CHAPTER XXX

A CLUSTERING APPROACH TO INTENTION RECOGNITION

Fariba Sadri, Weikun Wang, Afroditi Xafi

Department of Computing, Imperial College London
180 Queens Gate, London SW7 2AZ, UK
E-mails: {fs, ww2210, ax10}@doc.ic.ac.uk

Intention recognition has significant applications in ambient intelligence, assisted living and care of the elderly, games and intrusion and other crime detection. In this chapter we explore an approach to intention recognition based on clustering. To this end we show how to map the intention recognition problem into a clustering problem. We then use three different clustering algorithms, Fuzzy C-means, Possibilistic C-means and Improved Possibilistic C-means. We illustrate and compare their effectiveness empirically using a variety of test cases, including cases involving noisy or partial data. To our knowledge the use of clustering techniques for intention recognition is novel, and this chapter suggests it is promising.

1. Introduction

Intention recognition (IR) is the problem of recognising the intentions[1] of an agent by (incrementally) observing its actions. Plan recognition goes further than intention recognition, and additionally attempts to recognise the plan (sequence of actions, including some not yet observed) the observed agent is pursuing. Many applications of intention recognition have been explored, including Unix-based help facilities and story understanding, in its earlier years, and ambient intelligence, elder care, e.g.[20,11,21], computer games, e.g.[5], prediction of military maneuvers, e.g.[19], and criminal intent detection, e.g.[10,14], more recently.

Ambient intelligence (AMI) environments must be capable of anticipating the needs, desires and behaviour of their inhabitants[1] in order to provide suitable support to the inhabitants. Intention recognition can make a significant contribution to AMI systems by enabling and enriching their anticipatory capabilities.

Various techniques have been used for intention recognition. The most common are logic-based[2,8,23], case-based[7] and probabilistic approaches[4,20,11].

[1] In this chapter we use intention and goal synonymously.

In this chapter we explore the use of clustering techniques for intention recognition. Clustering or cluster analysis is the task of classifying objects into groups in such a way that the objects in each group are more "similar" to one another than to objects outside the group. Clustering is more commonly applied to pattern recognition, image analysis, information retrieval, and bioinformatics. To our knowledge the application of clustering to intention recognition is novel.

In order to apply cluster analysis, the intention recognition problem has to be crafted as a clustering problem. Intuitively the fundamental functionality of an IR system is to classify observed actions into intentions (and plans to achieve intentions). Thus actions "related" to one another, according to some suitable criteria, have to be grouped within clusters identifying potential intentions.

In order to map intention recognition to a clustering problem, we have to overcome several difficulties. For example clustering is usually applied to elements modeled in Euclidean spaces. Thus we must map plans and actions to a format suitable for clustering, and we must do so in a robust fashion that can deal with noisy and partial data. To this end we need to devise a measure of "relatedness" or "similarity" between actions, and we need to devise a way of interpreting the result of the clustering, to associate an intention with each cluster, and a ranking with each intention indicating its likelihood, given some observed actions.

In the following sections, after presenting separate backgrounds for intention recognition and clustering, we discuss how we can overcome the difficulties mentioned above, to build a bridge between the two fields of intention recognition and machine learning via clustering, with promising results. We show how three clustering algorithms, Fuzzy C-Means, Possibilistic C-means and Improved Possibilistic C-means, can be applied to intention recognition, and we compare them empirically.

2. Background

2.1. *Intention Recognition*

The input to an intention recognition system usually consists of a sequence of observed actions (actions executed by an agent whose intention is being determined), and either a plan library, providing plans for intentions, or an action theory describing the semantics of actions in terms of their pre- and post-conditions. The task of the intention recognition system then is to determine the most likely goal(s) the observed agent is trying to achieve by the actions that have been observed so far and others most likely yet to be executed. This is summarised in figure 1.

Cohen, et al.[6] classify intention recognition as either *intended* or *keyhole*. In the former the actor wants his intentions to be identified and intentionally gives signals to be sensed by other (observing) agents. In the latter the actor 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. This latter will be the most common case in AMI scenarios, for example in the home environment.

Intention recognition has been an active area of research for many years, and several approaches and applications have been proposed. For example, Demolombe and Fernandez[8] use logic-based specifications of macro-actions written in Golog[18], Sadri[23] and Hong[13] map reasoning about intentions with logic-based theories of causality into problems of graph generation and path finding, Geib and Goldman[11] use probabilistic techniques and plan libraries specified as Hierarchical Task Networks (HTNs), and Geib and Steedman[12] cast intention recognition as a parsing problem. They map Hierarchical Task Networks into context-free grammars, and use parsing techniques to group together individual observations into structures that are meaningful according to the grammars. A survey of the logic-based approaches can be found in[22,24].

