Unsupervised Learning Approaches to Intention Recognition

AUTHOR: ALEXANDROS ZENONOS
SUPERVISOR: DR FARIBA SADRI

Contents

Motivation

Motivation (1/3)

Mohamed Atta

Marwan al-Shehhi

Motivation (2/3)

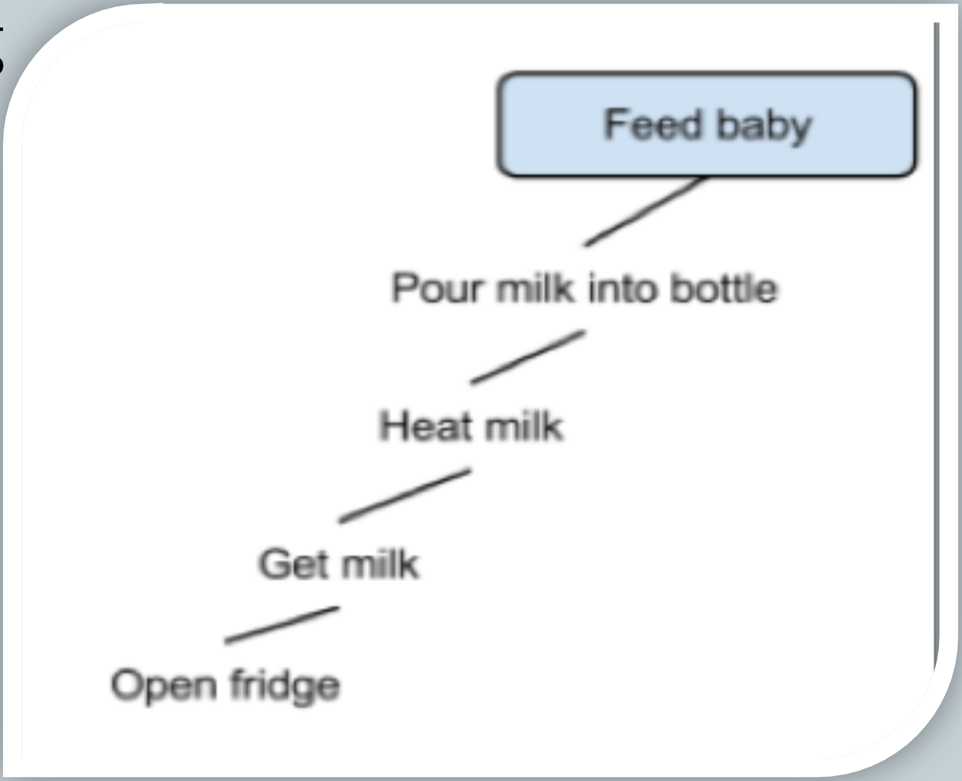
- Assist occupant with dementia or Alzheimer's disease carry out his/her daily routine.

- Strategy games, Intrusion Detection Systems, Elder care etc.

WARCRAFT®

Motivation (3/3)

- The system can recognise the intention of feeding the baby.



Summary & Challenges

Summary of our Work

- Model intention recognition in a binary form.
- Find suitable combination of unsupervised learning techniques, through experimentation.
- Develop an incremental intention recognition (IIR) system.
- Test IIR system on real datasets.
- Extend IIR system with temporal constrains and sensor reliability.

Challenges

- Understanding of background on Intention Recognition and Unsupervised learning.
- Finding a suitable model for the system.
- Convert real data to binary form that is compatible with our model.

Background

Intention Recognition (1/2)

- What is Intention Recognition (IR)?
  - It is the task of recognising the intentions of an agent (human or otherwise) by analysing his observed actions, the changes in the state (environment) resulting from his actions, the context and any information about (possibly learned) expected behaviour of the observed agent
  - IR can be classified as:
    - Intended: The agent wants his intentions to be identified and intentionally gives signals to be sensed by other (observing) agents. e. g. language understanding where the speaker wants to convey his intentions

Intention Recognition (2/2)

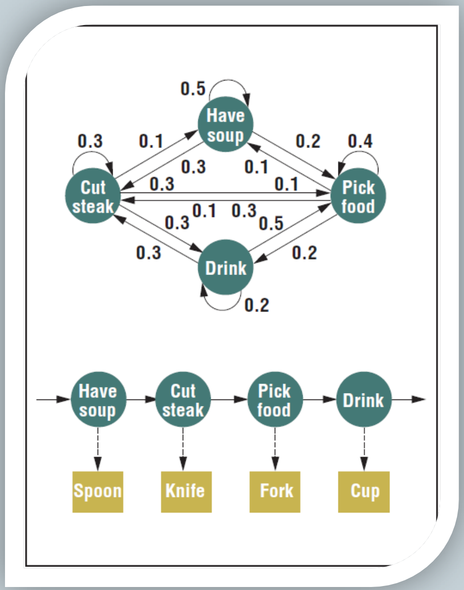
- Keyhole: The agent does not care whether or not his intentions are identified; he is focused on his own activities, which may provide only partial observability to other agents. e.g. dementia/Alzheimer patient

- Adversarial: The agent is hostile to his actions being observed. e.g real strategy game player (Warcraft)

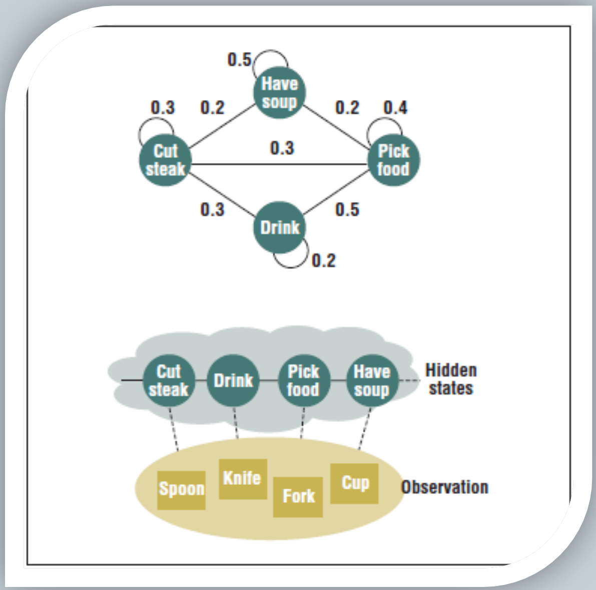
- Diversionary: The agent attempts to conceal his intentions by performing misleading actions. e.g intrusion in a network system.

Current Approaches (1/5)

Hidden Markov Model



Conditional Random Fields

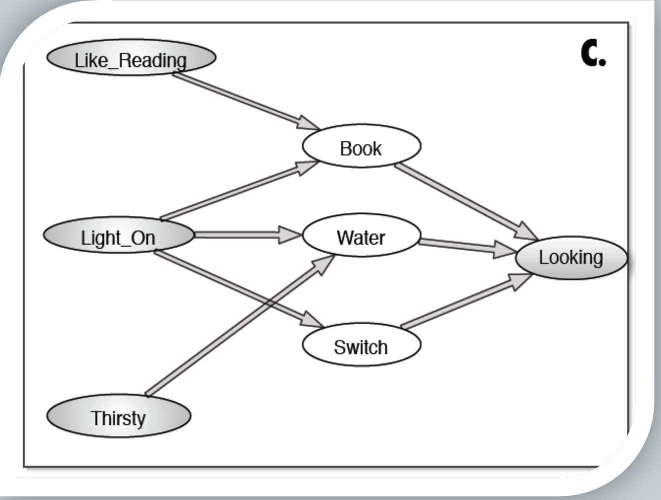


Based on the objects used (spoon, knife, fork, or cup, which are the observable variables), we can infer the HMM states and their transitions.

Observations aren't randomly generated, and hidden states depend on global observations.

Current Approaches (2/5)

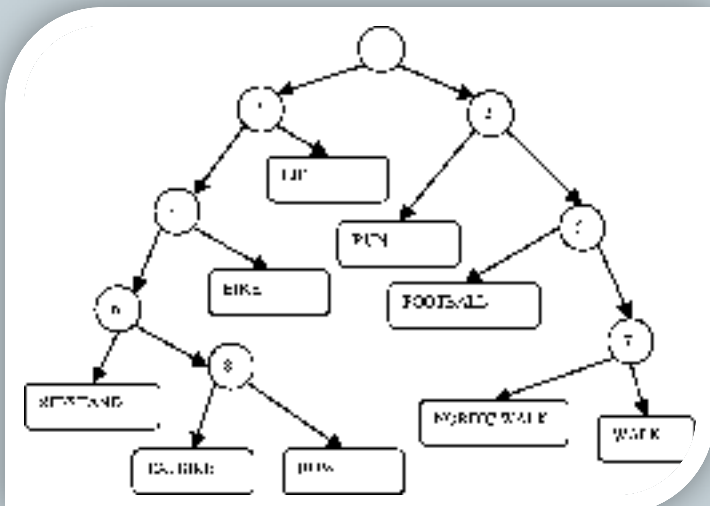
Bayesian Network



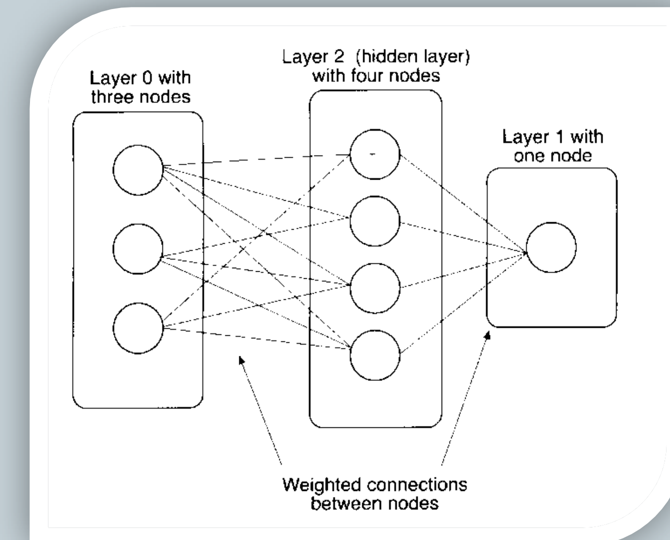
Constructing a three-layer Bayesian Network, upon which intention recognition is performed.

