A Multiple-Predictor Approach to Human Motion Prediction

Przemyslaw A. Lasota*[1] and Julie A. Shah[1]

Abstract— **The ability to accurately predict human motion is imperative for any human-robot interaction application in which the human and robot interact in close proximity to one another. Although a variety of human motion prediction approaches have already been developed, they are often designed for specific types of tasks or motions, and thus do not generalize well. Furthermore, it is not always obvious which of these methods is appropriate for a given task, making human motion prediction difficult to implement in practice.**

We address this problem by introducing a multiple-predictor system (MPS) for human motion prediction. In our approach, the system learns directly from task data in order to determine the most favorable parameters for each implemented prediction method and which combination of these predictors to use. Our implementation consists of three complementary methods: velocity-based position projection, time series classification, and sequence prediction. We describe the process of forming the MPS and our evaluation of its performance against the individual methods in terms of accuracy of predictions of human position over a range of look-ahead time values. We report that our method leads to a reduction in mean error of 18.5%, 28.9%, and 37.3% when compared with the three individual methods, respectively.

I. INTRODUCTION

A large variety of fields and applications stand to benefit from human-robot interaction and collaboration. In recent years, there has been a significant push toward introducing robots on factory floors [1], in homes [2], and even as assistants on board the International Space Station [3]. In these and other applications, close-proximity physical interaction is often required in order for robots to effectively collaborate with people. Consequently, there is a need for the development of techniques and methods that support safe and efficient physical human-robot interaction.

One way in which such interaction can be achieved is through robot adaptation based on prediction of human motion. By being able to anticipate where a person might be reaching or walking toward next, a robot can choose its actions or adjust its movement such that potential motion conflicts are avoided. Results from prior work indicate that the use of a human-aware motion planner, which avoids moving through locations of upcoming human occupancy, leads to more efficient teamwork, increased satisfaction with the robot, and higher perceived safety and comfort [4]. The ability to predict where a person will move to next is a key component of such an approach, motivating the development of accurate, real-time prediction of human motion.

As discussed in the following section, a variety of human motion prediction techniques have been recently developed; these include goal-based methods [5], [6], [7], [8], [9], [10], [11] and prediction based on the study of natural human motion [12], [13], [14], [15]. The majority of these approaches, however, were designed and tuned for specific types of motions or tasks, and thus would not necessarily generalize well to prediction in other domains; for example, predictors based on action models would not work well when applied to loosely structured tasks. This creates a barrier for utilizing human motion prediction, as it may not always be clear which approach is best suited for a specific task. In fact, it might even be possible that no single technique works well, and that a combination of approaches is required.

In order to address this drawback, we envisioned a data-driven approach to human motion prediction that, based on a variety of data encoding how a person moves within a shared workspace and how he or she performs tasks, will automatically select a favorable combination of prediction methods to accurately predict human occupancy at various future time frames. Such a technique would enable robust and generalizable prediction of human motion.

In this paper, we present a method that utilizes the relative performances of individual prediction methods — a velocity-based position projection method, a time series classification method, and a sequence prediction method — to form a multiple-predictor system (MPS) for human motion prediction. We show that by training on available task data, our method automatically learns to exploit the complementary strengths of each method to form a combined predictor that outperforms the three methods individually for a variety of task data variants that differ with respect to the number and sequence of actions.

II. RELATED WORK

Prior work in the field of human motion prediction can be classified into two main categories: works that rely upon prediction of goals and those that utilize motion characteristics without goal prediction.

The former category involves predicting the target or goal that a person is reaching or walking toward and then utilizing an appropriate motion model to predict how that person will move in transit to that goal. In one example of this approach, Gaussian Mixture Models (GMMs) are trained for each reaching goal position of a particular task, and Gaussian Mixture Regressions (GMRs) are used to generate representative reaching motions. Based on observations of

*Corresponding author: plasota@mit.edu

[1]Przemyslaw A. Lasota and Julie A. Shah are with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, Massachusetts.

