Machine Learning Final: Learning The Structures and Predicting Behavoir of Potential Function Based Multi-Agent Formations

Doug Beck
Aeronautics and Astronautics
Stanford Univeristy

Tzong-han Lee
Electrical Engineering
(Materials Science By Courtesy)
Stanford Univeristy

Chun-Yeon Lin
Mechanical Engineering
Stanford University

December 10, 2010

Abstract

K eeping formation is a technique used to maintain order in multi-agent dynamical systems. Understanding formations has implications for military applications [1] as well as better understanding group behavior [2,3]. In past work, potential functions involving the distances between agents are used to maintain formation for both centralized and decentralized systems [4,5]. In [6], formations are defined rigorously using sets and the notion of formation rigidity is explained. In [2], Virtual Leaders are established as a means to move formations held together by Artificial Potentials. Artificial Potentials are also used in robotics research, like in [7].

I n this project, we developed a series of tools in order to classify groups and their structures, as well as predict a region which encompasses as many agents in a targeted group as possible, without encompassing any members from untargeted groups. Classifying groups by observing the dynamics of multi-agent groups could help distinguish between friendly and unfriendly forces in a military scenario in which it is difficult to determine the intent of certain agents. In determining the overall structure of a group (the links that define a formation), one understands which agents are maintaining distance with which other agents in order to create global, coordinated behavior the formation is predicated on. It then becomes easier to identify agents pivitol to holding the structure together and destroy it more efficiently. For instance, if the group was a cluster of satellites which were all maintaining position based on a central comm satellite, destroying the comm satellite would render the entire group useless. Finally, better predicting an area in which some targeted group will be after a delay while avoiding untargeted groups is important for military scenarios in which innocent groups are involved. Being able to accurately predict such an area would have tremendous application in today's wars, which are more and more waged amongst civilian parties. The development of an accurate algorithm which can identify such an area with great confidence would help one predict whether to hold off on an area focused attack strategy until there is greater chance of targetted parties being affected without injuring innocent parties.

We use an algorithm highly motivated by k-means in order to distinguish each of the multi-agent groups from other groups that are operating in the same region. We also classify the structure of the multi-agent system by using logistic regression, using forward search in order to determine the best two features to use, where a group's structure is defined by the set of links between its agents. A link implies that two agents are trying to maintain the distance between them. We model this communication as internal linear springs which push the agents apart when the agents are less than some constant chosen distance apart and pull the agents together when they are more than some fixed distance apart. The motion of each agent is damped. It is important to note that similar potential functions are commonly used in order to maintain formation in decentralized systems [2,3,4,6]. Finally, we predict a circular area in which an entire targeted group will be (and which no untargetted agents will be) using an algorithm which relies heavily on linear regression.

1 The Chosen Multi-Agent Model

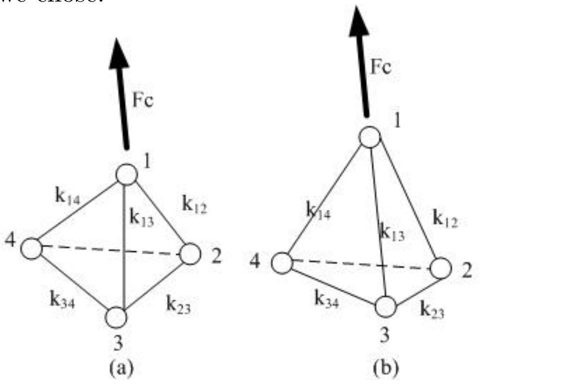
W e created our own simulation specifically for this project using the following model in order to test the algorithms we developed. We draw from a chosen uniform distrubution of the constants the model depends on in order to train and test our methods on multi-agent systems of varying size and performance. We discretized our continuous model by integrating forward in time with Runge-Kutta 4 (RK4).We consider there are m groups, and each group has n agents. It is assumed that each agent only knows the distance between it and other agents it is linked to and that the motion of all agents (except for the leader) is governed by applying a force to themselves proportional in to the difference between the current and desired distance between it and each of the agents it is linked to (like a first order spring relationship with spring $k_{xi,xj}$):

$$k(x_i, x_j) = \begin{cases} k, & \text{if } x_i \text{ and } x_j \text{ have connection} \\ 0, & \text{if } x_i \text{ and } x_j \text{ have no connection} \end{cases}$$

There is one leader in each group, who pulls the rest of the group with a control force:

$$fc(x_i) = \begin{cases} \textit{some} \text{ signal}, & \text{if } x_i \text{ is the leader} \\ 0, & \text{otherwise} \end{cases}$$