Current Approaches (3/5)

Decision Tree



Artificial Neural Network

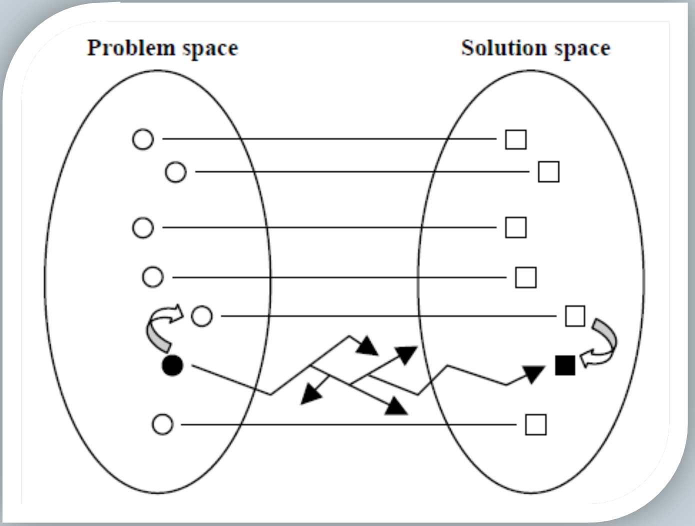


e.g If footsteps are detected, but not the characteristics of running or Nordic walking, the activity falls through the tree to a class "walk".

Use of resilient back propagation as the training algorithm was used as the ANN classifier.

Current Approaches (4/5)

Case Base Reasoning



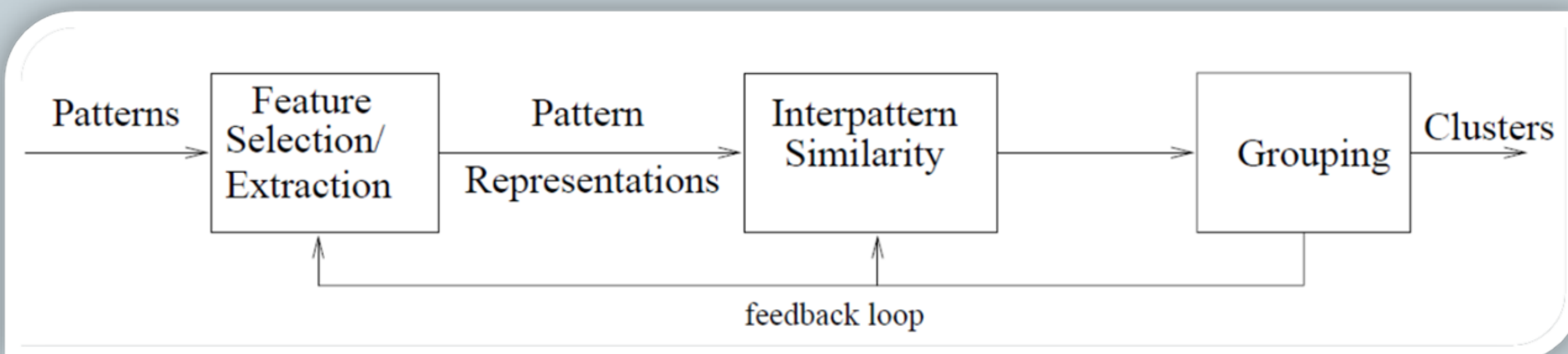
Rather than solve every problem from scratch, case-based reasoning uses past experience in the form of previously solved problems to solve new problems that share similar situations.

Current Approaches (5/5)

- Abduction
  - room-is-hot ← heating-is-on
- Weighted Abduction
  - building(X, public) ^ door-open(X)$^{0.1}$ → may-enter(X)
  - building(X, private) ^ door-open(X)$^{0.9}$ → may-enter(X)

Unsupervised Learning Approaches (1/2)

- What is Unsupervised Learning?
  - Find hidden structures or patterns in data
  - The data have no target attribute
- Steps in Unsupervised Learning
  - Dimensionality reduction (feature extraction/feature selection)
  - Clustering



Unsupervised Learning Approaches (2/2)

- Work of Afroditi Xafi
  - Find similarities between pair of actions (Similarity Matrix)
  - Transform matrix into Euclidean plane (Laplacian Eigenmap)
  - Fuzzy C-Means Algorithm (Membership Matrix)
  - Incremental Intention Recognition

- Work of Wang
  - Create similarity matrix and transform it into Euclidean plane as Xafi.
  - Compare Fuzzy C-Means, Possibilistic C-Means, Improved-Possibilistic C-Means
  - Incremental Intention Recogntion using I-PCM

Unsupervised Learning

Dimensionality Reduction

- What is dimensionality reduction?
  - A procedure applied to a dataset in order to obtain a reduced representation of the original data.
- Why we want to do this?
  - Clustering techniques work more efficiently when dealing with low-dimensional data.
  - Possibility of visualizing the data .
- Famous techniques: PCA, Laplacian Eigenmaps, Diffusion Maps, t-SNE, Isomap, LLE, MDS

Clustering (1/2)

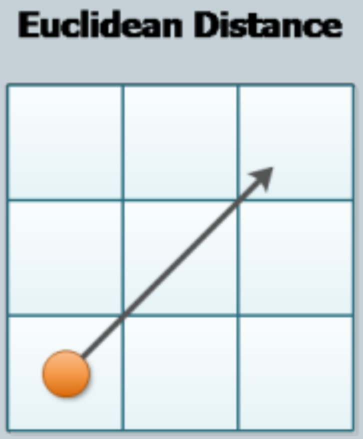
- What is clustering?
  - A way to partition a dataset in a set of meaningful sub-classes or clusters.
  - Assuming that the data were generated from a number of different classes, clustering aims to group data belonging in the same class together.
  - Datapoints in a dataset are said to be close to each other based on some notion of similarity (similarity metric).
  - Most important categories of clustering algorithms:
    - Hierarchical VS Partitional
    - Hard VS Fuzzy

Clustering (2/2)

- Clustering Techniques:
  - Agglomerative Hierarchical
  - K-means
  - Mixture of Gaussians
  - DBSCAN

Similarity Measures (1/6)

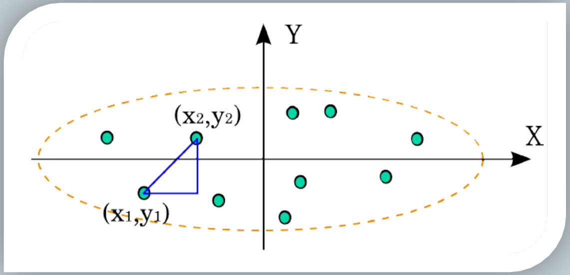
- What similarity metric?
  - Euclidean distance
    - Pythagorean metric
    - It is the "ordinary" distance between two points that one would measure with a ruler .



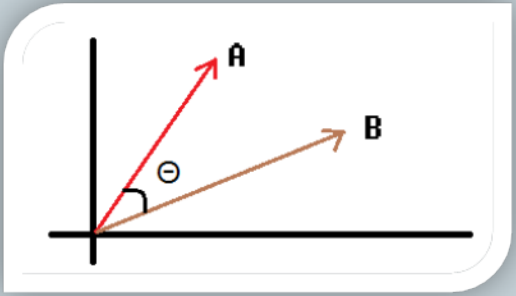
$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Similarity Measures (2/6)

- Mahalanobis distance
  - Provides a relative measure of a data point's distance (residual) from a common point.



- Cosine distance
  - Angle (Θ) between two vectors



Similarity Measures (3/6)

- Jaccard distance
  - The intersection divided by the size of the union of the sample sets (categorical data)
  - $J = 1 - \frac{|A \cap B|}{|A \cup B|}$
- Tanimoto distance
  - Extension of Jaccard distance
  - $T = \frac{A \cdot B}{|A|^2 + |B|^2 - A \cdot B}$

Similarity Measures (4/6)

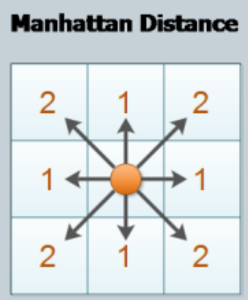
- Hamming distance
  - The number of positions at which the corresponding symbols are different.

"**toned**" and "**roses**" is 3.

- Spearman distance
  - Measures the correlation between two sequences of values

Similarity Measures (5/6)

- Cityblock/Manhattan distance
  - This is simply the number of edges between points that must be traversed to get from "a" to "b" within the grid.



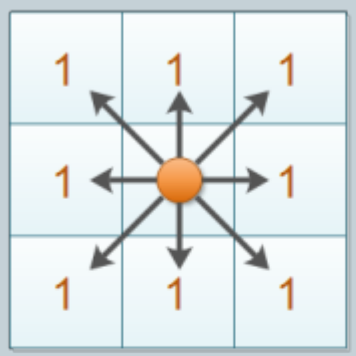
$$|x_1 - x_2| + |y_1 - y_2|$$

- Minkowski distance
  - Generalization of both the Euclidean distance and the Manhattan distance
  - P=1 then Manhattan distance, P=2 then Euclidean distance

Similarity Measures (6/6)

- Chebychev distance
  - The distance between two vectors is the greatest of their differences along any coordinate dimension

Chebyshev Distance



$$\max(|x_1 - x_2|, |y_1 - y_2|)$$

Modelling

Basic Terms

- Terminology used in intention recognition
  - Action: A basic operation an agent can do e.g open the fridge.
  - Plan: A set of actions associated with an agent's goal/intention.
  - Intention: The goal of an agent which is associated with a number of different plans.
  - Plan library: Contains a set of plans.
  - Action stream: A set of actions that are observed of an agent attempting to achieve a goal/intention.