This work was supported by a NASA Space Technology Research Fellowship.

the beginning segment of a new reaching motion and the computed GMMs and GMRs, the framework calculates the likelihood of human occupancy of the shared workspace via computed swept volumes [5]. Another recent goal-based prediction approach utilizes a time series analysis in which multivariate Gaussian distributions over tracked degrees of freedom of the human arm are computed for each time step of the motion. The learned models are then used to perform Bayesian classification on the initial stages of motion to predict what goal location the person is reaching toward [6].

Another method for anticipating motion through goal prediction explored in prior work is based on modeling of motions via inverse optimal control. In one such approach, the authors leveraged the assumption that people move efficiently when navigating a space and modeled human motions using maximum entropy inverse optimal control. By learning a cost function that is a linear combination of features based on objects in the environment, the approach can generalize to new situations in which objects change locations [7]. A different method, specifically designed for reaching motions performed during known, collaborative tasks, also utilizes inverse optimal control for predicting human motion, but in a manipulation scenario. This method incorporates example data of pairs of people performing a manipulation task to learn reaching motion cost functions that are then used with a human kinematic model and an iterative motion planner to predict where the person will reach [8].

The concept of social forces is also useful in the context of motion prediction. For example, Growing Hidden Markov Models (GHMMs) and the social forces method are used to infer goals from partial trajectories and to then predict the path a person will take toward that goal [16]. This work builds upon the social forces-based human motion models and predictors presented by Luber et al [9].

Objects and other constraints in the environment can also prove useful for goal-based prediction of human motion. In one example of such work, RGB-D input and object affordances are used to make goal predictions based on the human's current pose and surroundings. Bzier curves are then used to define the potential paths of the human hand toward the predicted goal [10]. An extension of this work combines a high-dimensional model, used for properly assessing physical plausibility, with a low-dimension representation used in learning relationships between people and objects, allowing for improved motion trajectory prediction [11].

Another approach to goal prediction is to reason directly on previously observed action sequences. One technique explored in prior work is to compare a current action sequence to all those observed in the past to determine whether the current sequence is an instance of an action that has been previously observed. As the system correctly anticipates a particular sequence and the user validates the prediction, the system builds confidence about that sequence and adjusts its anticipatory actions accordingly [17]. In another work on sequence prediction, the developed framework is built on the concept that the set of actions in a sequence has predictive power over future actions and not the specific order of that set. This approach incorporates a vector of variables that defines relationships between items in the observed part of the sequence and potential future items, which is learned via an optimization process [18].

The second major category of human motion prediction focuses on analyzing how people move and plan natural paths without predicting specific goal locations. One such method employs motion capture data to encode skeletal motion patterns as Hidden Markov Models, with the goal of encoding likely transitions between motion patterns [12].

A different approach, based on the principle of maximum entropy, considers features such as the amount of time needed to reach a goal, acceleration profiles, walking velocity, and collision avoidance behavior. The authors investigated which of these features of walking motion could be utilized to learn how to characterize and predict typical human walking behavior [13].

Xiao et al. [14] developed a framework that uses previously observed human trajectories to train an SVM classifier into high-level classes (e.g., "wandering" or "stopping") and then forms clusters within these classes. The clusters are then used to develop prototypes that are matched to observed partial trajectories for prediction.

Finally, Unhelkar et al. instructed study participants to walk toward several locations in a room and recorded their position and head orientation [15]. Results from this work indicated that the head orientation and body velocity normalized by height can signal the direction in which a person will turn prior to the physical turn itself. These indicators were also employed successfully for goal-based prediction using the method by Pérez-D'Arpino and Shah [6].

While many of the approaches mentioned above are capable of accurate motion prediction, they are often developed with a certain class of tasks in mind, and thus might not generalize well to other scenarios. The algorithms and techniques used also require careful tuning of various model parameters in order to achieve accurate prediction. We aimed to learn the parameters for different methods and form a combined predictor that is automatically tuned to perform well across different classes of tasks.