The following picture helps illustrate the model we chose:



The state space of each group can be described as follows:

$$\dot{x} = Ax + b$$

$$A_{2+4*(m-1),2+4*(n-1)} = \begin{cases} \sum_{i,j=1}^{\infty} k_{ij} * (1 - b_{ij})/d_{ij} \\ \text{if } i \neq \text{j} \\ 0, \text{ i=j} \end{cases}$$

k_{ij}=the virtual spring rate between agent i and j
b_{ij}=the natural length between agent i and j

$$x = \begin{bmatrix} x1 \\ \dot{x1} \\ y1 \\ \dot{y1} \\ . \\ . \\ xn \\ \dot{xn} \\ yn \\ \dot{yn} \end{bmatrix}, b = \begin{bmatrix} F_c \\ 0 \\ F_c \\ 0 \\ . \\ . \\ 0 \\ 0 \end{bmatrix}$$

It should be noted that this state matrix is equivalent to applying the following force to the agents:

$$\vec{F}_{ij} = \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \left[k_{ij}(1 - d_{ij}/\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}) \right]$$

where $k_{ij} = k_{ji}$ and $d_{ij} = d_{ji}$ by definition.

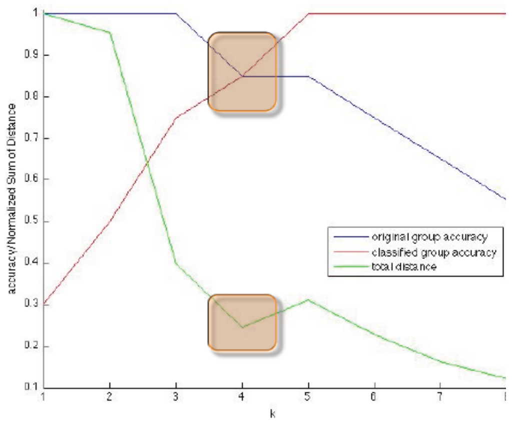
$$\ddot{\vec{x_i}} = \frac{\sum\limits_{j=1}^{n} \vec{F}_{ij}}{m}$$

The fact that d_{ij} is a function of the state, makes this a nonlinear problem. We add damping to the system by subtracting a damping term, $b_i \dot{x_i}$ from the acceleration of agent i, $\ddot{x_i}$. This causes the agents to converge asymptotically to a final state with no velocity or acceleration in the absense of external forces as proven in [4].

2 Group Classification

T he first tool was developed to determine which group each agent corresponded to by observing the position component of each of the agents' state over time alone (wouldn't know the original group association in the field). In order to do this, we used a modified version of k-means with a range of k (the number of cluster centroids) that matched the range of possible multi-agent groups in the set:

```
1. Select a k value
2. Initialize k cluster points randomly
3. REPEAT unitil all cluster centroids w/in $\epsilon$ {
    i. Perform K-Means
    ii. Update the initial cluster points from (i)
    iii. Increment state of all agents by one
    time step
  }
4. Agents associated with the same cluster point
   are said to be in the same group at simulation
   end
```



How k value chosen for group classification

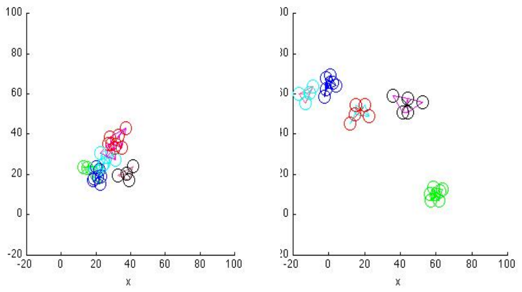
At each simulation time step, the number of time increments that the ith agent was associated with the jth centroid cluster is updated in the matrix, S_{ij}. At the end of the algorithm, each agent is put in a group for which $max_j(S_{ij})$ $\forall$ agents i. The maximum distance between any cluster centroid and any of the agents summed over all the simulation time steps and normalized by the largest value for any k is plotted in green against k in the above figure. We call this "total distance." Also, the original group error and classified group error are plotted against k in blue and red respectively in the above figure.

The original group error is the number of agents across all of the original groups which were associated with the same cluster point that the majority of the agents in their same original group were divided by the total number of agents. This error approaches (and reaches) 1 as k approaches 1 because when all of the agents are associated with the same cluster point, all of the agents in each agent's original group are associated with that same cluster point. The classified group accuracy is the number of agents who are associated with a cluster point which contains more agents in their original group than any other group divided by the total number of agents. This approaches (and reaches) 1 as k approaches the number of total agents because when there are as many cluster points as agents, every agent must be associated with a cluster point which contains more of its group than any other.