Formulation of Plan Libraries

- How to model actions and plans?
  - Abstract action representation:
    - A1->Open the fridge
    - A2->Pick up dirty clothes
    - A3->Get milk
    - A4->Open cupboard
    - A5->Get glass
    - A6->Turn on TV
    - A7->Heat milk
    - A8->Pour milk to glass
  - Binary representation of actions to achieve a goal (e.g have a warm cup of milk):
    - 1 0 1 1 1 0 1 1

Model Example

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	I
P1	0	1	0	0	1	0	1	1	0	1	I1
P2	0	0	0	0	1	0	1	1	0	1	I1
P3	1	0	0	1	0	0	1	0	0	0	I2
P4	1	0	0	1	0	1	1	0	0	0	I2
P5	0	0	1	0	1	0	1	1	1	0	I3
P6	1	0	1	0	1	0	1	1	1	0	I3

Binary representation of a plan library

I1		I2		I3	
P1	P2	P3	P4	P5	P6
2	5	1	1	3	1
5	7	4	4	5	3
7	8	7	6	7	5
8	10		7	8	7
10				9	8
					9

Abstract representation of a plan library

Experiments

Experiments Setup

- 9 synthetic datasets with different properties
  - Number of intentions
  - Number of plans per intention
  - Total number of actions
  - Plan mutation percentage
  - Noise percentage

Experiment datasets

Properties of datasets used

	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 9
Intentions	3	3	3	3	3	3	3	3	3
Plans per Intention	300	300	200	200	300	300	300	300	300
Maximum number of actions	800	800	300	800	800	800	800	800	800
Plan mutation percentage	10%	30%	30%	30%	30%	30%	30%	30%	30%
Noise percentage	0%	0%	0%	10%	5%	10%	15%	20%	40%

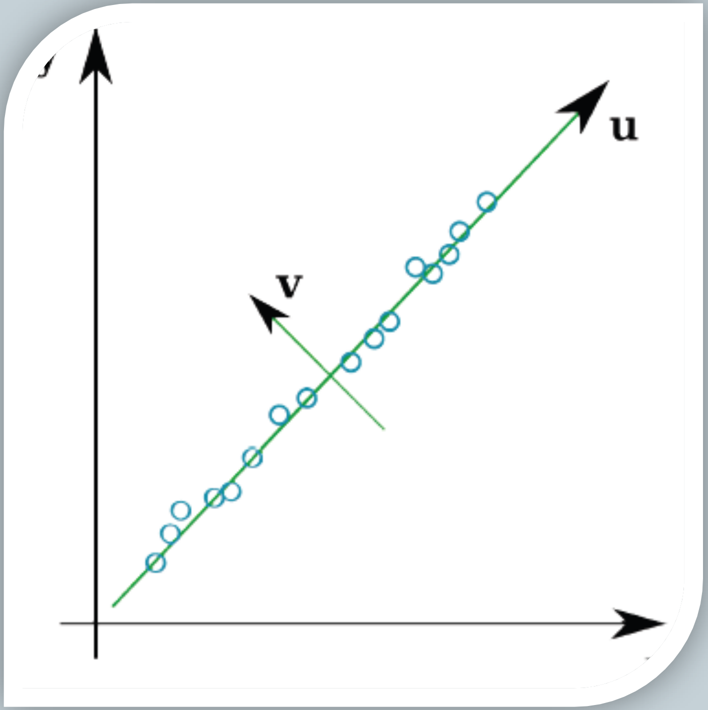
Dimensionality Reduction Technique

- Which dimensionality reduction technique is suitable for IR?
  - No straightforward way to measure the suitability of a dimensionality reduction technique.
  - Use of nearest neighbour preservation ratio. I.e focus on local structure of points.
  - Techniques to compare: PCA, t-SNE, Laplacian Eigenmap, Diffusion Maps

PCA (1/2)

- Enables plans with similar actions to be close to each other in the Euclidean space while dissimilar plans are kept far apart.
- Any plan has some actions that are more important than others, in the sense that we could characterize the plan by them.
- Assumes linear relationship between variables
- Steps:
  - Construct covariance matrix $\Sigma = \frac{U^T U}{N-1}$, where U is the mean centred matrix and N the total number of plans
  - Find m largest eigenvalues/eigenvectors.

PCA (2/2)



Put the axes to the direction of the greatest variability. (eigenvector that corresponds to the largest eigenvalue)

t-SNE (1/4)

- Visualizes high-dimensionality data through dimensionality reduction

- Calculating the pairwise similarities between plans.

- Two plans are similar if the same actions co-occur to both of them and at the same time actions that are present in other plans are not present in them. (Hamming distance)

t-SNE (2/4)

- Construct a conditional probability matrix (p) based on Hamming distances and Gaussian kernel.

$$Pp_ip_j = \frac{e^{-\frac{H_{p_ip_j}}{2\sigma^2}}}{\sum_{\iota \neq \kappa} e^{-\frac{H_{p_ip_j}}{2\sigma^2}}}$$

- Initialize a distance matrix at random (|P|x|P|).

t-SNE (3/4)

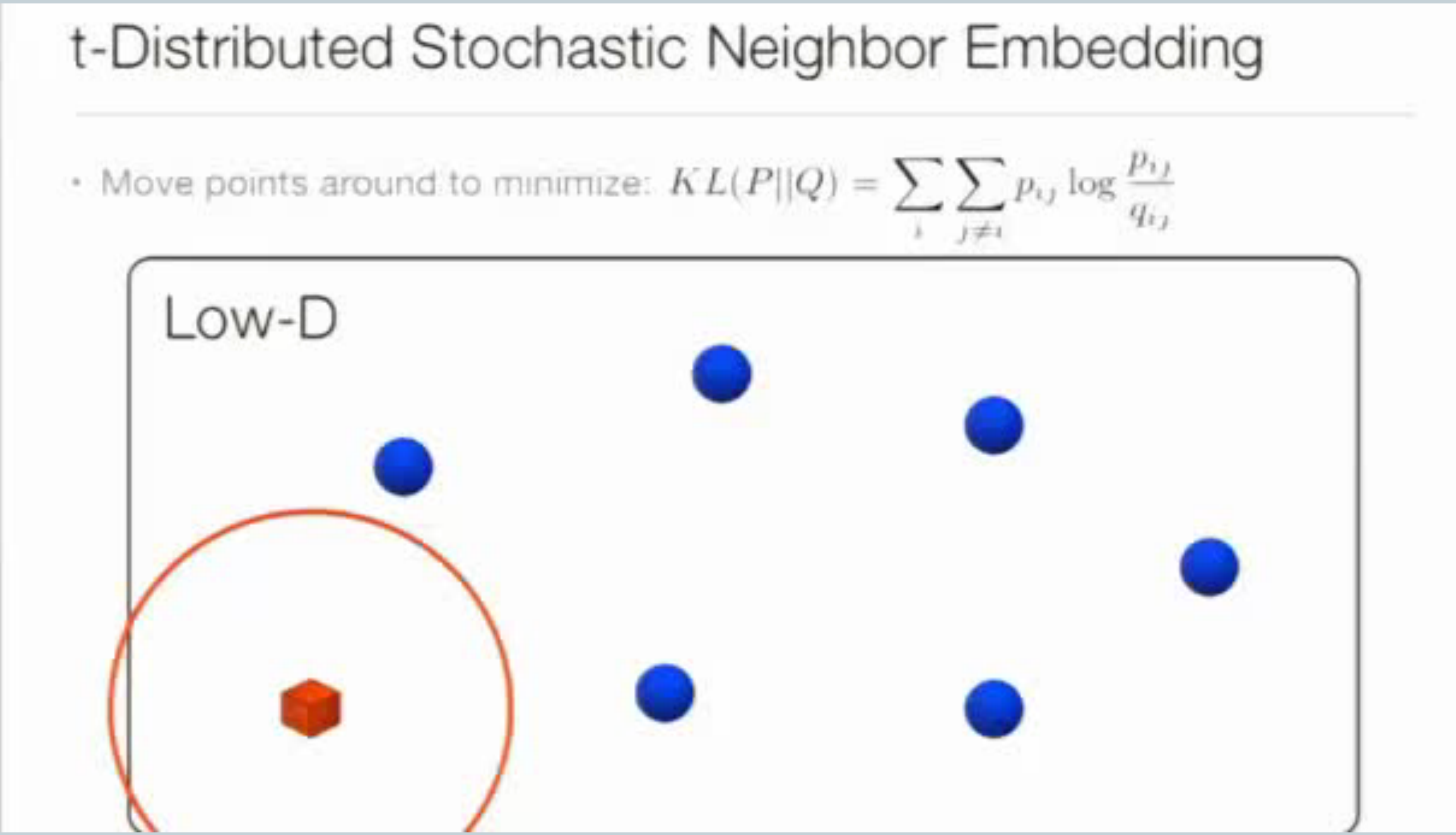
- Construct conditional probability matrix "q" for lower dimension using student t-test kernel.

$$q_{p_i p_j} = \frac{(1+H_{p_i p_j})^{-1}}{\sum_{i \neq k}(1+H_{p_i p_j})^{-1}}$$

- Use of mutual entropy as an objective function. Use of gradient descent algorithm to minimize that function.

- The key idea is that if "p" is the same as "q", then lower dimension counterparts can model the problem.

t-SNE (4/4)


t-Distributed Stochastic Neighbor Embedding
• Move points around to minimize: $KL(P||Q) = \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$
Low-D

Laplacian Eigenmaps (1/2)

- Find non-linear relationships in data.
- Retain the local structure of the datapoints after reducing the dimensions.
- Algorithm:
  1. Use Hamming distance to construct a nearest neighbours matrix.
  2. Build a matrix, W, representing the connections in the graph.
  3. Build diagonal matrix D (degree matrix), with each entry being the sum of each row of matrix W.

Laplacian Eigenmaps (2/2)

4. Create Laplacian matrix $L = D - W$.
5. Solve equation $L f = \lambda D f$, to get eigenvalues/ eigenvectors.

- Take the smallest eigenvalues and corresponding eigenvectors .

Diffusion Maps (1/2)

- Discover non-linear relationships in data
- Focus on retaining the global structure of datapoints i.e not only nearest neighbours should be near in lower dimension but also far away points in high dimension should be far away in low dimension.
- Datapoints are nodes in a graph. Their connection strength is calculated by using their hamming distance.

Diffusion Maps (2/2)

$$w_{p_i p_j} = e^{\frac{-H_{p_i p_j}}{2\sigma^2}}$$

- Normalize over the sum of weights to get the Markov matrix, M.
- Solve eigenproblem Mv=λv

Results

	PCA	t-SNE	Diffusion Maps	Laplacian Eigenmaps
Dataset 1	**68.21%**	67.84%	24.10%	66.56%
Dataset 2	**68.31%**	68.16%	38.89%	66.61%
Dataset 3	**90.80%**	**90.80%**	34.37%	**90.80%**
Dataset 4	**85.48%**	84.01%	35.28%	84.41%
Dataset 5	**66.24%**	65.91%	37.36%	64.96%
Dataset 6	**63.71%**	63.48%	29.28%	62.71%
Dataset 7	**61.65%**	61.30%	35.38%	60.97%
Dataset 8	**59.51%**	59.48%	25.77%	58.63%
Dataset 9	52.99%	**53.36%**	28.37%	52.10%

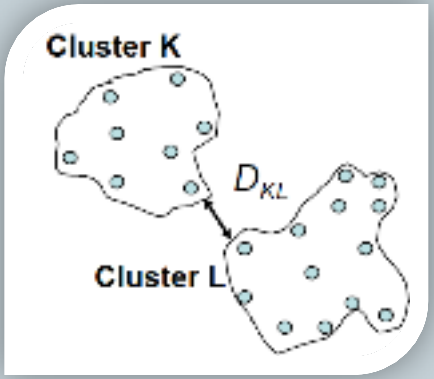
Cluster Analysis Part

Clustering

- Again, no straightforward way to measure suitability of a technique.
- We use a statistical measure, called Silhouette value to get an idea of suitability.
- Silhouette value for each datapoint (plan) is a measure of how similar that point is to points in its own cluster, compared to points in other clusters. (from -1 to +1 or -100% to +100%)