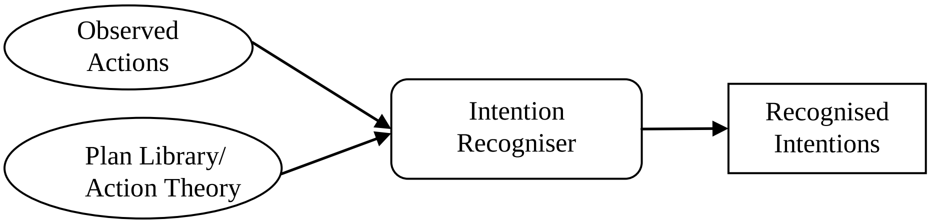

Fig. 1. Intention recognition

2.2. *Clustering Techniques*

Clustering is an unsupervised learning technique and involves the task of classifying objects into groups in such a way that the objects in each group are more “similar” to one another than to objects outside the group. Clustering involves several steps shown in figure 2. These steps will be elaborated later in the context of intention recognition.

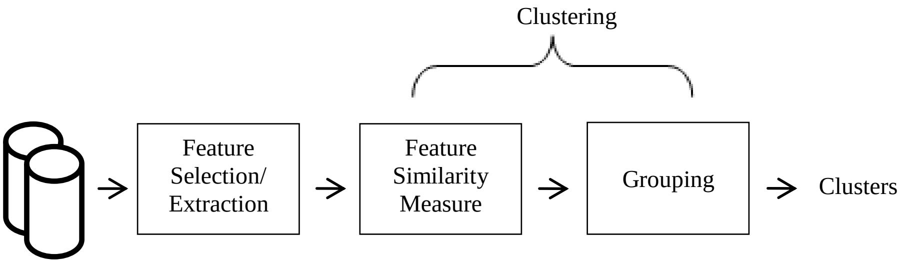

Fig. 2. Clustering procedure

Clustering algorithms may be *exclusive* (or hard), classifying objects into non-overlapping clusters, or *fuzzy* allowing overlapping clusters, where each object belongs to each cluster to a certain degree. They can also be *hierarchical* or *partitional*. Hierarchical approaches proceed successively by either merging smaller clusters into large ones, or by splitting large clusters into smaller ones. The end result of the algorithm is a tree of clusters called a dendrogram, which shows the hierarchical relationship between the clusters. By cutting the dendrogram at a desired level, a clustering of the data items into groups is obtained. Partitional approaches, on the other hand, directly divide the data into a pre-determined number of clusters.

For our work we have chosen the basic C-means clustering algorithm[9] and two of its refinements[17,26]. All three algorithms are fuzzy and partitional. These types of algorithm seem more appropriate for the application of intention recognition than hard or hierarchical types, because of the following reasons. Firstly an action may be part of a plan for achieving more than one intention, thus the suitability of fuzzy techniques. For example *getting milk from the fridge* may be an action in a plan for making tea and a plan for making porridge. Secondly, in common with all other intention recognition algorithms we assume that there is a pre-determined set of possible intentions that the algorithm can recognise, and thus the suitability of partitional clustering techniques.

The similarity measure used for clustering is dependent on the domain of the data and the feature extraction applied. For instance, when data entries are represented as points in a Euclidean space, each dimension represents a feature that has descriptive power, and the Euclidean distance can be used as a way to compare proximity of two points. If the clusters involve a sufficiently small number of dimensions they can be plotted and visualized. For example one may produce a feature space of points representing different water samples across the country. Each dimension can represent the percentage of a particular chemical in the sample. Then one may apply clustering to detect areas that share common water types.

There are cases, however, where the number of dimensions/features can be high. Then dimensionality reduction is attempted by combining or transforming features or by removing features that have less discriminatory power. A number of feature extraction techniques are available, including Principal Component Analysis[15], Isomap[25], and Laplacian Eigenmap[3].

We have chosen the Laplacian Eigenmap technique because it is efficient and popular, and, crucially, it ensures that points close to each other with respect to the chosen similarity measure will be close to each other in the low dimensional space.

3. The Intention Recognition Task

We focus on the task of recognising the intention(s) of a single agent and cover both the intended and keyhole cases. The agent may have multiple intentions and may be interleaving action executions in pursuit of these intentions, and may make mistakes, or the sensor data may be faulty. Moreover, the agent may miss some relevant actions, or the sensors may miss recording them. Thus the data of executed actions may be partial and imperfect. We assume we have a library of plans.

Definition 1: *Plan*
A plan is a (non-empty) sequence of actions and is associated with an intention.
In effect a plan for an intention denotes the sequence of actions the execution of which will achieve the intention. An intention may have more than one plan, an action may occur in none, one or several plans, possibly for different intentions, and an action may be repeated in a plan. Example 1 shows a simple plan library consisting of plans for three intentions.

Example 1:
Intention I1: *Make Tea* Plan 1: 1, 2, 3, 4, 5
Intention I2: *Make Cocoa* Plan 2: 1, 2, 6, 7, 5
Intention I3: *Make Breakfast* Plan 3: 1, 8, 9, 10, 11

where the numbers correspond to actions as follows:

1	2	3	4	5	6	7	8	9	10	11
get milk	get cup	put tea-bag in cup	pour boiled water in cup	add milk to cup	boil milk	put cocoa in cup	get bowl	put cereal in bowl	pour milk in bowl	add sugar to bowl

We observe the actions of an agent.

Definition 2: *Observations, Partial and Noisy Observations*
Observations are sequences of actions (executed by the agent whose intention is being determined). We assume the observed actions are ground (variable-free), and, as in plans, for simplicity, we denote them by numerical identifiers.

