal{M}$ is a partition of $G$ into disjoint sets $G_k \subset G$ such that generators belonging to the same modality $M_k \in \mathcal{M}$ have similar properties.

**Remark 1.** A modality $M_k \in \mathcal{M}$ is induced by a similarity $s \in S$.

Generators $g \in G_k$ are similar to each other with respect to some property. For instance, for the modality receptacle, $G_k = \{\text{bowl, cup, pan}\}$ and the common property is that all these objects can be used as a liquid or food container. For simplicity and convenience in design, we use the names of the modalities in $\mathcal{M}$. For example, receptacle, stirrer, utensil, food, feature as the set of possible bond values; however, in general, the set of modalities $\mathcal{M}$ is independent of the set of bond values $\mathcal{B}$.

The concept of modality adds flexibility to the construction of interpretative structures by allowing generators of same modality to be exchangeable and take on different roles in one interpretation. Generators combine with each other through their bond structures to form configurations that represent complex patterns of interest. An example of configuration is shown in Fig. 2 for which the associated semantic interpretation is **stirring carrots in a bowl using spatula**.
a bowl using spatula. Hereafter, the terms configuration and interpretation will be used interchangeably.

The modality set will be initially defined by the domain knowledge ontology. From the point of view of semantics, categories that uniquely qualify groups of objects and actions can be used as a start point. For example, in the cuisine domain, the set of kitchen utensils that are used to perform the action stir may be represented by a modality stirrer; the generators for this modality are (spoon, spatula, whisk). Nonetheless, the modality set can evolve to accommodate changes to the formulation of the problem.

3.2. Interpretations

Generators interact with other generators through bonds to form structures referred to as configurations. In this paper, configurations will be interpreted as semantic interpretations (or descriptions) of video scenes.

**Definition 3.** An interpretation $c = \sigma(g_1, g_2, \ldots, g_n)$ is a collection of generators interrelated by a connector graph $\sigma$. The generators are the content of the interpretation as a list of elements, i.e. actions and objects. The connector graph establishes the interactions of these elements expressing the ongoing events in a scene.

Connections between generators form regular closed bonds when they are compatible. Such compatibility is defined by the bond relation function $\rho$. The bond relation $\rho$ is a truth-valued function that determines whether two bonds $\beta'$ and $\beta''$ are or are not compatible. These relations are derived from some domain-specific knowledge, for example, by mining annotations of the dataset instances or can be manually specified.

A complete illustration of an interpretation is shown in Fig. 2. The semantic interpretation for the input video at the bottom is “stirring carrots in a bowl using spatula”. This description is expressed in terms of a graph structure constructed using generators and bonds. There are ontological generators that account for action and object labels (e.g., stir, carrots, bowl) as well as feature generators (e.g., HOG and HOF) that account for different types of features. Bond interactions between feature and ontological generators provide data evidence for the corresponding concepts. The bond interactions formed between ontological generators help establish supportive evidence from priors about the co-occurrence of concepts; thus, strengthening semantic explanations with concepts for which data evidence is weak for the uncertainty of the classification models.

3.3. Probability of an Interpretation

The probability of an interpretation $c = \sigma(g_1, \ldots, g_n)$ is expressed as a product of terms associated with generators and bond interactions, written as

$$p(c) = \frac{k_n}{n!Z} \prod_{i=1}^{n} Q(g_i) \prod_{g_i \in \lambda \in c} A(g_i, g_j),$$

where $Q(g_i)$ captures the prior likelihood of a generator in a domain. Note that it is not the marginal probability of a generator.

In our experiments, we fixed all $Q(g_i)$ to one single constant value, but it can be used to influence the choice to favor a subset of generators, if needed.

We capture the degree of semantic acceptance between two generators, $g_i$ and $g_j$, made through a particular pair of bonds, $\beta'_i$ of $g_i$ and $\beta'_j$ of $g_j$, by $A(\beta'_i(g_i), \beta'_j(g_j))$, denoted here by $A(g_i, g_j)$ for simplicity. More details on this term are given in the next section.