III. METHODS OF HUMAN MOTION PREDICTION

Multiple classifier systems (MCS) are often designed to incorporate mutually complementary individual classifiers [19]. Following this guiding principle, we selected a set of three methods of predicting human motion for our multiple-predictor system: velocity-based position projection, time series classification, and action sequence prediction.

The complementary nature of these techniques is derived from their spanning many categories and sub-categories of motion prediction. The first two methods, for example, reason directly on observed human motion, while the third technique utilizes discrete action labels as input. The last two methods focus on predicting action goals as a proxy for predicting position, while the velocity-based position projection technique assesses motion directly without predicting goals. By combining these three techniques, we aimed to form a

multiple-predictor system capable of producing accurate predictions under a variety of scenarios that no single predictor could adequately address alone.

A. Velocity-Based Position Projection

The first method of human motion prediction used in our framework is based on projecting the human's current position through an estimate of his or her velocity. Namely, once an estimate of the current velocity, $\widetilde{\boldsymbol{v}}_t$, is obtained, this method computes $\widehat{\boldsymbol{x}}_{t+\Delta T}$, the predicted position in ΔT seconds, by assuming the human will maintain the same velocity for that time period:

$$\widehat{\boldsymbol{x}}_{t+\Delta T} = \boldsymbol{x}_t + \widetilde{\boldsymbol{v}}_t \cdot \Delta T \tag{1}$$

To allow for the use of this technique in online motion prediction, $\widetilde{\boldsymbol{v}}_t$ must be computed from observed position data prior to t, $\{\boldsymbol{x}_i\}_{i=1..t}$. This precludes the use of many common filtering techniques that utilize the complete position time series to calculate estimates of velocity at each time t. Furthermore, online prediction necessitates a computationally efficient approach for calculating $\widetilde{\boldsymbol{v}}_t$, so that predictions can be made rapidly as new position data is gathered.

Consequently, we elected to use the Savitzky-Golay Filter [20] to smooth position data and compute velocity estimates. This method works by fitting low-degree polynomials to successive sets of points, and thus does not require the entire trajectory in order to perform smoothing. Furthermore, if the position data is sampled at a uniform rate, there exists an analytical solution to the least-squares fit that can be represented by a set of coefficients. A simple convolution of these pre-computed coefficients with the successive position data can be used to compute a smoothed position signal and its derivatives, rendering the process of estimating velocity from observed position data with this method computationally efficient.

The convolution coefficients of the Savitzky-Golay method are a function of two main parameters. The first is the order of the polynomial to be fit to the data. The best polynomial order to select depends upon the attributes of the position signal, such as the amount of noise present and the sampling rate. The second parameter is the frame size, which defines which portion of the observed position signal is to be used for estimating the current velocity. Namely, for a uniformly sampled signal with time steps $t = 1 \dots T$, a frame size of f indicates that, at time t, the set of positions $\{\boldsymbol{x}_i\}_{i=t-f \dots t}$ will be used for velocity estimation.

B. Time Series Classification

The second motion prediction method in our framework is an extension of the goal-based time series classification method developed by Pérez-D'Arpino and Shah [6]. This approach uses human demonstrations of motion toward several goal locations to build a library of representative motions based on statistical analysis of tracked degrees of freedom (e.g., positions of various points on the body, head orientation, etc.). Each time step of the motion is encoded as a multivariate Gaussian distribution over these degrees of freedom. The intended goal location is predicted based on early stages of motion by calculating the maximum likelihood that a given partial trajectory belongs to one of the motion classes in the training set.

While this technique can predict the goal location a person is walking or reaching toward, it does not directly allow for prediction of that person's position in a given amount of time, $\widehat{\boldsymbol{x}}_{t+\Delta T}$, which is the desired output of our framework. Consequently, we extended the approach to utilize the predicted goals for position prediction by making use of the mean trajectories of each computed motion class. We denote the mean trajectory of goal g as the set of positions $\boldsymbol{x}_t^g$ sampled at uniform time steps $t = 1 \dots T_g$, or $\{\boldsymbol{x}_i^g\}_{i=1 \dots T_g}$.