Observations can be *partial*, in the sense that we may not observe every action that the agent executes. Observations may be *noisy*, in two different senses. Firstly, due to sensor or action recognition faults, we may observe actions incorrectly. Secondly, the agent, due to forgetfulness or confusion may execute an action by mistake, or may execute an action towards an intention that he later abandons.

Example 2: Given the plans above, sequence S1, below, is a partial sequence of observations, S2 is noisy, and S3 is an interleaved partial sequence that goes towards achieving both intentions 1 and 3 (quite a likely sequence when one is preparing breakfast!).

S1= 1; 6 S2= 1; 2; 12; 3 S3= 1; 2; 8; 3; 9.

In section 4 we report results obtained for noisy and partial observations. We have also obtained similar results for interleaved observations. But, as space is short, we ignore interleaved observations in the remainder of the chapter.

Given a set of intentions I={I_1, I_2, …, I_n}, a library L of plans for these intentions, a sequence of observed actions A= A_1; A_2; …; A_k, the intention recognition task is to identify a subset I' of I, of the *most likely* intentions in I associated with A, according to the library L. As the sequence of observed actions grows the set of most likely intentions may change.

It may help to note that in the special (and easy) case, where we have complete and "perfect" (i.e. not noisy, partial or interleaved) observations A_1; …; A_i; A_{i+1}; …; A_r;… ; A_s; ... ; A_m, then I'={J_1, J_2, …, J_p}, such that A_1; …; A_i is a plan for achieving J_1, A_{i+1}; …; A_r is a plan for achieving J_2, … and A_s; …; A_m is a plan for achieving J_p.

In the next sections we refer to a slightly more elaborate library of plans than in example 1. This library is given in table 1. There are three intentions, each with three plans. The actions are represented by numerical identifiers. Thus, for example, the first plan for Intention 1 is the sequence 11; 5; 11; 9; 12; 3. These numbers are not related to example 1.

Table 1. A Library of plans.

Intention 1			Intention 2			Intention 3		
Plan 1	Plan 2	Plan 3	Plan 4	Plan 5	Plan 6	Plan 7	Plan 8	Plan 9
11	11	11	2	2	2	10	10	10
5	5	5	1	1	1	6	6	2
11	11	11	4	4	2	2	2	2
9	9	2	5	11	5	8	11	8
12	4	12	12	4	12	8	4	8
3	3	3	9	9	9	7	7	7

4. The Intention Recognition Task as a Clustering Problem

To apply clustering to intention recognition we have to follow a number of steps. First we use the information in the plan library to cluster actions that occur in plans. In order to achieve this we have to invent an appropriate similarity metric for actions. The similarity metric is used to provide a pairwise similarity matrix. To this matrix we apply the Laplacian Eigenmap technique, which will then allow us to visualise the resulting clusters and identify their prototypes (centroids). Thus we will obtain a membership matrix giving the likelihood of each intention given an observed action. Finally with each incoming observed action this membership matrix is used to compute the accumulated likelihood of each intention. These steps are summarised in the following diagram.

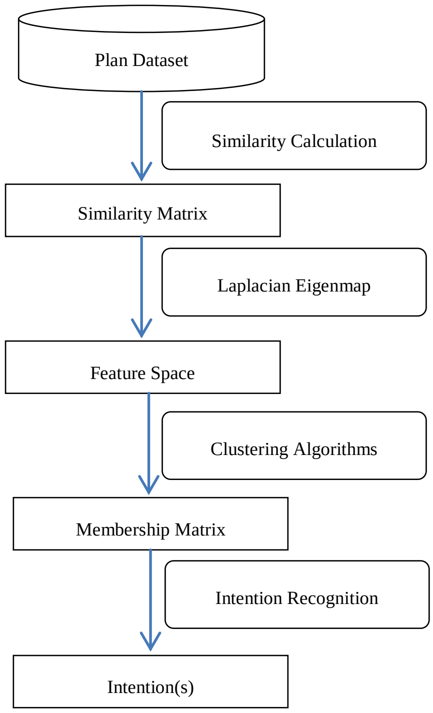

Fig. 3. Flow chart of proposed algorithm

Below we describe the main components of the algorithm.

4.1. *Similarity Calculation for Actions*

Normal similarity metrics, such as Euclidean distance and Mahalanobis distance, are not suitable for intention recognition since we do not have a coordinate system for actions. Instead, we propose a new similarity measure *W(i, j)* between two actions *i* and *j*, as follows:

$$\mathrm{W(i,j)} = \begin{cases} \mathrm{freq(i,j)}\dfrac{|P(i) \cap P(j)|}{|P(i) \cup P(j)|}, & i \neq j \\ 0 & , i = j \end{cases}$$

where *P(i)* denotes the set of plans that include action *i*, and *freq(i, j)* denotes the maximum number of times the two actions *i* and *j* occur together in any plan. The term *freq(i, j)* acts as a weight, so that if a pair of actions occurs many times in a plan, their relationship (similarity) will be stronger. The term $|P(i) \cap P(j)| \ / \ |P(i) \cup P(j)|$ has the effect that a pair of actions is similar if they co-occur in a large number of plans, but not if either of them appears in many plans (if an action is present in many plans, it is considered to be an untypical action). An analogy could be the prominence of words such as "a" and "the" in the English language, and their lack of usefulness when it comes to identifying topics of a document, for example.