The denominator term $Z$ is the normalization constant given as a sum over configurations formed with all possible connector graphs $\sigma \in \Sigma$

$$Z = \sum_{\sigma \in \Sigma} \sum_{c \in C(\sigma)} \frac{k_n}{n!} \prod_{i=1}^{n} Q(g_i) \prod_{g_i \in \lambda \in c} A(g_i, g_j).$$

The inner sum is over all possible configurations with a fixed underlying connector graph structure, $C(\sigma)$. The outer sum is over all possible underlying connector graph structures. This outer sum differentiates it from a traditional Markov random field (MRF) formulation. In an MRF, we have a fixed connector graph structure and consequently the outer sum does not exist. As a result, we cannot use well known, efficient MRF optimization methods, such as graph cuts or message passing, for inference in our case.

The ratio $\frac{k_n}{n!}$ captures the complexity of the interpretation. This complexity term is chosen to follow a Poisson distribution with mean at $\xi$, where $k_n = O(\xi^n)$. For practical purpose, $k_n = \frac{\xi^n}{n!}$, where $\xi$ is a constant dependent on the maximum number of closed bonds allowed in a configuration. This multiplicative ratio factor helps the probability function to have a desirable property – probability of an interpretation that is the simple union of two smaller interpretations is lower than the product of the probabilities of the individual interpretations. There is an built-in bias against simple addition of two configurations, with no bonds linking them. We can also show that the partition function is bounded if $\xi$ is set to less than a value that is dependent on the maximum number of links.

3.4. Bond interactions

We capture the degree of acceptance between two generators, $g_i$ and $g_j$, made through a particular pair of bonds, $\beta'_i$ of $g_i$ and $\beta'_j$ of $g_j$, by the affinity of the bond interaction $A(g_i, g_j)$. We specify the logarithm of this bond affinity as the bond energy. The affinity function $A(g_i, g_j)$ will take different forms, depending on the nature of the bond interaction. There are essentially two kinds of bond interactions, namely, ontological bonds and support bonds.

Bonds between ontological generators, i.e. ontological bonds, encode contextual relevant information that help form semantically coherent interpretations. Affinities of these bonds are calculated as a function of co-occurrence frequency values of actions and objects. These values indicate the frequency in which actions and objects occur together, that is, the fraction of times that a particular pair of ontological labels (i.e. action–object, object–object, and action–action category pairs) exists in the training data set annotation. This frequency table, hereafter represented by a $f(x, y)$, is computed according to

$$f(g_i, g_j) = \sum_{k=1}^{V} \delta[g_i \downarrow g_j \in V_k], \text{ s.t. } i \neq j,$$

where $\delta[g_i \downarrow g_j \in V_k]$ outputs 1 if $g_i$ and $g_j$ co-occur in the video clip $k$ (or 0 otherwise), $V$ represents the set of annotations of the training data set and $V_k$ is the list of concepts that appear in the video clip $k$. The ontological bond affinity is given by

$$A(g_i, g_j) = \exp(\tanh(w_1 f(g_i, g_j))).$$

where $w_1 = 0.025$, an empirically defined choice and depends on the range of $f(g_i, g_j)$. We sought to model the bond affinity as a monotonic, bounded function; therefore, the hyperbolic function was a possible choice. Bounding each bond affinity prevents one bond from dominating the total energy function.
Support bonds serve to ground the action and object generators upon feature evidence. They connect action and object generators to feature generators. The support bond affinities are computed as a function of classification scores provided by machine learning-based classifiers of actions and objects using the linked features. For example, given a linear-SVM classification model based on HOG features for recognizing the category bowl, the bond interaction $g_i = \text{bowl} \downarrow g_j = \text{HOC}$ will be quantified by the classification confidence score of “bowl" based on the connected HOG features. We quantify the support bond affinity by

$$A(g_i, g_j) = \exp(\tanh(w_2 h(g_i, g_j)))$$

(5)

where $w_2 = 2$ and $h(\cdot, \cdot)$ is the classification confidence score output by the classification model for the concept label represented by generator $g_i$. Scores are real numbers in the range 0 to 1, computed by the classifiers, with absolutely confident classifier being 1.

3.5. Inference

Finding the best interpretation for a video requires searching over a very large combinatorial space of interpretations. With the restrictions imposed by the bond relation function $\rho$, the space of interpretations is reduced to a manageable size for which there is a polynomial-time algorithm that we describe in this section. The goal of this algorithm is to find the highest probability interpretation in such space of feasible interpretations. We accomplish this by minimizing the equivalent energy form of $p(c)$. Maximizing $p(c)$ in Eq. (1) is the same as minimizing the corresponding energy $E(.)$, given by $-\log Z_p(.)$.

$$E(c) = \log n! - n \log \xi - \sum_{i=1}^{n} \log Q(g_i) - \sum_{g_i \in c} \log A(g_i, g_j)$$

(6)

The inference engine is kept general to allow for simultaneous variable and model estimation; a particular tool in this setup is the random sampling of posterior. We can synthesize random possible interpretations by an iterative procedure of simple and/or composite moves through the interpretation space. We employ an MCMC based simulated annealing process.

