Large-scale Representation Learning of Recipes

Ozan Sener

Cornell University

Teaching Robots How to (Cook/Clean/Fix household items)

Q: Can robot learn "how to ..." by themselves

- How we learn "how to .." ?
  - Ask experts *Robots asking for help [Tellex et.al.]*
  - Read recipes (recipe books, wikihow.com, e-how.com, etc.) *Robots making pancake [Beetz et.al.] (single recipe, hard coded perception, hard coded motion)*
  - Watch people performing these tasks (youtube.com, etc.) *activity recognition [Koppula et.al., Schiele et.al.]*

Large scale, multi-modal, extensive information is available; however, we do not know how to represent/understand them.

Large-Scale Recipes for Daily Activities

Cooking Recipes, Recipes to fix house hold items/cars

Text Based Resources
wikihow.com, etc..

- ✔ Step-by step natural language description
- ✘ Assume basic human knowledge
- ✘ Lack specific details (vague descriptions)

Video Based Resources
youtube.com, etc..

- ✔ Highly detailed and complete information
- ✘ Only a specific example
- ✘ Lots of environment specific/unrelated information

- ✘ There are many recipes (both text and video) for a single task -task is ambiguous- (eg 281,000 video, 600 text recipes for *how to tie a bow tie*)

How to Understand Large-Scale Recipes

Representation

- Represent different recipes of a single task as a structured DAG ?
- Combine the generality of the text based recipes with the detailed/structured knowledge in videos ?

Challenges

- (Weakly/Un)Supervised learning using large-scale unlabelled/weakly labelled data
- Noisy data (unrelated search results etc.)
- Incomplete data (some recipes will miss some of the steps)
- Ambiguity of the language

Related Problems

Choose the optimal path over the DAG via

- (Human User) interactive visualization of the DAG ?
- (Robot User) environment specifications and hardware limitation (e.g. physical simulation) ?

Nodes with Automatic Tests

- Robots/Humans can fail during the execution
- Can we automatically find a test for each node ?
- Can we locate the state within the DAG ?

Representation Problem

Assumptions

- Multiple recipes of a single task can be represented as a DAG such that
  - Edges represent actions and nodes represent states with a unique start and end node

Representing the Data

- Each video represents a path between start and end node (with omitted steps).
- Each text based recipe represents a set of paths between start node and end node.

Machine Learning Question

- Unsupervised multi-modal representation learning to detect activities/objects available in text and video.
- Given set of incomplete paths between start and end, how can we reconstruct the graph ?

Incomplete Review of the Literature

Web enabled robots

- **Bakebot [Bollini et.al.]:** Manually convert the available recipe to finite state machine and make robot perform it successfully .
- **Recipe Language Understanding [Beetz et.al., Mori et.al.]:** Converting a specific recipe to a formal plan via language parsing and processing via pre-learned ontologies.
- **Robots Making Pancake [Beetz et.al.]:** Experimental validation that robots can follow the extracted recipe (no learning). Predefined motion/perception libraries.

Incomplete Review of the Literature

Co-representing text and video

- **Text from image [Yang et.al.]:** Text description via object descriptors and a language model.
- **Grounding action descriptors [Regneri et.al.]** Given videos and text description, find similar action verbs. Later used for paraphrase detection
- **Domain adaptation [Elhoseiny et.al.]** Given complete text descriptions and incomplete visual information (missing classes), how to train visual classifier.
- **Learning visual meanings of text [Zitnick et.al.]:** Given clip-art images and texts, find visual meanings of words like near, run-to, etc.

Incomplete Review of the Literature

Machine learning for graph completion/representation

- **Plot Graph [Li et.al.]:** Generate story-graph from crowd source stories. Output is graph represented since input is set of nodes.
- **Network Inference [Nowak et.al., Kubica et.al.]:** An EM algorithm to recover the network topology from the onordered paths. There is no uncertainty about the nodes, only the ordering.
- **Learning Bayes Networks [Teysier et.al.]:**
- **StoryLine from Videos and Texts [Gupta et.al.]:** Fully supervised, only and/or graph structure.

Datasets/Available Tools

Text data:

- (Large-scale) Wikihow.com ontology / (Cyc+WordNet)
- (Large-scale) Some recipe web pages has nice interface/APIs to crawl (cookingforengineers.com)

Visual data:

- (Large-scale) some of wikihow.com entries have video/image and they are accessible via API
- (Large-scale) youtube-dl + youtube search API
- (Small) CAD-120 dataset of everyday tasks

Text+Visual data:

- (Small) Cooking dataset of videos and text descriptions
- (Small) MSR data of clip-arts representing some daily activities with text descriptions

Preliminary Set-up

(Large-scale) wikihow.com/youtube: After manually choosing a few tasks, crawl wikihow.com recipes and youtube videos.

- Wikihow crawler is already implemented (there is just one recipe for a single task since this is a wiki)
- Youtube crawler is already implemented (5k videos per-day limit)

(Small-scale) Basic Indoor Activities: Enrich the CAD-120 with text based recipes, and use it as an evaluation set-up.