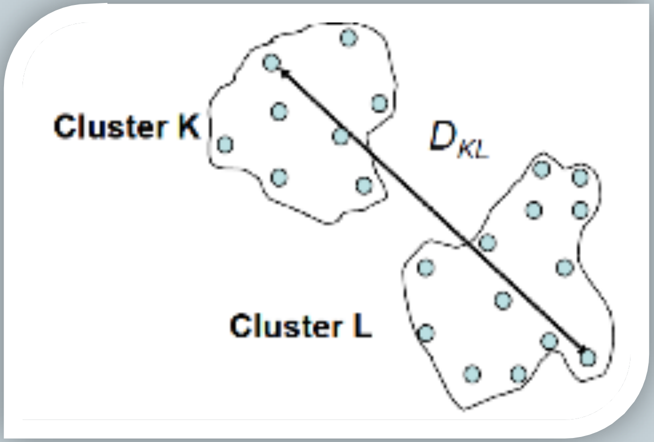
Agglomerative Hierarchical Clustering (1/3)

- Begin with as many clusters as objects. Clusters are successively merged until only one cluster remains.
- Different algorithms to find distance between two clusters:
  - Single Link
    - The distance between two clusters is based on the points in each cluster that are nearest together



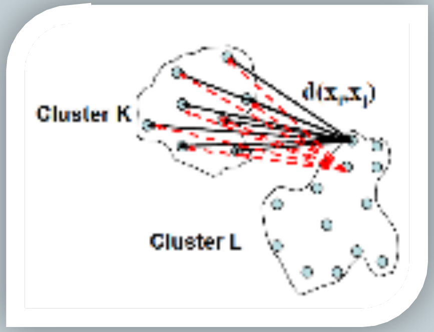
Agglomerative Hierarchical Clustering (2/3)

- Complete Link
  - The distance between two clusters is based on the points in each cluster that are furthest apart



Agglomerative Hierarchical Clustering (2/3)

- Average-Link
  - The distance between clusters is the average distance between pairs of observations



- Ward's method
  - Combine the 2 clusters whose combination results in the smallest increase in ESS (sum of squared deviations from the cluster centroid)
  - Ward's method joins clusters to maximize the likelihood at each level of the hierarchy (minimizes the total within-cluster variance).

Results

- Ward’s method and Average-link with Euclidean distance shown to be the best from agglomerative hierarchical clustering but
  - Average linkage tends to join clusters with small variances.
  - Ward's method is sensitive to outliers.

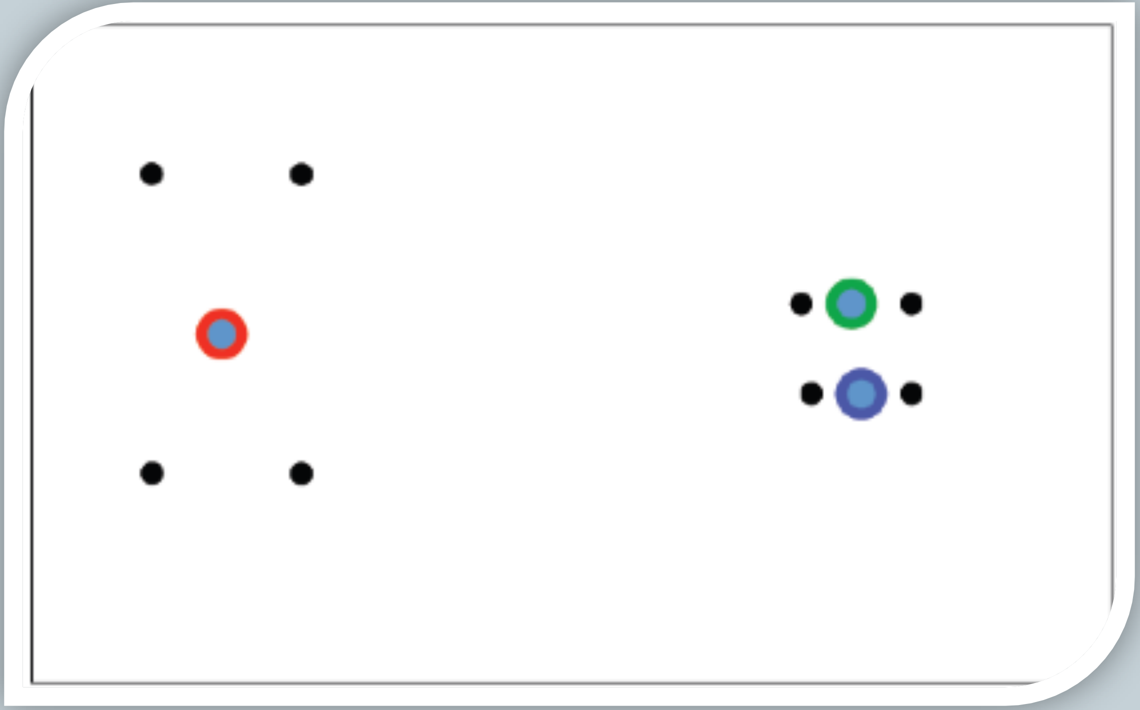
K-means (1/2)

- Partitional algorithm that aims to group data into predefined "k" number of clusters.
- Algorithm:
  1. Initialize centroids in Euclidean space at random
  2. Assign points to their nearest centroid
  3. Refit centroid to the gravity of the points assigned to it
  4. Iterate (go to step 2) until convergence

- Euclidean distance shown to give the best results.

K-means (2/2)

Local Minima when k=3



Mixture of Gaussians

- Similar to K-means but it is fuzzy instead of hard.
- Assumes data were generated from a normal distribution.
- Tries to fit Gaussian distributions to data using EM algorithm.
- Created a membership matrix but plans achieve only one intention (take the most possible one)
- Bad results obtained.

DBSCAN

- Density-Based Spatial Clustering of Applications with Noise
- Works better in large datasets.
- Finds the number of clusters.
- Performs well on synthetic datasets.

Results

- Agglomerative Hierarchical Clustering shown to perform best overall when tested on both Xafi's example (page 23 of Xafi's report) and on a made up domestic scenario.
- DBSCAN fails because points are not dense enough. K-means falls into local minima (it is dependent in its initialization)

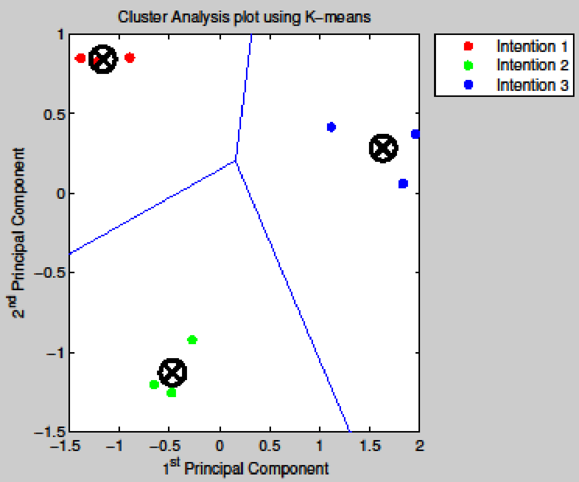
Xafi's Example

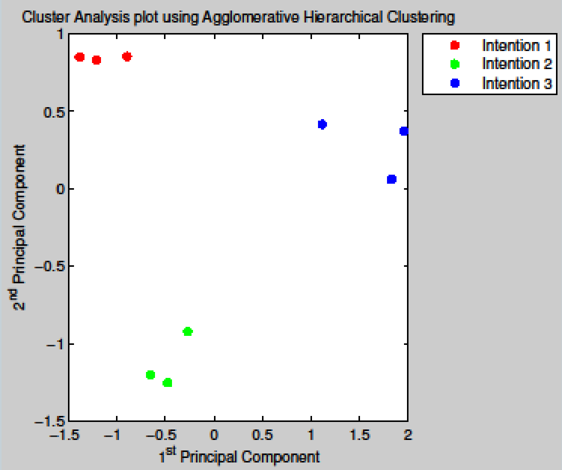
	Hierarchical	K-means	DBSCAN
Mean Silhouette Value	95.31%	95.31%	94.16%

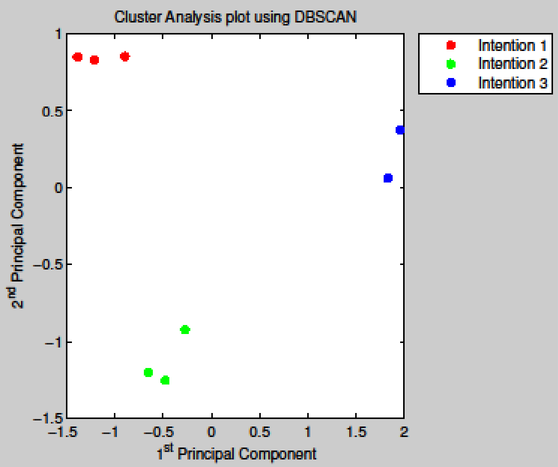
Domestic Example

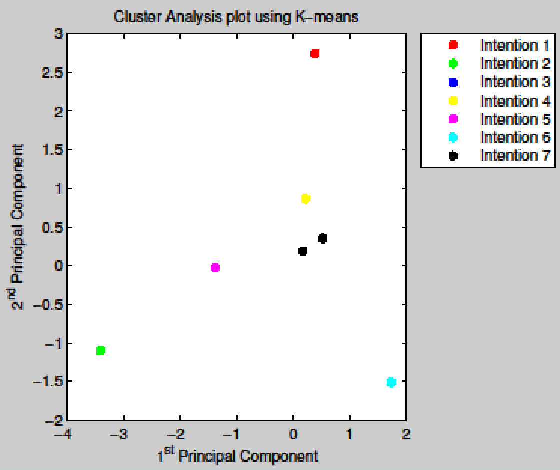
	Hierarchical	K-means
Mean Silhouette Value	100%	90.91%