The first step of this extension of the original time series classification method is to search the mean trajectory of a goal's motion class to identify a suitable representative point that corresponds to the current observed position, $\boldsymbol{x}_t$. Specifically, based on the predicted goal g and current time step t, our method selects a representative point $\boldsymbol{x}_\lambda^g$, where the index λ is selected by identifying the point in the mean trajectory that is closest to the current position within a moving window with a size defined by the parameter α:

$$\lambda = \underset{i \in [t-\alpha \dots t+\alpha]}{argmin} \|\boldsymbol{x}_i^g - \boldsymbol{x}_t\| \tag{2}$$

The motivation for searching within this moving window is to allow for some temporal misalignment between the mean and observed trajectories. We also attempted to use an implementation of online Dynamic Time Warping (DTW) for this task, which led to poorly selected $\boldsymbol{x}_\lambda^g$ in practice. Although we did not do so in our current implementation, the selection of an algorithm for temporal alignment can be incorporated as another learned parameter of the time series classification method, as it is possible that online DTW would perform well on a different data set.

Once the representative point $\boldsymbol{x}_\lambda^g$ is identified, the algorithm steps forward in the mean trajectory from time step λ until it reaches a point that is the desired ΔT ahead. Assuming a sampling rate of f Hz, the predicted position is represented as follows:

$$\widehat{\boldsymbol{x}}_{t+\Delta T} = \boldsymbol{x}_{\lambda + f \cdot \Delta T}^g \tag{3}$$

There are two main parameters that must be selected when using this prediction method. The first is the set of indices of the available degrees of freedom to be used for prediction. While the coordinates of a person's hand can certainly provide a useful signal for predicting the goal that person is reaching toward, this may not be the case with other tracked degrees of freedom, such as head orientation. The degrees of freedom that will be most effective for prediction will depend upon the given task.

The second tunable parameter is the window size, α. Higher values of α allow for greater tolerance to temporal misalignment between the mean and currently observed trajectories, but setting the value too high can result in a poor choice of $\boldsymbol{x}_\lambda^g$, especially if the given motions pass over the same regions of space multiple times within a single trajectory.

C. Sequence Prediction

The final motion prediction method implemented in our framework is based on the sequence prediction algorithm developed by Letham et al [18]. One key difference between this method and other sequence prediction approaches is that it reasons on what sets of actions occur before others, and not on the specific order in which these actions occurred. This dramatically reduces the dimensionality of the problem, as it is not necessary to consider all specific orderings of actions, allowing for prediction of sequences with large numbers of possible actions — a desirable ability for a generalizable prediction framework.

Similarly to the time series classification method described in the previous section, the goal of the sequence prediction method is to predict which action a person will take — and, therefore, which goal region he or she will move toward. Unlike the time series classification and velocity-based position projection methods, however, the sequence prediction method reasons on discrete action labels as input instead of working with raw position data.

The implemented sequence prediction approach by Letham et al. incorporates a set of previously observed sequences to learn a set of values, $\lambda_{a,b}$, that describe the relationship between combinations of actions a and b. Large values of $\lambda_{a,b}$ indicate that actions a and b appear together often. The vector of these values, $\boldsymbol{\lambda}$, is then used in a scoring function, which, given a partial sequence of actions and a candidate next action, assigns a score to this action by taking the sum of the λ values that relate the actions in the partial sequence and the candidate action. This is the "one stage" scoring model described in the publication by Letham et al. The candidate action with the highest score is then selected as the action most likely to occur next in the sequence.

The vector $\boldsymbol{\lambda}$ is fit through an optimization of a loss function based on the scoring function, denoted f, and a rule defining which actions should strictly be ranked higher than others based on a given partial training sequence and the known remaining actions. In our implementation, we simply designated that for any given partial training sequence, the next action within that sequence should be ranked higher than all other possible actions.