Example 3: The similarity between actions 3 and 5 in table 1 is $\mathrm{W}(3{,}5) = 0.6$. This is because the number of plans containing both actions 3 and 5 is |P(3)⋂P(5)|=|Plan1, Plan2, Plan3|=3, the number of plans containing either action is $|P(3) \cup P(5)| = |\mathrm{Plan1, Plan2, Plan3, Plan4, Plan6}| = 5$, and the maximum frequency is $\mathrm{freq}(3, 5) = 1$.

4.2. *Application of Laplacian Eigenmap*

After obtaining the similarity measure between pairs of actions, we use the Laplacian Eigenmap on the *W* matrix to extract useful and typical features from the data. The Laplacian Eigenmap technique is commonly used for clustering, and we omit the details here for lack of space. Suffice it to say that the technique solves the following minimization problem:

$$\underset{f}{\operatorname{argmin}} \frac{1}{2} \sum_{i,j} (f_i - f_j)^2 \, W_{ij} = \underset{f}{\operatorname{argmin}} \, f^T L f$$

$$f^T D f = \mathbf{1}$$

where *D* is a $|W| \times |W|$ diagonal matrix where each element is the summation of the respective column of *W*, $L = D - W$ is the Laplacian matrix and *f* is a mapping from original space *W* to a new space which minimises this equation. This optimisation problem is equal to solving the generalized eigenvalue problem $Lf = \lambda Df$, where λ is the eigenvalue.

These considerations can be related to the problem of intention recognition as follows. For a large value of similarity *W(i, j)*, the mapping aims to minimise the distance between *i* and *j* in the new space, which means actions *i* and *j* should be close in the new space. On the other hand, a small value of similarity *W(i, j)* will incur a heavy penalty in the objective function, resulting in the two points being far from one another.

Example 4: Table 2 shows Eigenvalues 1, 2, 3 for actions 1-5, related to the plans in table 1 and the similarity metric of section 4.1.

Table 2. Laplacian Eigenmap.

Action	1	2	3
1	0.1690	0.1355	0.3829
2	0.1690	-0.0385	0.1308
3	0.1690	0.1667	-0.3825
4	0.1690	0.0625	0.1223
5	0.1690	0.1637	-0.0925

4.3. *Clustering: Fuzzy C-Means (FCM)*

FCM[9] is based on the minimization of an objective function defined as:

$$\mathrm{J}_{FCM}(\mathrm{X};\mathrm{U},\mathrm{V}) = \sum_{i=1}^{c}\sum_{j=1}^{N}\left(u_{ij}\right)^{m} D_{ij}$$

where N is the number of data points in the dataset, c is the pre-determined number of clusters, X= $[\mathrm{x}_1, \dots, x_n]$ is the dataset matrix, U = $[u_{ij}]_{c*N}$ is the fuzzy membership matrix, $u_{ij}\in[0,1]$ is the membership degree of the j-th data in the i-th cluster, $\sum_{i=1}^{c} u_{ij} = 1$, V = $[\mathrm{v}_1, \dots, \mathrm{v}_c]$ is the cluster prototype (centre) matrix, m $\in (1, \infty)$ is the weighting exponent (fuzzy index) which determines the fuzziness of the clusters and is usually set to 2, D_{ij} is the distance measure between data x_j and cluster prototype v_i. Typically, an $L2$ norm distance $\mathrm{D}_{ijA} = \left\|x_j - v_i\right\|_A^2 = \left(x_j - v_i\right)^T A(x_j - v_i)$ is used, where A is the norm-inducing matrix, usually taken to be the identity matrix.

Statistically, the objective function can be seen as a measure of the total variance of X_j from V_i. The minimization could be solved by using a variety of methods for nonlinear optimization problems.

The application to intention recognition produces clusters corresponding to the intentions in the plan library, one cluster for each intention. Figure 4 is based on the plan library of table 1. It shows (fuzzy) clusters and their prototypes resulting from the application of FCM and Laplacian Eigenmap visualization using eigenvectors 2 and 3 of table 2 (extended for all the actions). The bottom right (blue) cluster corresponds to intention 1, the top right (red) cluster corresponds to intention 2, and the left (yellow) cluster corresponds to intention 3. The cluster prototypes are denoted by hollow circles.