NLP Tools: Experimenting with POS tagging/word ontologies to detect synonyms etc. We can use the existing basic setup without changing anything (Stanford CoreNLP, Wordnet, Cyc, Freebase)

Initial Idea

- **Object Co-Proposal:** Find the object proposals over the multiple set of videos for the same task.
- **Dimensionality Reduction:** Represent the small clips by using spatio-temporal relations of the objects. Concatenate with text captions (available for $\sim$ half of the videos - either user provided or ASR).
- **Subtask proposals:** Cluster the low dimensional visual space+language space.
- **Generate DAG Graph**
  - **M:** Estimate a CTMC over the set of videos
  - **E:** Compute the probabilities of the videos

Object Co-Proposal

Well studied problem in computer vision. Two successful solution directions are:

- *Category Independent Object Proposals*: Find super-pixels and propose regions. Rank them using a set of features and learning from a dataset (Relatively fast and accurate).
- *Constrained Parametric Min-Cut/Max-Flow*: Use a pre-learned potential to define Min-Cut/Max-Flow energy function. Generate multiple solutions by varying the parameters (Most accurate solutions, extremely slow).

Vidoe Object Co-Proposal - Keysegments

- Extension of the *Category Independent Object Proposals* to video setting.
- Basically scoring each region with saliency and motion features. Express similarities as distance of unnormalized histograms. Then, finding the best cluster by using spectral graph clustering. $\max \frac{u^T A u}{u^T u}$
- There is no trivial extension to multi-video case
  - If we separately process each video, there are lots of environment specific objects.
  - If we concatenate all object proposals from all videos; there is a serious problem of computational complexity and the result might not be the common object in all videos.

Multi Video Co-Keysegments

Solving $\max \sum_i \frac{u_i^T A_i u_i}{u_i^T u_i} + \sum_i \sum_j \frac{u_i^T A_{ij} u_j}{u_i^T u_i u_j^T u_j}$

- Still computationally intractable.
- Solution is keeping video relations to k-nn graph.

Proposed Algorithm:

- Crate k-nn graph by using language distance between video descriptions over Youtube.
- Solve $u_i^0 = \arg\max \frac{u_i^T A_i u_i}{u_i^T u_i}$ via power iteration
- Iteratively solve $\max \sum_i \frac{u_i^T A_i u_i}{u_i^T u_i} + \sum_i \sum_j \frac{u_i^T A_{ij} u_j}{u_i^T u_i u_j^T u_j}$

Solving $\max \sum_i \frac{u_i^T A_i u_i}{u_i^T u_i} + \sum_i \sum_j \frac{u_i^T A_{ij} u_j}{u_i^T u_i u_j^T u_j}$

$$r^1(u_i) + r^2(u_i, u_j) = \sum_i \frac{u_i^T A_i u_i}{u_i^T u_i} + \sum_i \sum_j \frac{u_i^T A_{ij} u_j}{u_i^T u_i u_j^T u_j}$$

$$\nabla_{u_i} r^1(u_i) + r^2(u_i, u_j) =$$

$$= 2A_i u_i - 2u_i r^1(u_i) + \sum_j \frac{A_{ij} u_j}{u_j^T u_j} - 2 \sum_j u_i r^2(u_i, u_j)$$

If we equate to 0 and write the fixed point iteration,

$$u_i^{t+1} = \frac{1}{r^1(u_i^t) + r^2(u_i^t, u_j^t)} \left[A_i u_i^t + \frac{1}{2} \sum_j \frac{A_{ij} u_j^t}{u_j^{t^T} u_j^t} \right]$$

Multi Video Co-Keysegments - August 5

Previous idea failed due to the

- Dimensionality mismatch.
- Non-convex energy function.

Solution:

Update the energy function to have dimensionality match

$$\max \sum_i \frac{u_i^T A_i u_i}{u_i^T u_i} + \sum_i \sum_j \frac{u_i^T A_{ij} u_j}{u_i^T \mathbb{1}\mathbb{1}^T u_j}$$

Intuitive Explanation:

Maximize the normalized cut and average fit within cluster entries

Solving $\max \sum_i \frac{u_i^T A_i u_i}{u_i^T u_i} + \sum_i \sum_j \frac{u_i^T A_{ij} u_j}{u_i^T \mathbb{1}\mathbb{1}^T u_j}$

Good news:

- Energy function is quasi-convex.
- If we fixed the coordinate, it is quasi-linear.
- It can be optimized by using sub-gradient method.