Using notation from the paper by Letham et al., given a set of training sequences (denoted X_1^m) indexed $i = 1 \dots m$ of length T_i, with the next action in the training sequence i at time t being $k_{i,t}$, the set of all other actions being $L_{i,t}$, the partial observed sequence i at time t being $x_{i,t}$, and N being the number of possible actions, the loss function we are trying to minimize is represented as follows:

$$R(f, X_1^m; \boldsymbol{\lambda}) = \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{T_i - 1} \frac{1}{T_i} \frac{1}{N-1} \times \sum_{l \in L_{i,t}} e^{f(x_{i,t}, k_{i,t}; \boldsymbol{\lambda}) - f(x_{i,t}, l; \boldsymbol{\lambda})} + \beta \|\boldsymbol{\lambda}\|_2^2 \quad (4)$$

The final term is an l_2-norm regularization scaled by β. The vector $\boldsymbol{\lambda}$ is derived by running an unconstrained optimization routine that minimizes the value of R given the training set of sequences and the scoring function f.

Once $\boldsymbol{\lambda}$ is computed from the training data, prediction of the next action in a new sequence is performed by identifying the candidate action a with the highest value of $f(x_{i,t}, a; \boldsymbol{\lambda})$. Similarly to the time series classification method, however, the sequence prediction method must predict not only the next goal, but also estimated future positions, $\widehat{\boldsymbol{x}}_{t+\Delta T}$. We utilized the same approach as that used for the time series classification method, incorporating the computed mean trajectories for each action and equations (2) and (3).

IV. Formulation of the Multiple-Predictor System

Next, we discuss the strategy we employed for synthesizing individual predictors into a multiple-predictor system through a two-stage process. An outline of the system architecture is depicted in Figure 1. First, a subset of the data, $\mathcal{D}_{Train}$, is used to learn the parameters of each individual prediction method. Next, using the parameters found in the first step, each prediction method is used to compute predictions on a second subset of the data, $\mathcal{D}_{ModelSelection}$. The goal of each predictor (and the framework itself) is to generate a prediction of position for a given amount of time in the future, $\widehat{\boldsymbol{x}}_{t+\Delta T}$. The prediction results from each method are employed in the formulation of the rules the multiple-predictor system uses to select which prediction methods to use.

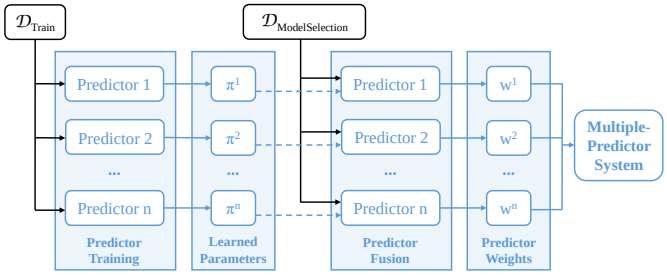

Fig. 1. An outline of the process of forming the multiple-predictor system.

A. Training Individual Component Methods

The first step toward formulating the multiple-predictor system involves learning the parameters of each individual prediction method. As the goal is to generate predictions of position for a given amount of time in the future, a natural global parameter to consider when training all of the prediction methods is ΔT, which we refer to as the "look-ahead time."

Consider a set of discrete look-ahead times of interest of length k, defined as $\mathcal{L}$. We denote the individual look-ahead time values as l_p, where $p = 1 \dots k$. Given n available prediction methods, for each method $i = 1 \dots n$, we define the set of parameters of a method as $\mathcal{Z}^i = \{z_1^i \dots z_{m^i}^i\}$, where m^i is the number of tunable parameters for method i. Constraining the individual parameter values to given discrete sets, the goal is to identify a set of assignments to

each z_j^i that minimizes the training error of each prediction method for each $l_p \in \mathcal{L}$. If the overall training error of method i given look-ahead time l_p, parameter assignments $\mathcal{Z}^i$, and input data D is defined to be E^i, the parameter assignment from the specified discrete sets that achieves the highest performance for that method and look-ahead time is as follows:

$$\pi_p^i = \underset{\mathcal{Z}^i}{argmin}\, E^i(l_p, D; \mathcal{Z}^i) \tag{5}$$