Algorithm 1 outlines the structure of the proposed MCMC process. It receives as input the generator space $G$, a configuration $c'$ in which each feature generator is connected to the $K$ best actions or objects that can explain it (using support bonds only), the parameter $\alpha$ that regulates the rate of change of the annealing temperature at each iteration $k$, a probability $p$ determining how probable it is to sample interpretations using the global proposal function, and the maximum number of iterations $k_{\text{max}}$. The algorithm starts in line 2 by initializing the process with a configuration $c$ sampled from the global proposal function. In line 5, a random number $z$ between 0 and 1 is sampled according to a uniform distribution. In lines 6 and 7, a new configuration $c'$ is sampled from the global proposal function with probability $p$ or from the local proposal function with probability $1-p$. The annealing temperature $T$ is updated in line 8; the closer $\alpha$ is to 1, e.g., 0.999999, the lower the drop of the temperature $T$ at each iteration $k$. This value can be adjusted to regulate the percentage of time spent on the exploratory phase. In lines 9 and 10, it is decided whether to accept or to reject the newly proposed interpretation $c'$ given the current interpretation $c$. This process is repeated $k_{\text{max}}$ times and the best interpretation found during this process is kept in $b$.

The global proposal function makes significant changes in the interpretation structure and the local proposal function proposes local modifications to the structure. They are specified in Algorithm 2 and Algorithm 3, respectively.

Algorithm 1:

**Input:**

- $G$ - generator space, i.e. the list of actions and objects
- $c'$ - a configuration in which each feature generator is connected to the $K$ best actions or objects that can explain it
- $\alpha$ - a constant used to regulate the rate of change of the annealing temperature at each iteration $t$
- $t$ - probability value for choosing the global proposal at each iteration
- $k_{\text{max}}$ - maximum number of MCMC iterations

**Output:**

- $b$ - best interpretation found in the iterative procedure

1: procedure MCMC Simulated Annealing($G$, $c'$, $\alpha$, $t$, $k_{\text{max}}$)
2: $c \leftarrow \text{GLOBALPROPOSAL}(c')$
3: $b \leftarrow c$
4: for $k \leftarrow 1, k_{\text{max}}$ do
5:  $z \sim U(0, 1)$
6:  if $z < \alpha$ then $c' \leftarrow \text{GLOBALPROPOSAL}(c')$
7:  else $c' \leftarrow \text{LOCALPROPOSAL}(G,c)$
8:  $T \leftarrow T_0 \times \alpha^k$
9:  if $P(E(c'), E(c), T) > \text{True}$ then
10:    $c \leftarrow c'$
11:  if $E(c') < E(b)$ then
12:    $b \leftarrow c'$
13: return $b$
14: procedure $p'(c', c, T)$
15: if $c' < c$ then return $\text{True}$
16: else $z \sim U(0, 1)$
17: if $z < \exp(-E(c') - E(c))/T$ then return $\text{True}$
18: else return $\text{False}$

The first step in the global proposal function consists of sampling one action or object generator to explain each feature generator (see lines 2–7 in Algorithm 2). The selected generator represents one of $k$ categories of actions or objects given their classification score on the target features. The result of this is an interpretation composed by bond interactions between action/object generators and feature generators. The next step lies in placing connections between action and object generators while obeying the bond relation $\rho$. Note that the final constructed interpretation is an arbitrary sample from the space of interpretations.

In the local proposal function (Algorithm 3), first an action or object generator $g_0$ from the current configuration $c$ is chosen for removal. Then, three candidate generators semantically closer to $g_0$ (i.e. belonging to the same $G_k$) are selected as possible replacements. Finally, it is verified which replacement generator produce a configuration with lowest total energy when compared to the other candidates. This is the generator selected for replacement, contributing for the formation of an interpretation with new ontological bonds and a different semantic interpretation.

The MCMC-based inference algorithm calls either of these proposal functions at each iteration with certain probability; thus, a new interpretation is sampled at each iteration. The new interpretation is accepted or rejected depending on its total energy compared to the current's. The simulated annealing process is run on a linearly decreasing temperature schedule.