Gradient is:

$$\nabla_{u_i} = \frac{2A_i u_i - 2u_i r^1(u_i)}{u_i^T u_i} + \sum_j \frac{1}{u_i^T \mathbb{1}\mathbb{1}^T u_j} \left(A_{i,j} u_j - u_j^T \mathbb{1} r^2(i,j) \right)$$

Experiment results for 2 Videos case

- Independent Normalized Cut
- Co-Localization (Keving Tang, Fei-Fei Li)
- Co-Cluster

Next Week(s) Plans

- 10 Videos experiment
- Improve Object Co-Proposal Performance
  - Try different set of region proposals
  - Find the best features (currently it is un-normalized histogram)
  - Optimize parameters
- Represent each frame as histogram of clusters and their motions

Time Stamp

August 26

Recipes to Work On

- Hard Boil an Egg (Objects are extracted from 10 Videos)
- Make Scrambled Eggs (Pre-processing is done, objects will be extracted)
- Poach an Egg (Pre-processing is done, objects will be extracted)

Scaling Issues

- Previous Implementation was not scalable
  - Current implementation is fully parallel.
  - It uses L-BFGS to find the the Co-Proposals.
- Improve Object Co-Proposal Performance
- Try different set of region proposals
- Find the best features (currently it is un-normalized histogram)
  - Replaced the region proposals with the CPMC(Constrained Parametric Min-Cut)
  - As a feature, two options are implemented Dense Sift and histogram of LAB Color Space.

10 Videos test results

Representation of the Frame

- To represent object proposals, we can use either GMM or mean of the extracted histograms/sifts. In other words, given cluster i, we represent it as $\bar{x}_i$.
- From wikiHow entry, find all salient action verbs, salient object names and their synonyms.
- In order to represent each frame, I propose;
  - Propose M object segments from the frame.
  - For each segment, find the nearest cluster(Co-Proposal) and assign to it.
  - Compute the histogram of segments.
  - If subtitle exist, compute the histogram of words.
  - Concatenate two histograms.

Next Week Plan

- Running object proposals on 3 Recipes (10 Videos each).
- Representing each frame.
- Running baseline clustering.

Time Stamp

September 2

This Week

- (Not Finished) Running object proposals on 3 Recipes (10 Videos each).
  - Data is too noisy, I need to manually discard some videos.
- (Baseline Version) Representing each frame.
- (Baseline Version) Running baseline clustering.

Baseline Clustering

- Although the baseline model do not include any temporal reference, resulting clusters are continuous (mostly)
- Language(Sub-title) information seems noisy, but there is still nice anchor words like drain etc.

Problem Definition

- Is solving the problem without wikiHow **possible** ?
- Is solving the problem without wikiHow **more interesting** ?

Evaluation

We need to evaluate;

- Temporal segmentation
- Representing video as sequence of sub-tasks
- Language reference quality.

Metric;

- $\Delta_T =$ IOU metric for temporal segments after labelling.
- $\Delta_S =$ Smallest string matching distance between ground truth and resulting sequence.
- $\Delta_L =$ F-measure between extracted action/object names vs ground truth action object names.

Next Week(s) Plan

- Running baseline s on 3 Recipes (10 Videos each).
- Fixing bugs/optimizing parameters and finalizing the baseline.
- Language Only, Vision Only Tests.
- Measuring performance.
- Labelling videos.

Representing Recipes

Any recipe can be represented as admissible set of sub-tasks, their ordering requirements and their co-occurrence properties.

Co-occurrence

Using co-occurrence has problems like:

- Co-occurrence is generally modelled to be transitive, but within the recipes they are not. For example, (boiling, using a poacher) and (boiling, using a cup to poach) co-occurs while poaching an egg; but, (using a poacher, using a cup to poach)
- Number of admissible co-occurrence is high; hence, we need more data.

Hence, we need to mode **co-not-occurrence** for sparsity and intuitive understanding.

Ordering

Some sub-tasks have ordering requirements like pan need to be filled with water before boiling. Hence, we need to model a set of ordering requirements as well.

Model

- Since sub-tasks are latent within the application, we can not model co-not-occurrence and ordering over sub-tasks.
- We can model these relations over the atomic concepts which sub-tasks are using. Hence, We model them over extracted visual objects and extracted words.

Model

- Let's denote set of visual objects as $v_0 \dots v_{M^v}$ and language words as $l_0 \dots l_{M^l}$.
- We define Co-Not-Occurrence requirements over a matrices $C^{NO,v}$ and $C^{NO,l}$ as $C^{NO,v}_{i,j} = 1$ if v_i and v_j can not occur together. For example, the partial matrix for poaching egg case:

	Water	Pan	Tap	Gelatin	Poacher	Cup
Water		0	0	0	0	0
Pan	0		0	0	0	0
Tap	0	0		0	0	0
Gelatin	0	0	0		1	1
Poacher	0	0	0	1		1
Cup	0	0	0	1	1	

Model

- $C^{NO,\cdot}$ is sparse
- $C^{NO,\cdot}$ is symmetric
- If $C^{NO,\cdot}_{i,j} = 1$, $C^{No,\cdot}_{i,:}$ is similar to $C^{No,\cdot}_{j,:}$. Since, they can be replaced.
- We can relax the $0, 1$ to $[0, 1]$ and obtain the matrix from the observed data.

In order to obtain the sub-task co-not-occurrence,

- Given visual histogram $x^v_0 \dots x^v_K \in R^{M^v}$ and language histogram $x^l_0 \dots x^l_K \in R^{M^l}$ of sub-tasks, we can sample co-not-occurrence set $\mathcal{C} \subset [0 \dots K] \times [0 \dots K]$ as;
  - $P((i,j) \in \mathcal{C}) \sim x^{vT}_i C^{NO,v} x^v_j + x^{lT}_i C^{NO,l} x^l_j$