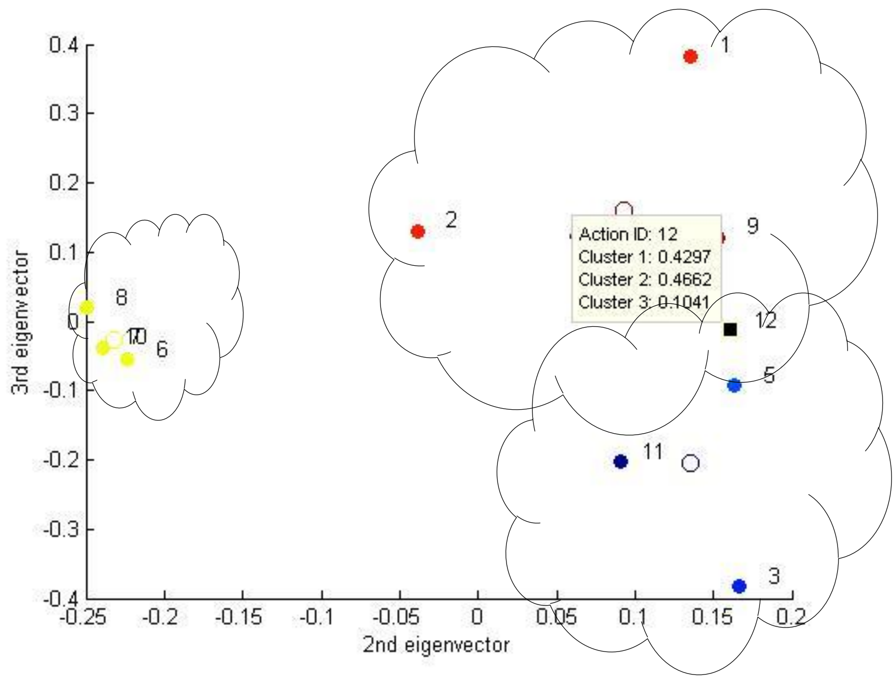

Fig. 4. Laplacian Eigenmap visualization using two eigenvectors

4.4. *Intention Recognition and Membership Matrix*

The iterative clustering algorithm, illustrated by fuzzy c-means above, provides not only the clusters, but also a membership matrix showing the probability of the membership of each action in each cluster. Table 3 shows the membership matrix based on the working example of table 1 and the clusters in figure 4. The membership matrix is then used to accumulate scores for the intentions as actions are observed.

Given a sequence of actions, we simply sum up the membership values of these actions for each intention. The intentions with the highest scores are the most likely intentions. Figure 5 shows how the scores of the intentions changes as more actions are observed. The lines in the graph from top to bottom correspond to intentions 1, 2 and 3, respectively.

Table 3. Membership matrix.

Action → / Intention ↓	1	2	3	4	5	6	7	8	9	10	11	12
1	0.1135	0.0888	0.8185	0.0194	0.7797	0.0062	0.0012	0.0115	0.0446	0.0012	0.9710	0.4288
2	0.7563	0.7054	0.0887	0.9605	0.1545	0.0064	0.0014	0.0168	0.9273	0.0014	0.0147	0.4670
3	0.1302	0.2058	0.0928	0.0201	0.0657	0.9874	0.9974	0.9717	0.0281	0.9974	0.0143	0.1042

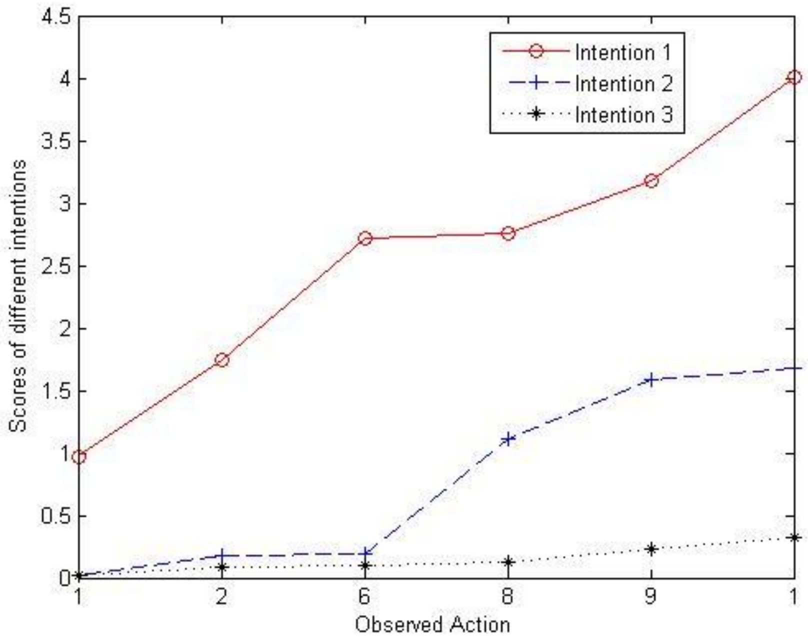

Fig. 5. Incremental intention recognition

4.5. *Other Clustering Algorithms*

4.5.1. *Possibilistic C-means (PCM)*

A problem with FCM is that noise points usually lie far but equidistant from the cluster prototypes and are given equal membership values for all clusters. But such points should be given very low (even zero) value in each cluster. The Possibilistic C-Means[17] has been designed to overcome this problem. PCM relaxes the fuzzy membership matrix constraint $\sum_{i=1}^{c} u_{ij} = 1$ to obtain a "possibilistic" membership constraint, $0 < \sum_{i=1}^{c} u_{ij} < c$. The objective function for PCM is defined as:

$$\mathrm{J}_{PCM}(\mathrm{X};\mathrm{U},\mathrm{V}) = \sum_{i=1}^{c}\sum_{j=1}^{N}\left(u_{ij}\right)^{m} D_{ij} + \sum_{i=1}^{c}\eta_i \sum_{j=1}^{N}\left(1-u_{ij}\right)^{m},$$

where

$$\eta_i = K\frac{\sum_{j=1}^{N}\left(u_{ij}\right)^{m} D_{ij}}{\sum_{j=1}^{N}\left(u_{ij}\right)^{m}}, 1 \leq \mathrm{i} \leq \mathrm{c}$$

is the scale parameter at the *i*-th cluster. *K* is typically chosen to be 1. The first term of the objective function demands that the distances from data to the cluster prototype be as low as possible, whereas the second term forces the u_{ij} to be as large as possible to avoid trivial solutions. The value of η_i determines when the membership value of a point in a cluster becomes 0.5.