While these error metrics were used in order to validate the use of total distance as a way to choose the best k, one would not be able to calculate them without knowing the original group association. The number of cluster centroids is selected at the beginning of each training run and then the k for which the total distance stops improving as dramatically is selected (most likely by a human). This value of k typically corresponds with both high original accuracy and high classified group accuracy (even when the value of k chosen isn't exactly true to the original number of groups being simulated).

One can see from the total distance vs. k curve (above green), that 4 is the value for k after which there are only marginal improvements. This makes sense since we would assume that the agents in a group are clustered together closer than agents from other groups. The rationale is that there shouldn't be very much improvement after ensuring that there are at least as many cluster points as there are relatively tightly bound groups on average over time.

The group of agents on right of the below graph represents a time step at which the agents were all correctly identified and the value of k corresponds exactly to the number of groups being simulated. In our graphical notation, circles represent agents, and same colored lines represent links between agents in the "real-world" model (the algorithm doesn't know which are related in a model sense). Agents with the same color circle have been classified in the same group by the algorithm for the displayed time step.



Simulation of 5 groups of agents classified by k-means algoritm at time=2000 and time=20000

The above left figure demonstrates a case when the groups happen to be fairly close to one another. This causes a mislabeling at this specific time step. The algorithm only works if the agents are watched long enough that these anomolies represent a small percentage of the cases and on average, agents associated with the same cluster point are all associated with the same original group.

Although we don't have hard proof based on a series of humans choosing the values of k at which they think total distance stops decreasing significantly, the correlation between what we believed this value of k was and high original and classified group accuracy led us to believe that this algorithm is a useful candidate for dynamic group classification.

3 Link Classification

N ext, we developed a tool for link classification. We use logistic regression to classify link or non-link for any two agents in a group ($y^{(i)}$=1 or 0 respectively). We used the mean and variance of distance, velocity, and acceleration over a set number of simulation time steps as features for all links spanning two agents in the same group. Each state was observed over a series of time steps and at the end of the simulation the candidate feature vector, x, (in the sense of the below equations) was generated. We then stored the feature vector, x, and whether there was a link between two agents or not and restarted the simulation with another randomly generated group.We used logistic regression to classify the data with Newton's method to optimize for l(θ) as outlined below. We obtained a θ, and used that θ to classify the test data. We repeated this method for 60 groups whose size ranged from 4-10 agents and reported the average error for each feature. Below are the average errors (per simulation run) when various features were used. Although this data is biased since there are about 70 % links to not links, logistic regression proved useful as a link prediction tool as our end accuracy was greater than 70 % . We found that it performed better than GDA since the data was not Gaussian. As displayed in the below tables, we used forward search, to find that the best two features are variance of the distance and mean of the acceleration. Also below is an example of training and test data for our chosen best features.

$$h_\theta(x) = g(\theta^T x),\ g(z) = \frac{1}{1+e^{-z}}$$

$$l(\theta) = \sum_{i=1}^{m} y^{(i)}\ log\ h(x^{(i)}) + (1-y^{(i)})\ log(1-h(x^{(i)}))$$

$$\theta\ := \theta - H^{-1}\bigtriangledown_\theta\, l(\theta)$$

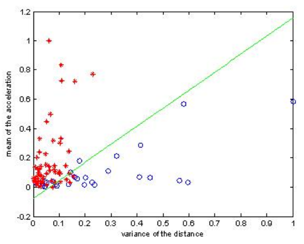
H is the Hessian matrix

Feature number=1

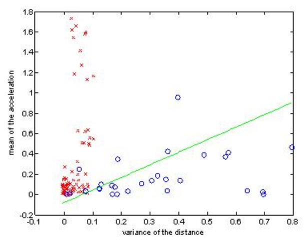
Feature	Average Accuracy
Mean of the distance	81.92%
Variance of the distance	**86.73%**
Mean of the relative velocity	80.58%
Variance of the relative velocity	78.27%
Mean of the acceleration	83.46%
Variance of the acceleration	80.77%