The training process, therefore, results in $k \times n$ set assignments π_p^i. Defining d as the individual trajectories contained in D and $T_{i,d}^*$ as the set of time steps of trajectory d for which a prediction can be produced by predictor i, the training error E^i is defined as the mean value of these prediction errors:

$$E^i(l_p, D; \mathcal{Z}^i) = \frac{1}{|D|} \sum_{d \in D} \frac{1}{|T_{i,d}^*|} \sum_{t \in T_{i,d}^*} \|(\widehat{\boldsymbol{x}}_{t+l_p})^i - \boldsymbol{x}_{t+l_p}\| \tag{6}$$

$(\widehat{\boldsymbol{x}}_{t+l_p})^i$ is the position prediction of method i, which is a function of the parameter values $\mathcal{Z}^i$. Assessing the prediction methods described in Section III, and indexing them in the order they were presented, note that for the velocity-based position projection there are $m^1 = 2$ tunable parameters as part of $\mathcal{Z}^1$. These two parameters are the polynomial order and frame size for the Savitzky-Golay filter.

For the time series prediction method, $\mathcal{Z}^2$ also consists of two components. The first tunable parameter is the subset of given degrees of freedom of the input data to use for prediction. Defining the set of all tracked degrees of freedom as $\mathcal{S}$, the set of position coordinates as $\mathcal{X} \subseteq \mathcal{S}$, and the set of the remaining degrees of freedom as $\mathcal{Q} = \mathcal{S} \setminus \mathcal{X}$, the combinations of degrees of freedom that are considered in the training phase, $\mathcal{S}^*$, consist of unions of set $\mathcal{X}$ with each of the elements of the powerset of $\mathcal{Q}$: $\mathcal{S}^* = \{\mathcal{X} \cup q^* \mid q^* \in \mathcal{P}(\mathcal{Q})\}$. The second parameter of the time series prediction method is simply the window size, α.

Finally, based on the given formulation, the sequence prediction method actually contains only one tunable parameter, the window size α. The training function also incorporates the set of action sequences in the training data, along with Equation (4), to compute the vector $\boldsymbol{\lambda}$ via an unconstrained optimization routine. Unlike the parameter α, however, the λ values are not a function of look-ahead time and are learned directly from the sequence data, rather than by minimizing the position prediction error E^i.

B. Fusion of Prediction Methods

Once the sets of parameters as a function of look-ahead time for each individual method are learned during the training phase, the next step in formulating our multiple-predictor system involves determining how to most effectively combine the outputs of these different methods.

When establishing the combination of position predictors, it is necessary to combine the output of several independent learners. Furthermore, the output of the predictors is the future position $\widehat{\boldsymbol{x}}_{t+\Delta T}$, which is a continuous variable, rather than an element of a discrete set of classes. This type of problem lends itself to a class of methods designed for combining the predictions of several experts by evaluating their relative performances and learning which experts tend to produce the best predictions, such as the Hedge Algorithm [21], or the Weighted Average Algorithm [22]

One algorithm within this class — the one we elected to use in our framework — is the Polynomial Weights (PW) algorithm [23]. In this algorithm, each predictor i is treated as an "expert" and is initially assigned a weight $w_0^i = 1$. Then, at each successive time step $t = 1 \ldots T$, an expert i is chosen at random in proportion to its normalized weight, $w_t^i / \sum_{i=1}^N w_t^i$. The weight of each expert is then updated by setting $w_{t+1}^i = w_t^i \cdot (1 - \epsilon L_t^i)$, where $L_t^i \in [0, 1]$ is the prediction loss of expert i at time t and ϵ is the learning rate. As the total number of time steps in all of the trajectories in $\mathcal{D}_{ModelSelection}$ is known, it is possible to set the learning rate ϵ to its optimal value, which is given by $\epsilon = \sqrt{\frac{ln(N)}{T}}$.