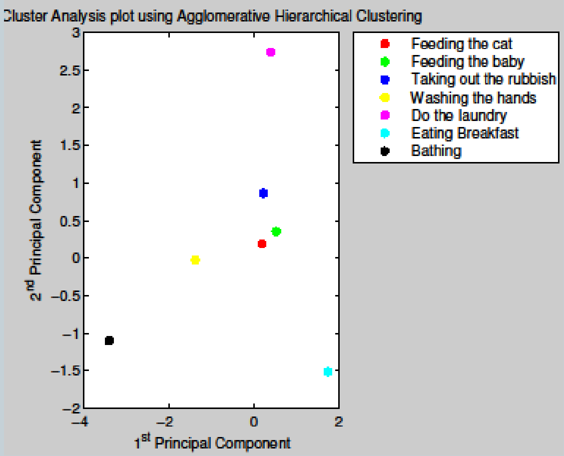
Visualization of Results











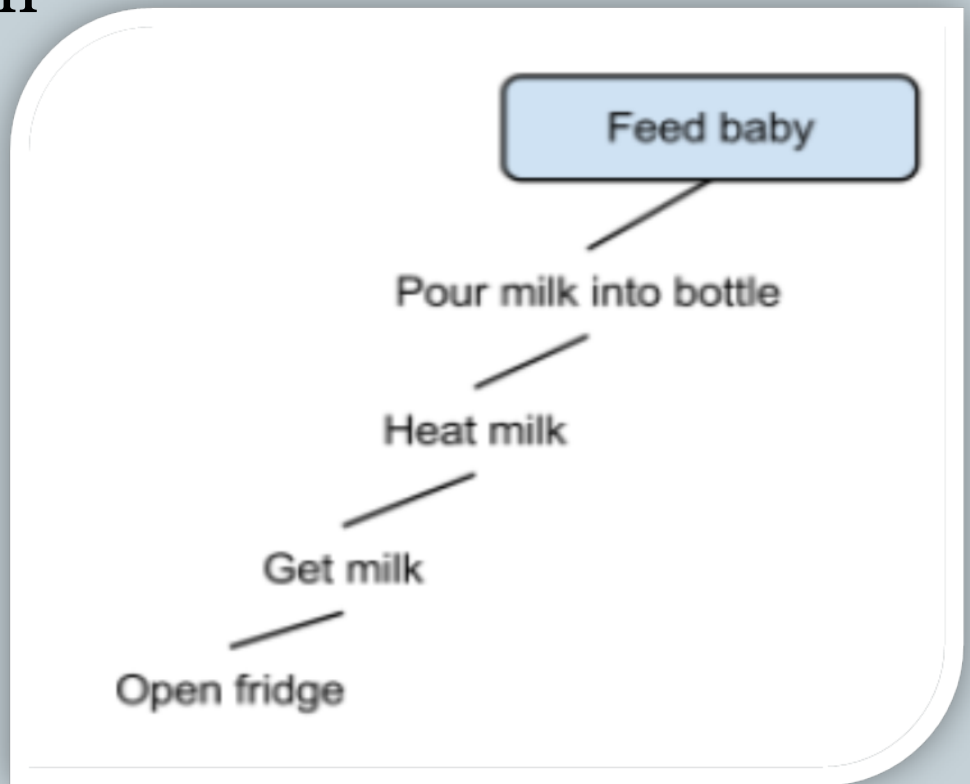
Incremental Intention Recognition

Incremental Intention Recognition

- Incremental intention recognition (IIR) is the problem of recognising the intentions of an agent by (incrementally) observing its actions.
- Make use of the unsupervised learning techniques.
- Two methods were proposed (H1 & H2).

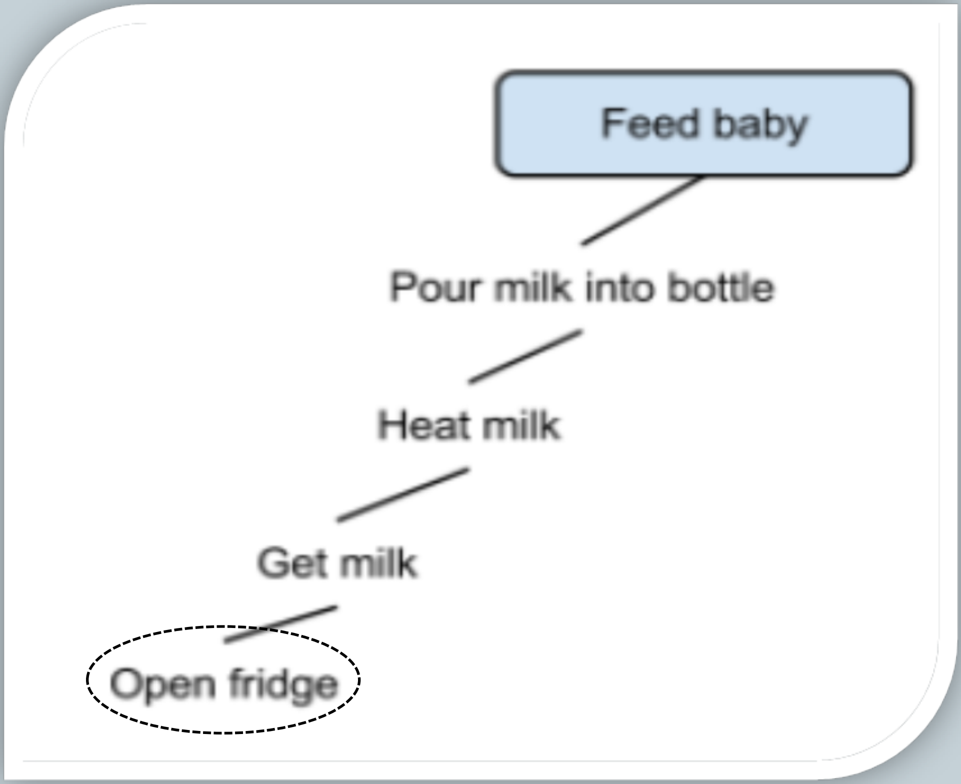
IIR example (1/3)

- Recognise the intention of an agent.



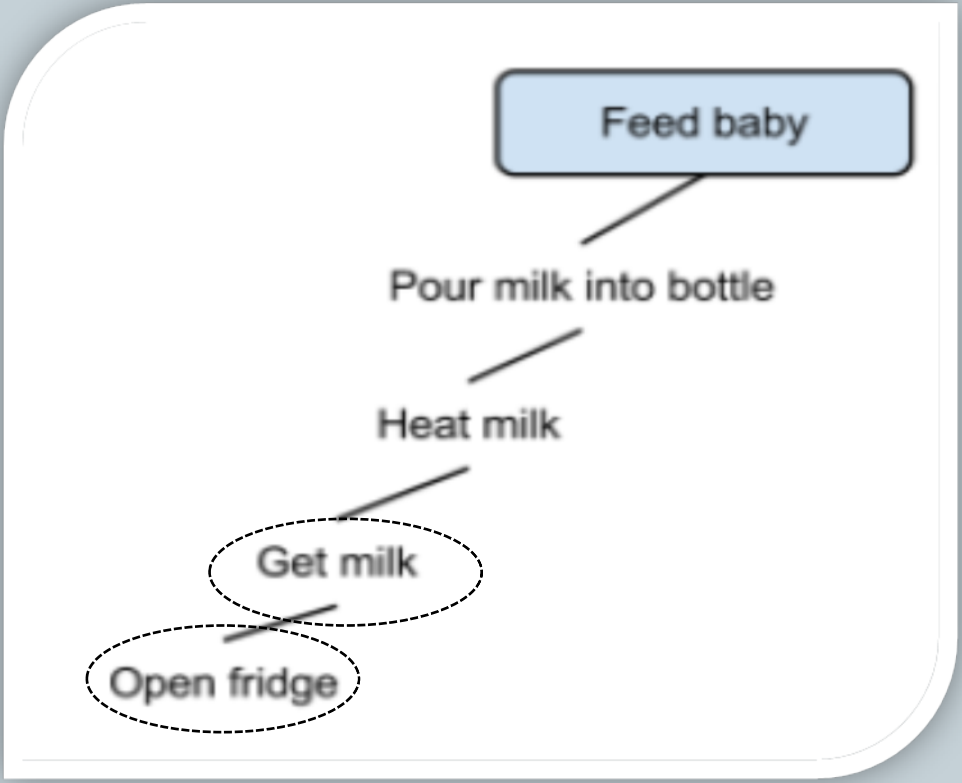
IIR example (2/3)

- Open fridge is observed.



IIR example (3/3)

- Get milk is observed.
- Increase probability for achieving the intention of feeding the baby.



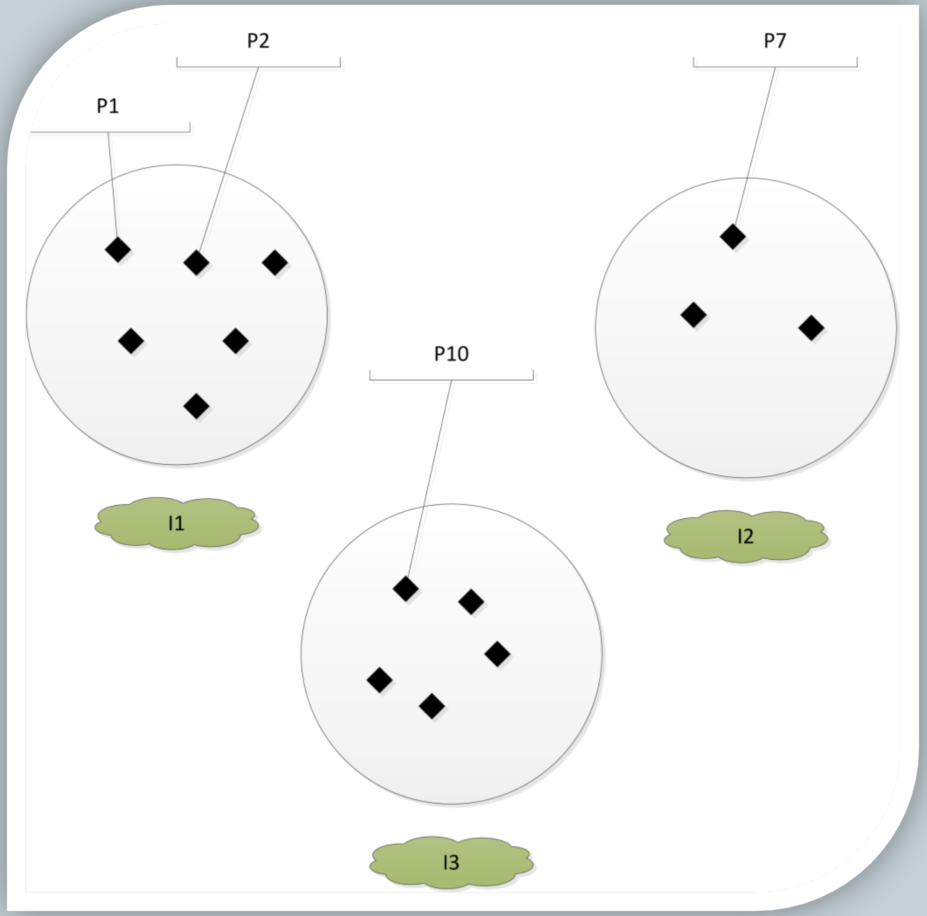
Methods for IIR

Method 1 (H1)

