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)

Preliminary Set-up

Subtitle:

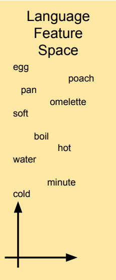
Lorem ipsum dolor
sit amet, consectetur
adipiscing elit.

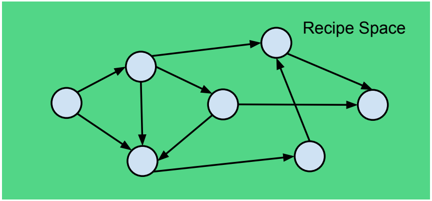
Lorem ipsum dolor
sit amet, consectetur
adipiscing elit.

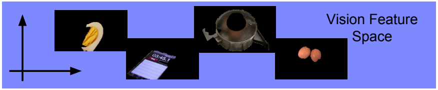
YouTube

wikiHow

- Sort the eggs. Place them in a bowl of salt water—if the egg sinks to the bottom it means it is fresh; if it floats at the top, get rid of it.
- Place the fresh eggs gently in an empty pot. Fill the pot with enough cold tap water to cover the eggs completely.
- Add a pinch of salt to the water.
- Put on a lid. Bring the water to the point of boiling over high heat.
- Stop the cooking process. To see if the egg is hard boiled, whirl it fast on a table. If it turns fast, it is hard boiled.







Subtitle:

Lorem ipsum dolor
sit amet, consectetur

Lorem ipsum dolor
sit amet, consectetur

Lorem ipsum dolor
sit amet, consectetur

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

- Tractable if we restrict video relations to k-nn graph.

Proposed Algorithm:

- Crate k-nn graph by using language distance between video descriptions over Youtube.
- Solve $\max \sum_i \frac{u_i^T A_i u_i}{u_i^T u_i} + \sum_i \sum_j \frac{u_i^T A_{ij} u_j}{u_i^T \mathbb{1}\mathbb{1}^T u_j}$

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:

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

Representing Recipes

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

Observation

- Let’s denote set of visual objects as $v_0 \ldots v_{M^v}$ and language words as $l_0 \ldots l_{M^l}$.
- 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 do not occur together. For example, the partial matrix for poaching egg case:

$$
\begin{array}{lcccccc}
 & \text{Water} & \text{Pan} & \text{Tap} & \text{Gelatin} & \text{Poacher} & \text{Cup} \\
\text{Water} & & 0 & 0 & 0 & 0 & 0 \\
\text{Pan} & 0 & & 0 & 0 & 0 & 0 \\
\text{Tap} & 0 & 0 & & 0 & 0 & 0 \\
\text{Gelatin} & 0 & 0 & 0 & & 1 & 1 \\
\text{Poacher} & 0 & 0 & 0 & 1 & & 1 \\
\text{Cup} & 0 & 0 & 0 & 1 & 1 &
\end{array}
$$

Model

- $C^{NO,\cdot}$ is symmetric
- 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;
  - $P((i, j) \in \mathcal{C}) \sim {x^v_i}^T C^{NO,v} x^v_j + {x^l_i}^T C^{NO,l} x^l_j$

Model

Similarly for ordering, we also model it over the extracted visual and language words.

- We define ordering requirements over a matrices $C^{OR,v}$ and $C^{OR,l}$ as $C^{OR,v}_{i,j} = 1$ if v_i has to occur after v_j. For example, the partial matrix for poaching egg case:

$$
\begin{array}{c|cccccc}
 & \text{Water} & \text{Pan} & \text{Tap} & \text{Gelatin} & \text{Poacher} & \text{Cup} \\
\text{Water} & & 1 & 1 & 0 & 0 & 0 \\
\text{Pan} & 0 & & 0 & 0 & 0 & 0 \\
\text{Tap} & 0 & 0 & & 0 & 0 & 0 \\
\text{Gelatin} & 1 & 1 & 1 & & 0 & 0 \\
\text{Poacher} & 1 & 1 & 1 & 0 & & 0 \\
\text{Cup} & 0 & 0 & 0 & 0 & 0 &
\end{array}
$$

Model

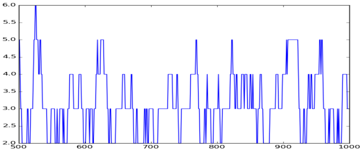
- Either $C_{i,j}^{OR,\cdot} = 1$ and $C_{j,i}^{OR,\cdot} = 0$ or $C_{i,j}^{OR,\cdot} = 0$ and $C_{j,i}^{OR,\cdot} = 0$
- Although it looks quadratic; by keeping an extra observation vector, data can be computed in linear time.
- We can relax the $0, 1$ to $[0, 1]$ and obtain the matrix from the observed data.

In order to obtain the sub-task ordering,

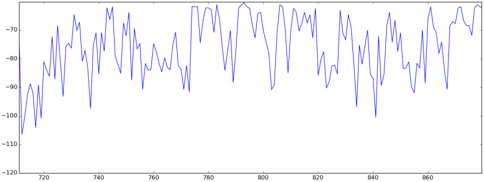
- Given visual histogram $x_0^v \dots x_K^v \in R^{M^v}$ and language histogram $x_0^l \dots x_K^l \in R^{M^l}$ of sub-tasks, we can sample ordering set $\mathcal{C} \subset [0 \dots K] \times [0 \dots K]$ as;
  - $P((i,j) \in \mathcal{C}) \sim {x_i^v}^T C^{OR,v} x_j^v + {x_i^l}^T C^{OR,l} x_j^l$

Experiments - Artificial Data

- It looks like algorithm is converging to the correct number of features (4 in this experiment)



- Log Likelihood seems like converging as well (starting one is -140)



Experiments - Artificial Data

- It looks like algorithm is recovering ordering etc.

Real:

$$\begin{matrix} 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. \\ 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \end{matrix}$$

Estimated:

$$\begin{matrix} 0. & 0. & 0.98 & 0.07 & 0. \\ 0. & 0. & 0. & 0. & 0.60 \\ 0.98 & 0. & 0. & 0. & 0. \\ 0.07 & 0. & 0. & 0. & 0. \\ 0. & 0.60 & 0. & 0. & 0. \end{matrix}$$

Experiments - Artificial Data

- NPBayes is clearly outperforming original HMM even with correct number of clusters.
- Intersection over union scores are:

HBPRecipes:	0.5339
EM-HMM w/GT:	0.4576
EM-HMM w/o GT:	0.3391

Problems & Questions

- Computational complexity: Current system is running in couple hours for 100 videos 20 activities recipe
- Results: We expect results to be better at least on artificial data, but I suspect there might be problems in implementaiton and HMMs are behaving weird when we feed 100 frames.
- Evaluation: We need better evaluation metric

Plans

- Continue to play with the implementation until we reach confident numbers.
- Starting to run and experiment the vision pipeline.