Feature number=2

Feature	Average Accuracy
Variance of the distance, Mean of the distance	83.08%
Variance of the distance, Mean of the relative velocity	87.88%
Variance of the distance, Variance of the relative velocity	86.08%
Variance of the distance, Mean of the acceleration	**89.42%**
Variance of the distance, Variance of the acceleration	84.61%



training

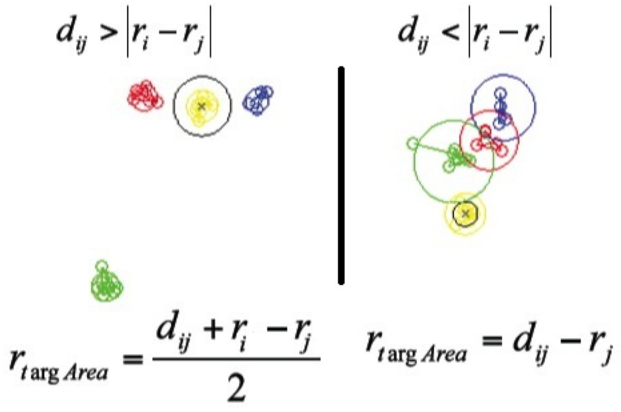


testing

Above is a one of the train/test runs we used in order to determine an average error. The above case corresponds to a 91.8% accurate run.

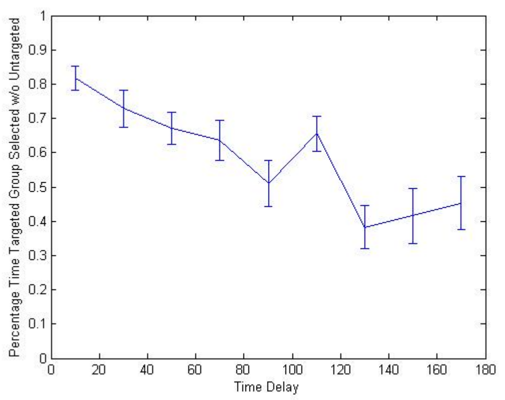
4 Group Behavior Prediction For Area-Based Attack Strategies

F inally, we developed a tool for predicting an area in which all agents of a targeted group could be attacked after some time delay without affecting other groups. Linear regression on the time history of each group's centroid and the average distance traveled along the line per time step were used to extrapolate the position of the group centroid after a time delay. The radius in which it could be expected all agents would fall within, was picked to be the max distance from centroid to any agent in any time step in the same training set. Finally, a radius about the targeted group, i's, expected centroid is calculated via the below equations where j the group whose calculated radius comes closest to i's.



$d_{ij} > |r_i - r_j|$
$d_{ij} < |r_i - r_j|$
$r_{t\arg Area} = \frac{d_{ij} + r_i - r_j}{2}$
$r_{t\arg Area} = d_{ij} - r_j$

The below figure shows the percentage time that the entire targeted group was located within the target area without any untargeted agents vs. time delay. As one might predict, the larger the time delay is, the worst the algorithm is at predicting a circular area which meets the criteria.



5 Conclusion

In this project, we develop tools to classify groups, classify links, and predict an area in which one can attack a target group after some delay (such as the delay after a bomb is launched) with as little collateral damage as possible.

We believe that these or similar tools, which use machine learning techniques, show promise in military applications and could be used in order to more quickly determine the nature of third party multi-agent systems. The tools could help to distinguish agents in one group from another, understand the rules under which a given group operates to destroy it faster, and determine the best time to launch an area based attack without affecting innocent parties.

References

[1] Balch, Tucker and Ronald C. Arkin."Behavior Based Formation Control for Multirobot Teams." IEEE Transactions on Robotics and Automation, Vol. 14.,1998.

[2] Leonard, Naomi Ehrich, and Edward Fiorelli."Virtual Leaders, Artificial Potentials and Coordinated Control of Groups." Proceedings of the 40th IEEE Conference on Decision and Control, 2001.

[3] Fiorelli, Edward and Naomi Ehrich Leonard. " Cooperative Control of Mobile Sensing Networks: Adaptive Gradient Climbing in a Distributed Environment." IEEE Transactions on Automatic Control, Vol. 49 No. 8, 2004.

[4] Murray, Richard M., and Reza Olfati-Saber. "Distributed Cooperative Control of Multiple Vehicle Formations Using Structural Potential Functions." IFAC World Conference, 2002.

[5] Raffard, Robin L., Claire J. Tomlin, and Stephen P. Boyd."Optimization for Cooperative Agents:Application to Formation Flight." 43rd IEEE Conference on Decision and Control, 2004.

[6] Murray, Richard M., and Reza Olfati-Saber. "Graph Rigidity and Distributed Formation Stabilization of Multi-Vehicle Systems." Proceedings of the 41st IEEE Conference on Decision and Control, 2002.

[7] O. Khatib,"Real time obstacle avoidance for manipulators and mobile robots." Int. J. Robotics Research, 90-99,1986.