I1 50%
I2 25%
I3 25%

Method 2 (H2)

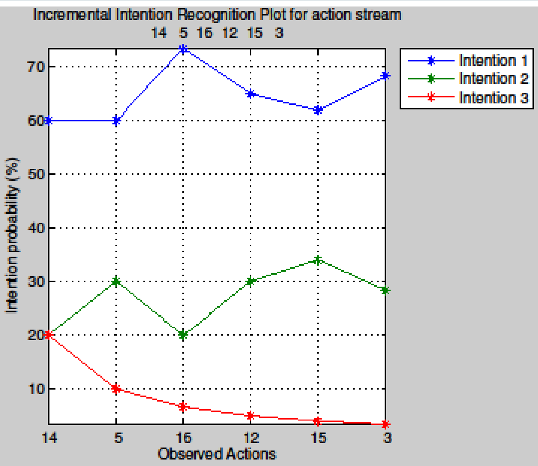
I1 38.45%
I2 38.45%
I3 23.10%



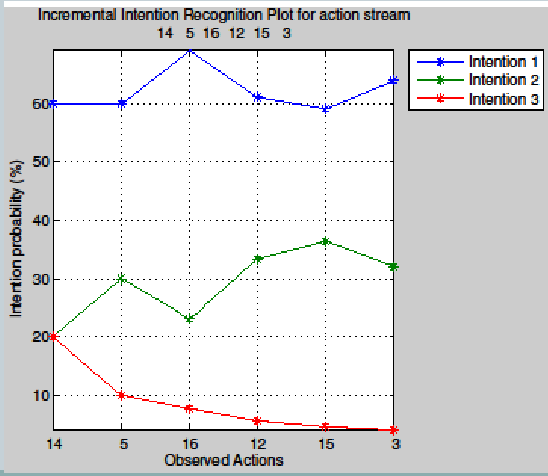
IIR Example (1/2)

I1			I2			I3		
P1	P2	P3	P4	P5	P6	P7	P8	P9
14	14	14	2	2	2	13	13	13
5	5	5	1	1	1	9	9	2
14	14	14	4	4	2	2	2	2
12	12	2	5	14	5	11	14	11
15	4	15	15	4	15	11	4	11
3	3	3	12	12	12	10	10	10

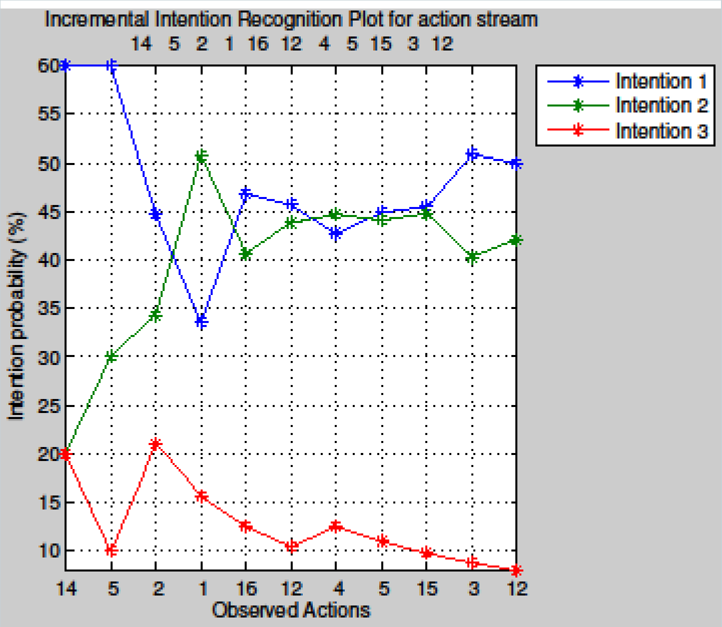
H1



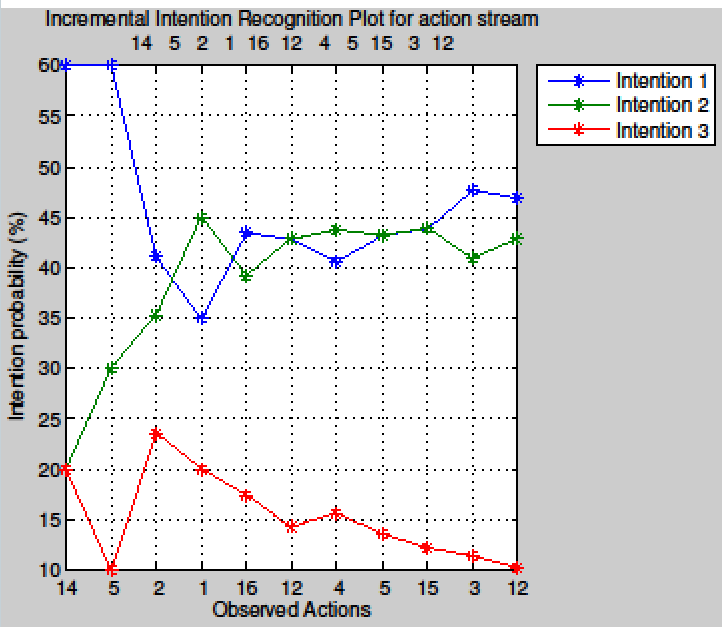
H2



IIR Example (2/2)



H1

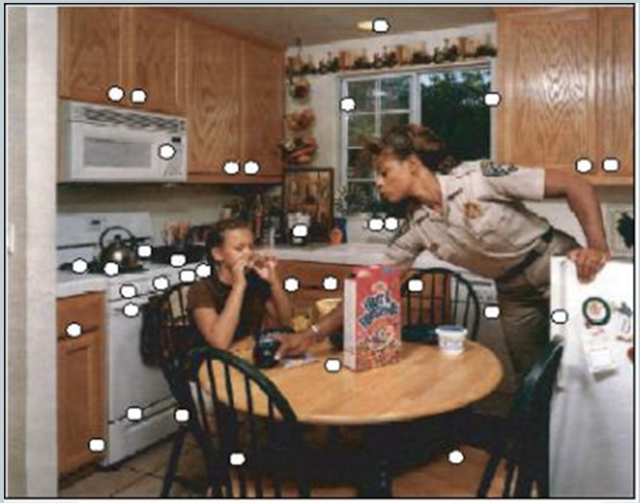


H2

Experiments with Real Datasets

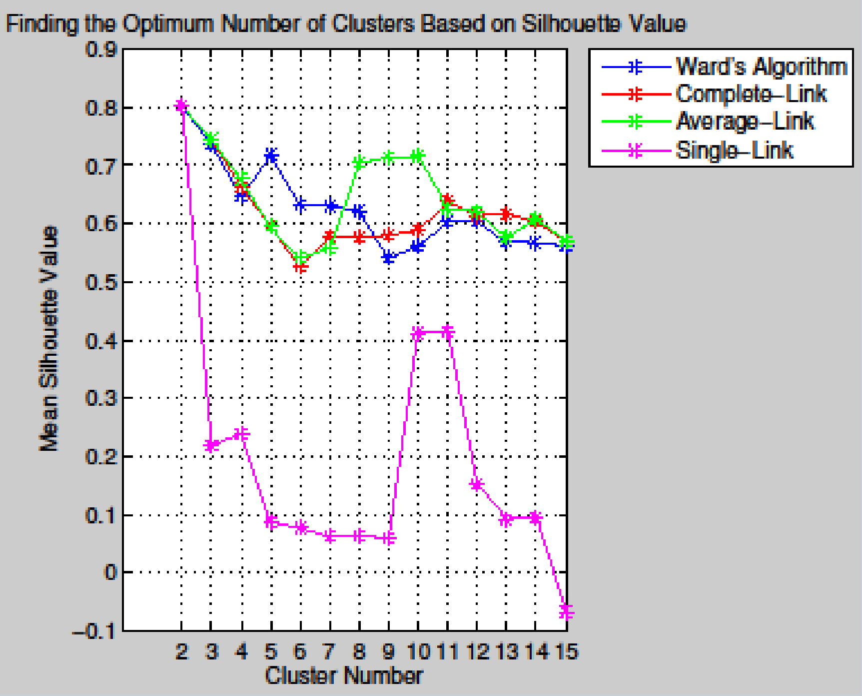
MIT Activity Dataset

- MIT Activity dataset
  - Preparing lunch
  - Toileting
  - Preparing breakfast
  - Bathing
  - Dressing
  - Grooming
  - Preparing beverage
  - Doing Laundry

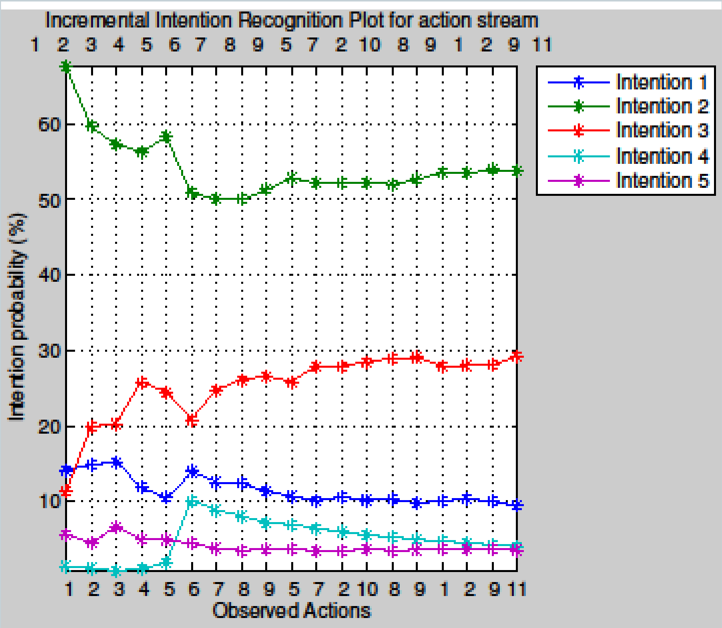


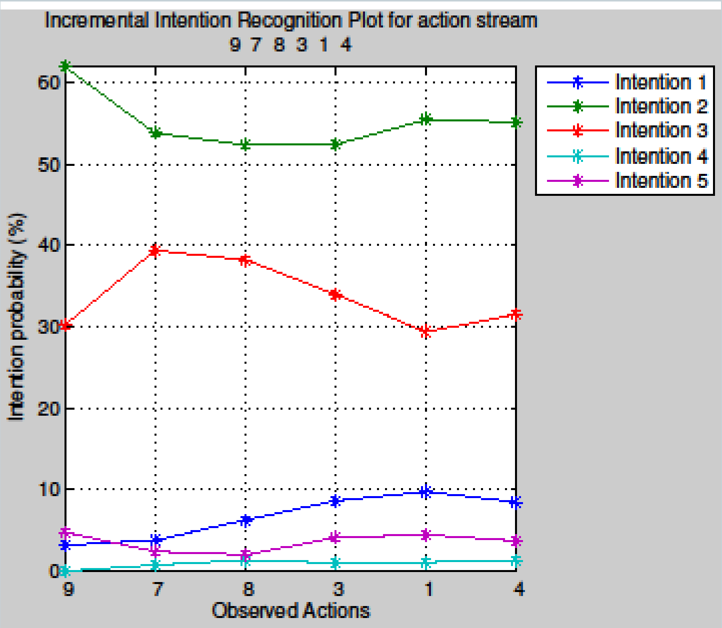
Finding the Number of Clusters

- How many clusters?
  - Naively say 8, but how do we differentiate Preparing lunch from preparing breakfast? (since we do not consider time)