Table 4 illustrates how PCM can give a different result compared to FCM, assuming a dataset with two noise points, A and B, and two clusters, where FCM would give values of 0.5 for the membership of each noise point in each cluster, and PCM can be more discriminating.

Table 4. Membership value of point A and B from FCM and PCM.

Membership value	FCM		PCM	
	Cluster 1	Cluster 2	Cluster 1	Cluster 2
Point A	0.5	0.5	0.1363	0.1363
Point B	0.5	0.5	0.0586	0.0586

4.5.2. *Improved Possibilistic C-means (IPCM)*

Although PCM improves on FCM it can cause coincident clusters, i.e. two or more cluster prototypes can settle at the same position. In order to solve this problem Zhang and Leung[26] proposed an improved PCM algorithm which integrates FCM into the objective function. This combination can determine proper clusters as well as achieve robustness against noisy data. The improved PCM algorithm is derived directly from the possibilistic approach. The objective function of IPCM is defined as:

$$\mathrm{J}_{IPCM}(\mathrm{X};\mathrm{U}^{(p)},\mathrm{U}^{(f)},\mathrm{V}) = \sum_{i=1}^{c}\sum_{j=1}^{N}\left(u_{ij}^{(f)}\right)^{m_f}\left(\left(u_{ij}^{(p)}\right)^{m_p} D_{ij} + \eta_i\left(1-u_{ij}^{(p)}\right)^{m_p}\right)$$

For further details we refer the reader to[26].

5. Empirical Results

5.1. *Test Data*

Two inputs are required for the intention recognition algorithm, namely the plan library and a sequence of observed actions to be classified. Regarding the plan library, many parameters can be varied. We vary two, *plan diversification (PD)*, that is how similar the plans for each intention are to each other, and *intention relatedness (InR)*, that is how similar the plans for an intention are to plans for other intentions. PD ranges from 0 to 1, such that for example, if it is 0.1 and the plan size is 100, then any two plans aiming for the same intention differ in 10 actions. For InR, the plans for different intentions consist of actions randomly chosen from an action set according to the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. For example, if the actions set has 100 actions and the three Gaussian distributions are $\mathcal{N}(25,5)$, $\mathcal{N}(50,5)$, $\mathcal{N}(75,5)$, the generated plans may have most of the actions around action 25, 50 and 75. The variance σ^2 determines the relatedness between different intentions.

Regarding the observed actions again several parameters can be varied. We vary two, the degrees of noise and partiality. We vary the noise parameter from 0 to 1, where 0 means there is no noise in the observed actions, while 1 means the whole sequence is randomly generated. Similarly, we vary the partiality parameter from 0 to 1, corresponding to the ratio of missing actions in the sequence.

We use an action set with 500 different actions, 3 intentions, each with 3 plans, and each plan having 150 actions. For intention relatedness we use three Gaussian distributions $\mathcal{N}(125,75)$, $\mathcal{N}(250,75)$, $\mathcal{N}(375,75)$. For each intention, its plans are formed from actions randomly picked from the action set according to the distribution of the intention.

5.2. *Experiments*

5.2.1. *Effectiveness of all Three Clustering Algorithms*

We use 9 sequences of observed actions, OA1-OA9, such that OA1-OA3 are predominantly related to intention 1 (I1), according to different plans of I1, OA4-OA6 to intention 2 (I2) and OA7-OA9 to intention 3 (I3). Table 5 shows the result for the most basic case where OA1-OA9 are non-noisy and non-partial. The plan diversification is 0.5. The figures in the table show the likelihood of each intention given the observation (likelihoods multiplied by 10 for easier readability). As shown the results are good. Similar results are obtained with noisy and partial observations for all three algorithms for all plan diversifications we tried (0.2, 0.5, 0.8). Some of these additional results are further illustrated in the next section.

Table 5. Performance of the three clustering algorithms with non-noisy and non-partial observations, with plan diversification 0.5.

		OA1	OA2	OA3	OA4	OA5	OA6	OA7	OA8	OA9
FCM	I1	118.34	102.93	105.34	18.75	21.48	22.66	12.30	24.50	19.55
	I2	18.68	26.78	22.06	110.24	105.06	98.41	19.55	17.26	24.65
	I3	12.98	20.29	22.60	21.01	23.46	28.93	118.14	108.24	105.80
Likeliest		I1	I1	I1	I2	I2	I2	I3	I3	I3
PCM	I1	96.12	86.72	89.57	18.94	20.40	3.45	15.78	24.02	20.52
	I2	20.53	25.50	23.44	87.88	82.09	96.64	21.19	19.60	24.39
	I3	16.91	21.22	24.02	20.56	22.29	49.91	96.39	88.85	89.33
Likeliest		I1	I1	I1	I2	I2	I2	I3	I3	I3
IPCM	I1	42.44	44.31	42.69	3.18	3.63	4.30	2.35	5.24	4.00
	I2	4.88	6.63	6.25	44.72	39.22	45.71	4.97	4.39	6.21
	I3	4.01	5.74	7.89	5.38	6.55	8.77	52.83	47.91	51.73
Likeliest		I1	I1	I1	I2	I2	I2	I3	I3	I3