Since the output of our multiple-predictor framework is a continuous variable, the prediction error, defined as the distance between the predicted and ground truth positions, is also a continuous variable. Consequently, one of the main advantages of using PW in our framework over other "combination of experts" algorithms is that it includes a continuous loss function. As this function is limited to the range $[0, 1]$, our fusion method normalizes the prediction errors with respect to the mean μ and standard deviation σ of the errors encountered in the training phase. Specifically, we define our loss function as follows:

$$L_t^i = min\left\{ \frac{\|(\widehat{\boldsymbol{x}}_{t+\Delta T})^i - \boldsymbol{x}_{t+\Delta T}\|}{\mu + \sigma}, 1 \right\} \tag{7}$$

As mentioned in the previous section, during the training phase an assignment of individual method parameters for each method and look-ahead time of interest is computed. Consequently, when forming the combined predictor, the predictor fusion routine runs the PW algorithm for each $l_p \in \mathcal{L}$. Defining the vector of final method weights for each look-ahead time l_p as $\boldsymbol{W}_p = [w_T^1 \ldots w_T^n]$, the predictor fusion results in k such sets (recall that $|\mathcal{L}| = k$).

Once the final method weight sets $\boldsymbol{W}_p$ are known, our predictor fusion method forms the combined predictor by defining which method should be used at each $l_p \in \mathcal{L}$. In our implementation, the predictor to use at look-ahead time l_p, defined as i_p^*, is represented as follows:

$$i_p^* = \underset{i}{argmax}(\boldsymbol{W}_p(i)) \tag{8}$$

$\boldsymbol{W}_p(i)$ returns the i^{th} element of $\boldsymbol{W}_p$. The reason for using (8) as our predictor selection rule, as opposed to selecting a method at random in proportion to the values in $\boldsymbol{W}_p$, is that due to potential safety-critical applications of position prediction (e.g., robot motion planning during human-robot interaction), it is undesirable to query a predictor that returns inaccurate positions, even if that predictor is selected with low probability.

V. EVALUATION

A. Source Data Set

We evaluated the efficacy of our proposed multiple-predictor method of human motion prediction via a human-robot interaction data set obtained from a previous experiment [4]. In this prior study, 20 participants collaborated with an industrial robot arm on a tabletop task that required participants to place screws at designated locations while the robot pretended to apply a sealant over the screws. The three-dimensional position of the participants' wrist was tracked using a motion capture system while the person reached between the screw pickup location and designated placement locations. The data was collected at an average rate of 140Hz.

The robot operated in one of two motion planning modes: a human-aware mode in which the robot selected paths that avoided portions of the shared workspace that the human was expected to occupy, and a standard mode in which the robot selected the quickest path to its goals without reasoning on where the human would be next. Each participant performed the task with both robot types, and within each task performed eight screw placements in a preset sequence. A video depicting sample task executions of the human-aware and standard modes is available at the following link: http://youtu.be/Dk5XVQBDJpU.

B. Data Set Variants

One useful quality of the experiment data set described in the previous section is that each complete task execution can be segmented into eight distinct placement trajectories, one for each screw. Furthermore, due to the participants always reaching back to the same location in order to pick up a new screw, it is possible to create new variations of the task in which the order of the actions is altered (or only a subset of the actions is performed).

We exploited these qualities to generate 12 variations of the dataset with which to evaluate our combined predictor. In these variants, we manipulated the action order by drawing the next action from a uniform distribution over the set of actions not yet performed in a given sequence, for either the data from all 20 subjects, 25% of the subjects, or for none of the subjects (causing the action sequence to be identical for all demonstrations). We also manipulated the number of actions to be either the full set of eight original actions or a subset of four actions with goal locations spaced roughly equally apart from one another. The goal of generating these variants was to evaluate how well our method generalizes across the spectrum of more- and less-structured tasks, as well as tasks with several nearby goal locations versus fewer, more separate goal locations. A summary of these data sets is depicted in Table I.