3.5.1. Time complexity

The time complexity of our MCMC-based simulated annealing algorithm is upper bounded by $O(km_0 + k(nT + n^2))$, where $k$ is the total number of sampling iterations. The first term refers to the computational cost incurred by the local proposal function and the second by the global proposal, where $m_0$ is the number bonds from a candidate generator for replacement, $m_0$ is the total number...
of open bonds in a current interpretation and $n_f$ is the number of feature generators.

We recorded and computed statistics of the execution time on a machine with 4 16-core 2.3 GHz CPUs (AMD Opteron 6376) and 16 16GB RAM units. The running times ranged from 46 s for a configuration with 5 feature generators and 5 ontological generators to 126 s for a configuration with 12 feature generators and 12 ontological generators.

Algorithm 2.

**Input:**
- $c$ - a configuration in which each feature generator is connected to the $K$ best actions or objects that can explain it

**Output:**
- $c'$ - newly proposed/assembled configuration

1: procedure GLOBAL PROPOSAL($c$)
2: $c' \leftarrow$ create_empty_configuration()
3: for each $g'_f$ in $c$ do
4:     $g'_o \leftarrow$ get_ontological_generator($g'_f$, U(0, 1))
5:     $c'.add_new_generator(g'_o)
6:     $c'.add_new_generator(g'_s)$
7: end for
8: connect_bonds($g'_o$, $g'_s$)
9: for each $g'_s$ in $c'$ do
10:     if $i \neq j$ then try_connect_bonds($g'_f$, $g'_s$)
11: return $c'$.

Algorithm 3.

**Input:**
- $G$ - generator space, i.e. the list of actions and objects
- $c$ - the current configuration

**Output:**
- $c'$ - newly proposed configuration by local

1: procedure LOCAL PROPOSAL($G$, $c$, $k = 3$)
2: $g'_o \leftarrow$ remove_any_generator($c$)
3: $C \leftarrow$ random_candidates($G$, $g'_o$, $k$)
4: for each candidate $g'_o$ in $C$ do
5:     contrib_energy($g'_o') \leftarrow 0
6:     for each open ontological bond $\beta_i \in g'_o$ do
7:         for each ontological bond $\beta_j$ in $c$ do
8:             if $\rho(\beta_i, \beta_j) == True$ then
9:                 contrib_energy($g'_o$) \leftarrow energy($\beta_i, \beta_j$)
10:     end if
11: end for
12: end for
13: $g'_o$ \leftarrow$ $with lower non-zero contributed energy
14: Update $c$ with the new replacement $g'_o$
15: return $c'$.

4. Experimental setup

In this section, we describe the benchmark data set chosen to study the pattern theoretic framework performance and describe the baseline algorithm implemented for comparative analysis. This section ends with a description of the metrics used.

4.1 Data

The proposed framework was evaluated on videos depicting cooking scenarios. The videos were taken from a challenging, recently published data set [4], called the YouCook data set, consisting of instructional videos of different cooking styles, such as assembling, baking, grilling, etc. This collection of videos is challenging with diverse scenarios containing cluttered backgrounds, different subjects, occlusion of objects, shot from different viewpoints, and different illumination conditions.

Only videos whose annotations were available were considered for purpose of performance evaluation. Such annotations contain information about the spatiotemporal location of objects and actions occurrences in the video scenes. There were 44 videos for which these annotations were provided. They were organized into two groups of equal size: a training and a test set (i.e. each containing 22 videos). The annotation from the training videos are used to extract prior knowledge about the interactions described by actions performed on objects. In particular, we computed concept co-occurrence tables as described in Section 3.4 (see Eq. 3).

The interactions involving actions performed on objects are varied and numerous in this data set. They are composed by the combination of actions and objects. In this study, 6 categories of actions and 18 of objects were considered. The action categories are stir, pickup, putdown, season, flip, and pour and the object categories are bowl, cup, spatula, knife, pan, tongs, plate, oil, pepper, tomato, butter, spreader, bread, spoon, lemon, carrot, meat, and egg. It is computationally unfeasible to create a classification model for each possible combination of actions and objects of arbitrary sizes. For example, there are $6 \times 18 \times 18$ possible interpretations for configurations containing only three concepts.