5.2.2. *Comparison of the Three Clustering Algorithms*

To see how performances vary according to the degrees of noise, partialness and plan diversification we define a score *r* as the ratio of the score of the dominant intention (the one the algorithm assigns the highest value to) to the sum of the scores of all the intentions:

$$\mathrm{r} = \frac{Score_{dom\mathrm{Intention}}}{\sum_{|I|} Score}$$

Figure 6 shows the relative performance of the three clustering algorithms under varying plan diversification, degrees of noise and degrees of partialness of observations. The bars with vertical stripes correspond to FCM, the bars with horizontal bars correspond to PCM, and the bars with diagonal lines correspond to IPCM.

From figure 6 we can see that overall IPCM has the best performance in all cases. With the increase of the diversification, generally the accuracy of PCM decreases. We believe this is because in diversified plans the cluster prototypes tend to move together in PCM.

With increasing degrees of noise the performance of all the algorithms declines somewhat, as one may expect. For a less diversified plan library, PCM performs slightly better than FCM. All three algorithms perform better in the presence of partial observations than in the presence of noise. We conjecture that this is due to the fact that, depending on the levels of plan diversification and intention relatedness, even with partial observations we have a chance of seeing "typical" actions, which accumulatively help the algorithms to guess the correct intention.

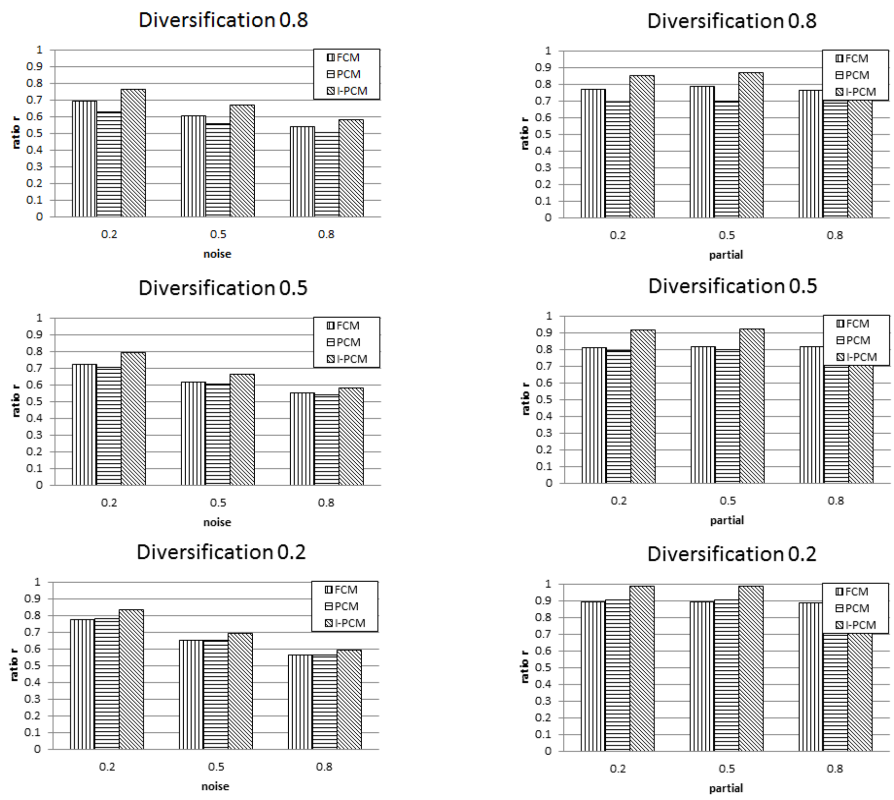

Fig. 6. Performance comparison of the three clustering algorithms

Table 6, below, summarises our conclusions.

Table 6. Summary of different clustering algorithms for intention recognition.

Algorithm	Advantage	Disadvantage
FCM	Robust in noise-free environment	Sensitive to noise; Sensitive to initialization*
PCM	Able to cluster noisy data samples	Coincident cluster prototypes may occur; Sensitive to initialization
IPCM	Robust to noisy and partial observation	Sensitive to initialization

*Sensitive to initialization means that given random initialization of the cluster prototypes, the algorithm may easily get into local optima. It is better to initialize the algorithm based on any pre-knowledge of the positions of the prototypes.

6. Conclusion and Future Work

We have explored the application of clustering techniques to the task of intention recognition, and have found the approach promising. We have also explored the suitability of three clustering algorithms, and found one, IPCM, the best fit for the task.

There is much more that can be explored in bridging the two fields of clustering and intention recognition. Other ways of computing similarity between actions can be investigated. There are several possibilities, for example assigning similarity in terms of resources the actions use, or the locations of actions, or their semantics via an action or causal theory, such as the event calculus[16].