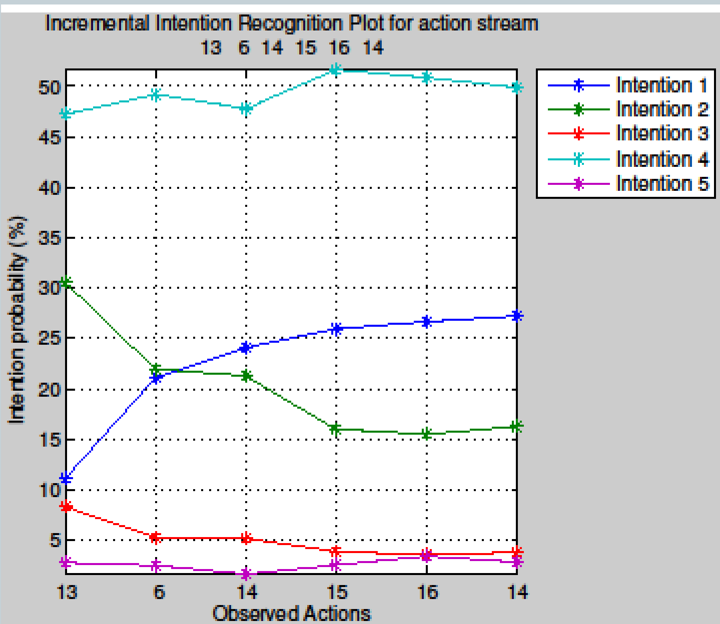
IIR on MIT Activity Dataset (1/4)

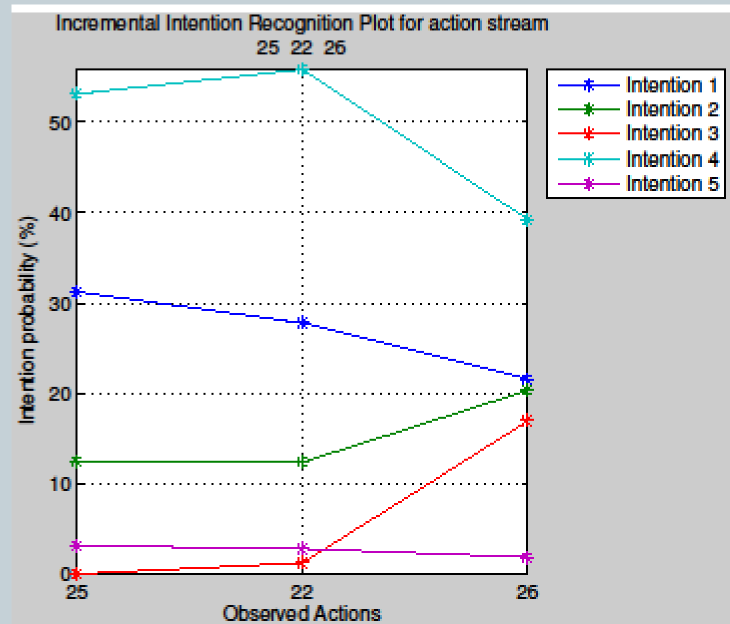




Actions from toileting and bathing used

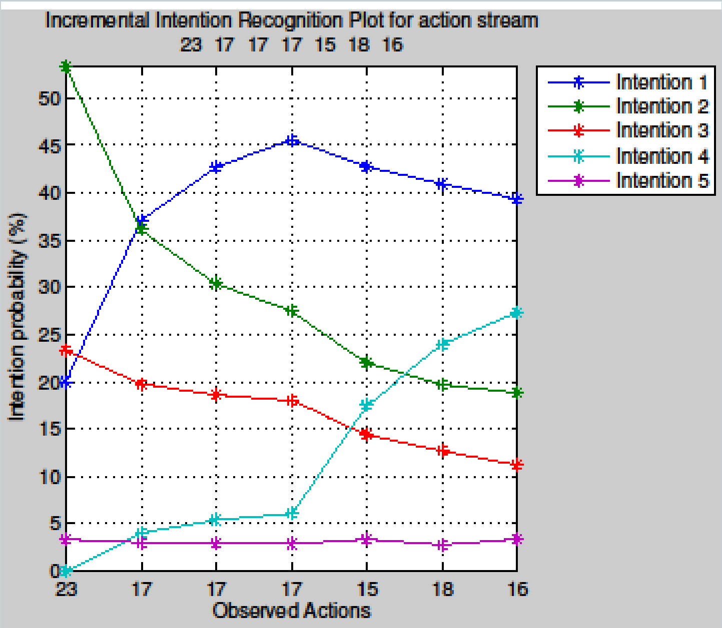
IIR on MIT Activity Dataset(2/4)

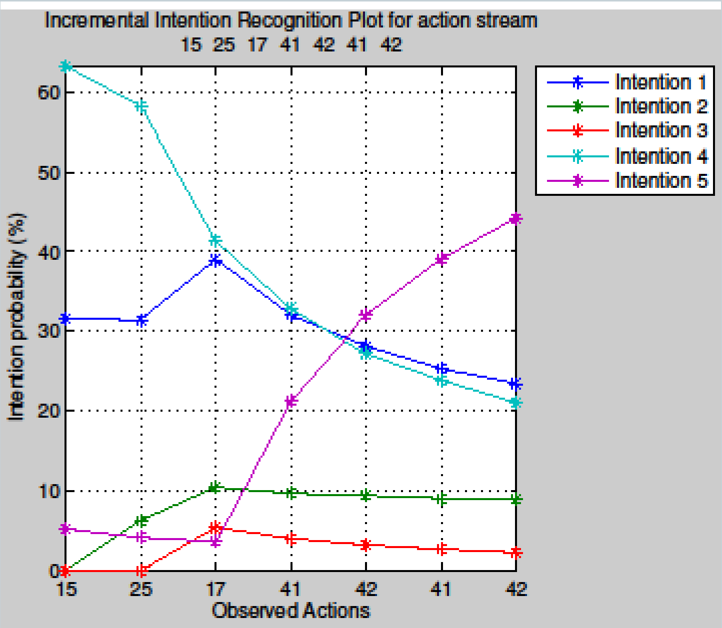




Actions from preparing breakfast and preparing beverage used

IIR on MIT Activity Dataset (3/4)

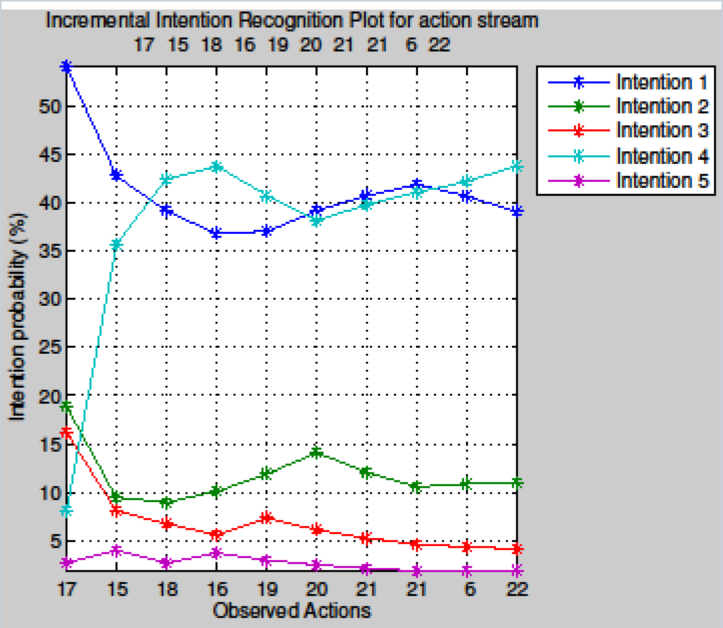




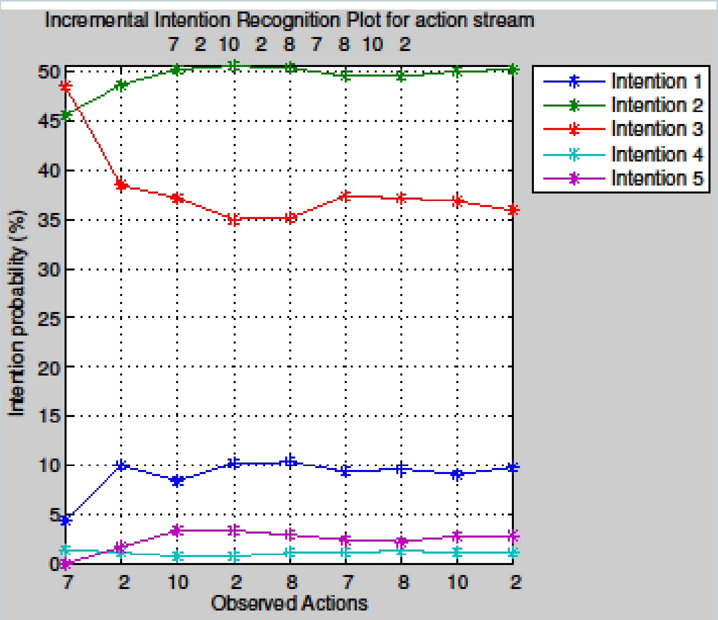
Dressing

Doing the laundry

IIR on MIT Activity Dataset (4/4)



Preparing Breakfast

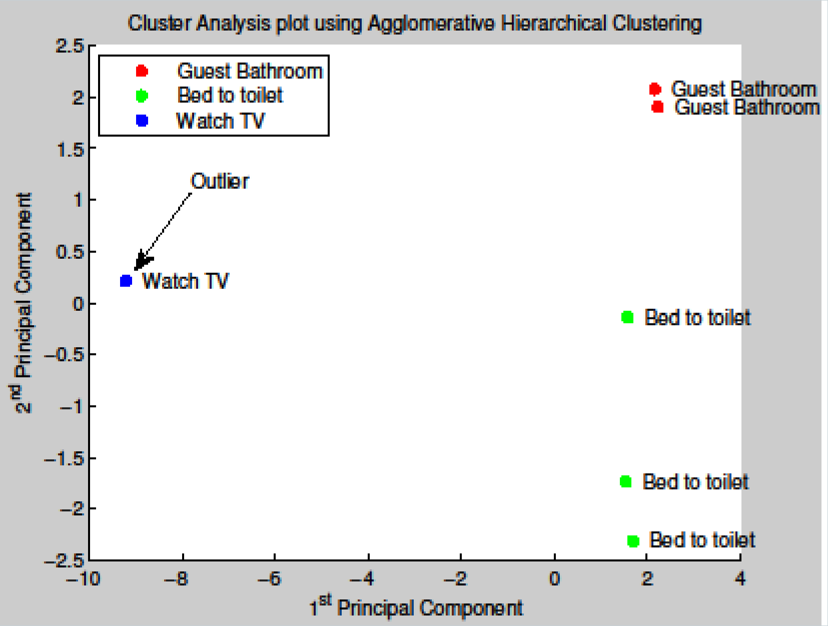


Grooming

Results for MIT Activity Dataset

- We are able to recognise broad categories of activities with over than 40% accuracy.
- Tapia et al. recognise activities from 25%-89%.