Each video depicts a sequence of steps of a cooking recipe. The temporal segmentation by the sequential occurrence of actions throughout the videos naturally split them into shorter video clips. This temporal video segmentation produces 309 video clips for training and 359 for testing (evaluation performance). These short videos are the units of interpretations used for performance evaluation purposes, each of them depicting some interaction described by an action performed on objects, for example, stirring carrots in a bowl using a spatula.

4.2 Feature generators

The motion dynamics of each video clip is modeled as a temporal sequence of histograms of optical flow (HOFs), where each HOF is computed from a pair of consecutive frames. This sequence is clustered into three time-ordered sets of temporally consecutive HOFs. Each cluster is represented by the sum of their histograms. Concatenating these three histograms forms the final motion feature histogram to represent the motion dynamics in each video clip. A feature generator is instantiated for each motion feature histogram constructed as described, which represents the motion dynamics in each short video scene.

Object bounding box tracks of detected objects will have feature generator instances associated with each of them. Histograms of oriented gradients are the features used for describing the shape of the objects enclosed in their bounding boxes. Thus, for each object track there is a set of HOGs, one for each bounding box. Note that for each object track, there is at most one bounding box at each frame. An object feature generator will represent a group of HOGs extracted from a sequence of bounding boxes that forms an object track in the video.

4.3 Learning phase

In the learning phase we specify the ontology generator space and bond strengths. Unlike graphical models, such as HMM, DBN, AND-OR graphs, Markov networks, there is no training needed over possible interpretation structures. The generator space is manually specified as a collection of categories of actions and objects that is discovered by parsing the video annotations files provided with the training data set. The bond structure of each action and object generator is specified based on domain knowledge.

For each elementary action and object category, we learn object and action classifiers using linear support vector machines based on bags of HOF or HOG features. We use the training data for this process. These classification models are used to quantify the bond interaction affinities between feature generators and actions/object generators (the support bonds). Specifically, we use multi-classification models for the action and for objects built...
Fig. 3. Semantic interpretations generated by the proposed pattern theoretic framework for different video clips (a–d) are depicted as graph structures of generators. Videos (a–d) increase in visual complexity from left to right and top to down. The feature generators are grounded to their corresponding regions in the images. The semantic descriptions expressed by each interpretative structure above are: (a) Putting down spreader in a bowl containing tomato. (b) Stirring lettuce in a bowl using spoon. (c) Pouring tomatoes from a bowl into another bowl. (d) Putting down tomato in a bowl. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
using linear support vector machines with LibSVM [2]. Because the number of training instances across categories is uneven, we generated synthetic samples using SMOTE [3] for the minority categories (e.g., categories season and flip had only 4 and 7 training instances).

We found that the confusion matrices of the classifier’s diagonals are only partially dominant. This means that the learned models are weak if recognition simply consists of labeling based upon the model’s best prediction scores. There is a noticeable confusion between overlapping categories. For example, there is a great deal of the picking up action involved in flipping action (pickup and flip). Additionally, training instances of objects typically used together add confusion to the classification models. For instance, spatulas are commonly used to stir ingredients in a bowl (bowl and spatula). This is the kind of confusion that we expect to be alleviated by the inclusion of prior knowledge, encoded in the ontological bond affinities.

Ontological bond affinities are learned as a table of frequency values of the co-occurrence of actions–objects and objects–objects in the training data annotations. Ontology constraints derived from the co-occurrence of objects and actions can also be used to overcome classification errors. For instance, the action pickup could be

Fig. 4. More complex semantic interpretations generated by the proposed pattern theoretic framework for different video clips (e–h) are depicted as connected structures of generators. Videos (e–h) increase in visual complexity from left to right and top to down. The feature generators are grounded to their corresponding regions in the images. The pair of videos (g) and (h) are consecutive interactions where multiple detected objects do not participate in the main activity. The semantic descriptions expressed by each interpretative structure above are: (e) Picking up lettuce from two bowls. (f) Pouring tomatoes from a bowl into another bowl. (g) Pouring carrots from a bowl into another bowl. (h) Stirring carrots in a bowl using spatula. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
ruled out in favor of pour to label the action happening in a scene after inferring the presence of cup and oil.

4.4. A Baseline interpretation algorithm

There is lack of a common benchmark that can be used for comparing approaches that produce different types of output. Additionally, comparative works do release code for public use. For this reason, we implemented a baseline algorithm that generates comparable interpretative structures exploring the confidence values output by action and object classification models learned with the linear support vector machine (SVM) algorithm. The baseline algorithm simply returns the object and action nodes connected to the observed features. There are no connections between the object and the action labels; thus, no judgments are made about what action is performed on which object.