Furthermore, the work reported in this chapter does not take into account the order of observed actions. However, such ordering information is useful in recognizing intentions. We have done some preliminary work in post-processing the results obtained from clustering to modify the likelihood of intentions according to the order of observed actions and other contextual constraints, such as the time of day, the capabilities and habits of the observed agent and so on. There is much more that needs to be done.

Finally, and crucially, more systematic testing for scalability, and testing with more realistic and meaningful data sets are necessary to evaluate the applicability of the clustering techniques further.

Acknowledgments

We are grateful to the anonymous reviewers for their helpful comments.

References

1. Aarts, E., Ambient intelligence: a multimedia perspective. *IEEE Intelligent Systems 19(1)*, 12 19 (2004).
2. Baier, J. A., On procedure recognition in the Situation Calculus. In Proceedings of the Twelfth International Conference of the Chilean Computer Science Society (SCCC-02), 33-42 (2002).
3. Belkin, M. and Niyogi, P., Laplacian Eigenmapsfor Dimensionality Reductionand Data Representation, *Neural Computation* 15, 1373-1396 (2002).
4. Bui, H. H., A general model for online probabilistic plan recognition. In Proceedings of the International Joint Conference on Artificial Intelligence (2003).

5. Cheng, D. C., Thawonmas, R., Case-based plan recognition for real-time strategy games. In Proceedings of the 5th Game-On International Conference (CGAIDE) 2004, Reading, UK, 36-40, November (2004).
6. Cohen, P.R., Perrault, C.R., Allen, J.F., Beyond question answering. In Strategies for Natural Language Processing, W. Lehnert and M. Ringle (Eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, 245-274 (1981).
7. Cox, M.T., Kerkez, B., Case-Based Plan Recognition with Novel Input. International *Journal of Control and Intelligent Systems* 34(2), 96-104 (2006).
8. Demolombe, R., Fernandez, A.M.O., Intention recognition in the Situation Calculus and probability theory frameworks. In Proceedings of Computational Logic in Multi-agent Systems (CLIMA) (2006).
9. Dunn, J., A fuzzy relative of the ISODATA process and its use in detecting compact well separated clusters, *J. Cybern.* 3(3), 32–57 (1974).
10. Geib, C. W., Goldman, R. P., Plan recognition in intrusion detection systems. In the Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX), June (2001).
11. Geib, C. W., Goldman, R. P., Partial observability and probabilistic plan/goal recognition. In Proceedings of the International Workshop on Modeling Others from Observations (MOO-2005), July (2005).
12. Geib, C. W., Steedman, M., On natural language processing and plan recognition. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1612- 1617 (2003).
13. Hong, J., Goal recognition through goal graph analysis, *Journal of Artificial Intelligence Research* 15, 1-30 (2001).
14. Jarvis, P., Lunt, T., Myers, K., Identifying terrorist activity with AI plan recognition technology. In the Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI 04), AAAI Press (2004).
15. Jolliffe, I. T., Principal Component Analysis. Springer, second ed., October (2002).
16. Kowalski, R. and Sergot, M., A Logic-based Calculus of Events, *New Generation Computing*, 4(1), 67-95 (1986).
17. Krishnapuram, R. and Keller, J., The possibilistic C-means algorithm: Insights and recommendations, *IEEE Transactions on Fuzzy Systems* 4, 385–393 (1996).
18. Levesque, H., Reiter, R., Lesperance, Y., Lin, F., Scherl, R., GOLOG: A Logic Programming Language for Dynamic Domains. *Journal of Logic Programming* 31, 59-84 (1997).
19. Mao, W., Gratch, J., A utility-based approach to intention recognition. AAMAS Workshop on Agent Tracking: Modelling Other Agents from Observations (2004).
20. Pereira, L.M., Anh, H.T., Elder care via intention recognition and evolution prospection, in: S. Abreu, D. Seipel (eds.), Procs. 18th International Conference on Applications of Declarative Programming and Knowledge Management (INAP'09), Évora, Portugal, November (2009).
21. Roy, P., Bouchard B., Bouzouane A, Giroux S., A hybrid plan recognition model for Alzheimer's patients: interleaved-erroneous dilemma. IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 131- 137 (2007).
22. Sadri, F., Logic-based Approaches to Intention Recognition, In Handbook of Research on Ambient Intelligence: Trends and Perspectives, Nak-Young Chong and Fulvio Mastrogiovanni Eds., IGI Global, May 2011, 346-375 (2011).
23. Sadri, F., Intention Recognition with Event Calculus Graphs and Weight of Evidence, Proc. of the 3rd International Conference on Agents and Artificial Intelligence, ed. J. Filipe, Springer January (2011).

24. Sadri, F., Intention Recognition in Agents for Ambient Intelligence: Logic-Based Approaches, to appear, Agent-Based Approaches to Ambient Intelligence, Bosse (ed.), IOS Press (2012).
25. Tenenbaum, J. B., de Silva, V. and Langford, J. C., A global geometric framework for nonlinear dimensionality reduction., *Science* 290, 2319-2323, December (2000).
26. Zhang, J. and Leung, Y., Improved possibilistic C-means clustering algorithms, IEEE *Transactions on Fuzzy Systems* 12(2), 209–217, April (2004).