CASAS dataset



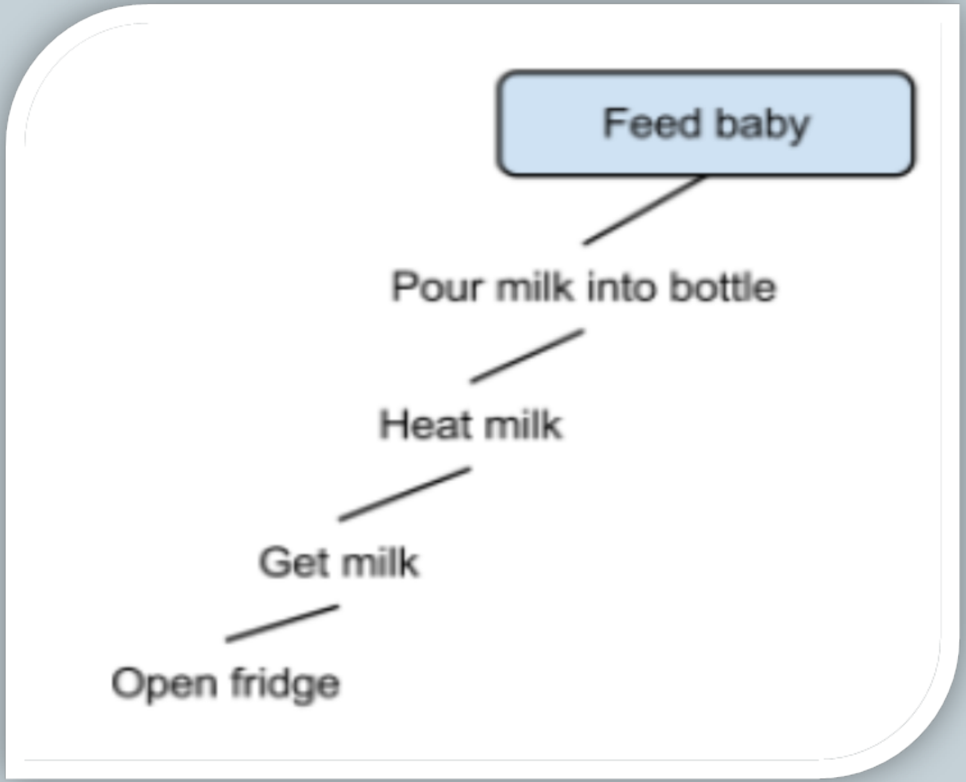
Data contained a lot of noise.
Sensor data do not always represent actions.

Post-Processing

Temporal Constrains and Sensor Accuracy

- Open fridge->
  Get milk->
  Heat milk->
  Pour milk into bottle->
  Feed the baby

- Reliability of observing Open fridge

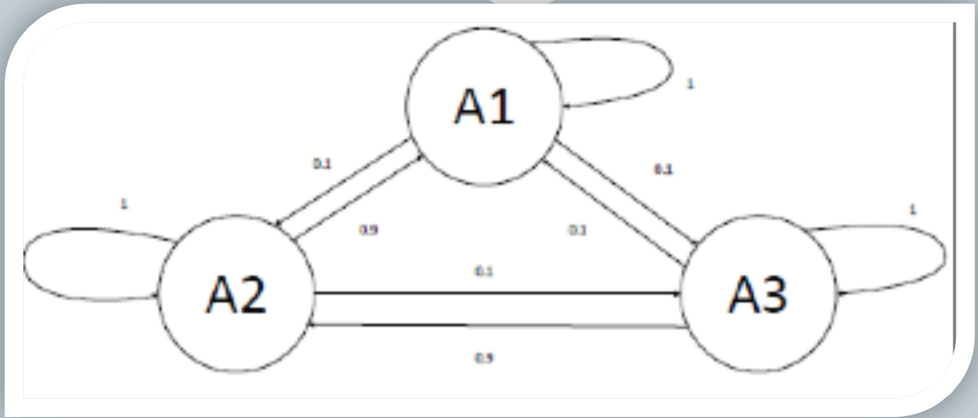


Temporal Constrains

- Incorporate temporal constrains.
  - For example, if an agent intends to have breakfast, in order to pour the milk in the glass, he has to open the fridge and then get the milk.

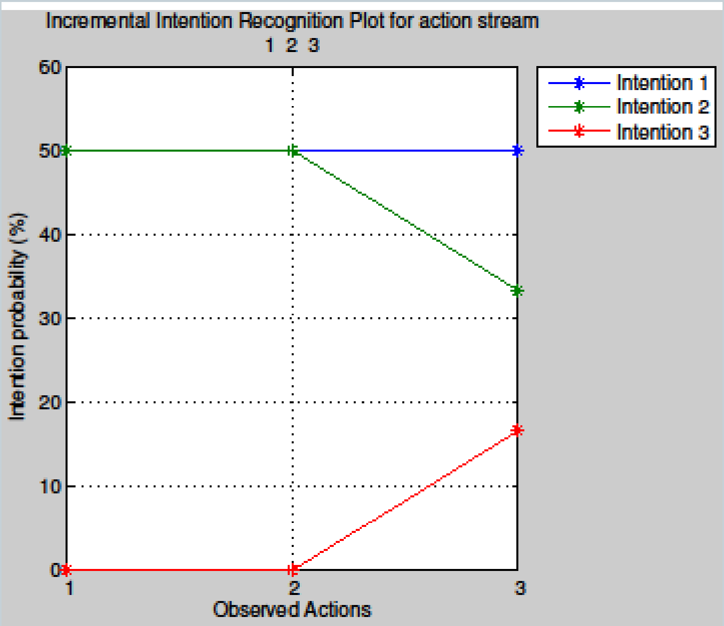
	A1	A2	A3	A4	A5	A6	A7	I
P1	1	1	1	0	0	0	0	I1
P2	1	1	0	0	0	0	1	I2
P3	0	0	1	1	1	0	0	I3

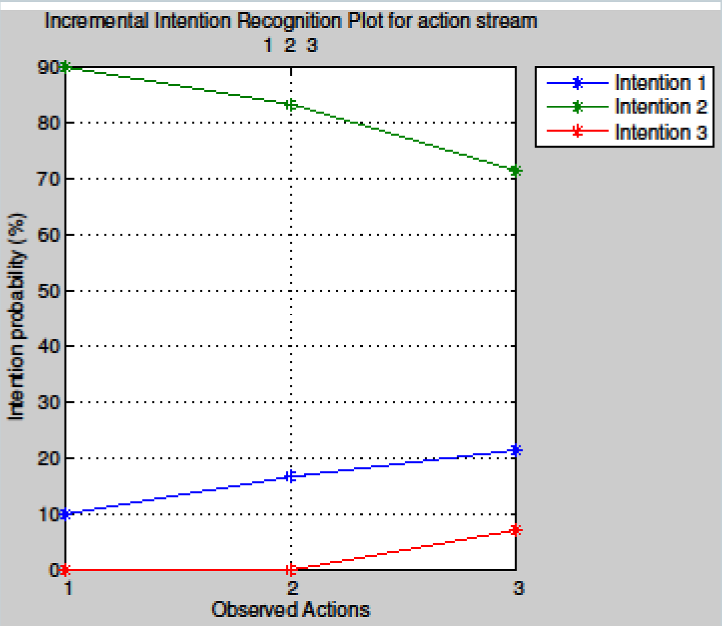
Model for Temporal Constrains



- A2 is most likely to precede A1, and A3 is most likely to precede A2. A1, A2 and A3 can be repeated in a plan as many times without any cost at all as the transition weight to themselves is one. We can represent the constraint as follows: A3 < A2 < A1.

Temporal Constrains Results



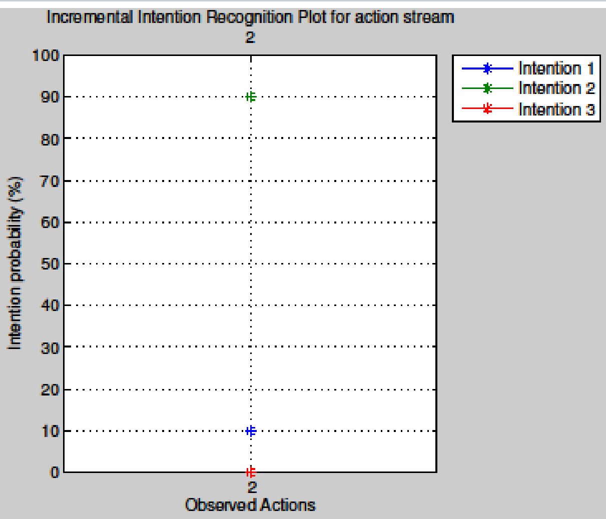


P1:A3 < A2 < A1
P2:A1 < A2 < A7 or A1 < A7
P3: A3 < A4 < A5 or A3 < A5

Sensor Reliability

- Add sensor reliability for recognising actions.
  - Real life sensors might not be reliable and recognise actions that have not happen or vice versa.

Action	Accuracy
A1	0.1



A2 belongs in P1 and P2. A1 might have happened before and thus, I2 is more likely to be the intended goal (temporal constrains are also applied).

Conclusion

Conclusions

- Unsupervised learning is helpful in IR
- No silver bullet exists (techniques are data dependent)
- Real datasets had low level data. The system was designed for more abstract data (actions not sensor firings).
- More work is needed in post-processing.
- More real data should be tested.

Q&A

Questions?