4.5. Evaluation metric

An interpretation consists of ontological and feature generators connected by bonds. The performance is measured by calculating the fraction of the number of correct ontological generator labels associated with each feature generator to the total number of feature generators. Using the examples depicted in Figs. 3 and 4, this would be the number of correct label associations (highlighted in blue) divided by the total number of feature generators.

5. Results

In Figs. 3 and 4, we present visual examples of the pattern theoretic framework’s output. They are presented in increasing order of scenario complexity (top-down, left–right). These scenarios are characterized by the manipulation of objects by the cook (e.g., cook puts down spreader in the bowl – Fig. 3(a)) as well as object interactions with other existing objects (e.g., carrots in a bowl being mixed with a spatula – Fig. 4(h)). The bond structures help to pull out labels that are not necessarily the ones declared with the highest classification scores by the SVM classifiers.

The baseline algorithm fails to find interpretations like those depicted in Figs. 3 and 4 since it relies solely on constructing them by looking the labels for which the classification scores were the highest. The proposed inference process seeks to make sense of explanations given by weak prediction values, instead of discarding them, by considering other co-relevant predictions. These improvements come from co-support provided by the labels predicted in the scene.

5.1. Ability to overcome classification errors

We studied the tolerance of our approach for different error rates of feature classification. One scenario consisted of varying the overall feature classification rate from 10% to 60% (Fig. 5(a)). In the second scenario we fixed the feature classification error rate to 50% and varied the correct label rank of the misclassified features from 2 to 5 (Fig. 5(b)). Fig. 5 shows that our pattern theoretic solution is resilient to severe scenarios of low-level processing error (i.e., classification errors); having its performance only slightly and gradually reduced. The pattern-theoretic approach was able to sustain higher performance rates by 20% or more. Note that an approach strongly based on the perfect performance of concept classifiers such the implemented baseline can easily fail in these situations.

5.2. Quantified performance

In Fig. 6(a) we compare the average performance of interpretations by the purely bottom-up baseline with pattern theoretic approach for each test video (a complete recording of one recipe). The interpretation performances are computed based upon the percentage of correct labels predicted on each shot. With the exception of 2 test videos, the baseline approach is outperformed by
ours. Overall, our approach doubled the performance rate. Fig. 6 (b) further demonstrates our approach’s superiority. It shows the histogram of scores for the two algorithms. We observed that the baseline algorithm results in mostly zero scores, whereas ours produced a higher rate of high scores.

6. Conclusion
The proposed pattern theoretic framework offered an elegant solution for integrating ontological constraints and machine learning-based classifiers. It was capable of constructing relevant semantic interpretations of video contents in terms of probabilistic algebraic structures. Additionally, it proved to be robust to increasingly difficult scenarios of feature classification errors. The resulting interpretable structures contain connections between object and action labels, which in turn are grounded to observed features, to express relevant interactions found in the video. The MCMC based simulated annealing performs well in optimizing this energy function. We demonstrated the superiority of our approach over a purely machine learning based approach (SVM) using more than 650 short video clips from the YouCook dataset. The results show that the framework is successful in constructing nearly correct interpretations by leveraging the ontological constraints even when the machine learning classifier is poor and the most confident labels are often incorrect.

This pattern theoretic framework shares similarities with frameworks based on probabilistic graphical models (PGMs). Both employ graphical connected structures for representation and inference can be accomplished by minimizing an energy function. However, there are fundamental differences in the representation, which imparts certain advantages to PT over PGM. Typical graphical model approaches such as MRF and BN are specific cases in the PT framework in which the model is based on a single fixed connector graph \(G\); thus, estimating a probability distribution of configurations of a single subspace \(C(G)\). A collection of generators populating the sites of a connector graph is the same as an assignment for the random variables of an MRF or a BN. In PT, the connector graph \(G\) is variable, both in the number of generators and connections, allowing us to propose interpretations with a larger range of variability in content. For instance, the PGM structure to model the interaction of a person with an object will be different from one that is used to model interaction of two persons with an object. We will have to pre-specify (or learn from prior labeled data) these structures. However, for PT, we just need to specify the energy model between people and object generators and it will be able to handle interaction among any number of people and any number of objects. In other words, PT is easily extensible to handling new situations.

Acknowledgment
This research was supported in part by NSF Grants 1217515 and 1217676.

References