C. Evaluation Strategy

In order to evaluate the relative performances of the individual prediction methods and our combined predictor, we performed a leave-one-out cross-validation with the 20 sets of trajectories. We repeated this process for each dataset variant shown in Table I.

TABLE I
DATA SET VARIANTS

Data Set	Actions Used	Action Order	Robot Mode
1	All	1,2,3,4,5,6,7,8	Human-Aware
2	1,4,5,8	1,4,5,8	Human-Aware
3	All	Random	Human-Aware
4	1,4,5,8	Random	Human-Aware
5	All	25% Random	Human-Aware
6	1,4,5,8	25% Random	Human-Aware
7	All	1,2,3,4,5,6,7,8	Standard
8	1,4,5,8	1,4,5,8	Standard
9	All	Random	Standard
10	1,4,5,8	Random	Standard
11	All	25% Random	Standard
12	1,4,5,8	25% Random	Standard

For each iteration of the cross-validation, we retained one subject's data for testing and randomly assigned the remaining data to $\mathcal{D}_{Train}$ and $\mathcal{D}_{ModelSelection}$, with roughly 70% of the data assigned to the former (13 of 19 subjects' data) and 30% to the latter (6 of 19).

For each of the tests, we computed the mean prediction error as a function of look-ahead time for the three independent prediction methods, which served as baselines for evaluation, and the combined predictor. The range of look-ahead times considered was 0.05s to 0.5s, in increments of 0.025s. The upper bound of the look-ahead-time range was dictated by the length of the example trajectories, 21% of which were less than 1.5s in length. We present the results of our evaluation in the following section.

VI. RESULTS AND DISCUSSION

A. Performance Computations and Statistical Analysis

In order to analyze our results, we retrieved the data from each of the 20 individual cross-validation outputs (one per original experiment participant) and computed the mean prediction errors for each of the four prediction methods (i.e., the three individual methods and the multiple-predictor system). We combined these results into separate vectors of prediction errors for each predictor, with one entry per participant. We then took the means of these vectors to calculate the overall mean prediction error of each individual prediction method, as well as the multiple-predictor system. This process was repeated for each of the 12 data set variants depicted in Table I. The final mean prediction errors, in meters, are shown in Table II, where the velocity-based position projection method is abbreviated as VBPP, the time series classification method as TSC, the sequence prediction method as SP, and the multiple-predictor system as MPS.

As the mean errors of each method evaluated on a specific participant's data are dependent values, we utilized a repeated-measures ANOVA to analyze mean prediction errors. The treatments of the ANOVA are the four predictor types (the three individual methods and the multiple-predictor system). The results of the ANOVA indicate that the effect of the prediction method is significant, at a confidence level of $p<0.05$, for all dataset variants, with the exception of Data Set 5 ($p=0.1$).

TABLE II

MEAN PREDICTION ERRORS (METERS)

Data Set	VBPP	TSC	SP	MPS
1	0.119	0.124	0.101	**0.082***
2	0.117	0.124	0.102	**0.082***
3	0.119	0.129	0.240	**0.100***
4	0.117	0.123	0.205	**0.097***
5	0.119	0.129	0.140	**0.105**‡
6	0.117	0.124	0.118	**0.092***
7	0.108	0.134	0.110	**0.084***
8	0.106	0.130	0.112	**0.085***
9	0.108	0.134	0.273	**0.099***
10	0.106	0.132	0.245	**0.097**†
11	0.108	0.134	0.151	**0.101***
12	0.106	0.132	0.138	**0.092***

* MPS error lower than all individual methods ($p<0.05$)
† MPS error lower than TSC and SP only ($p=0.058$ for VBPP)
‡ Main effect of prediction method not significant ($p=0.1$)

To assess whether the decrease in mean prediction error for the multiple-predictor system was statistically significant with respect to each of the individual methods, we performed a post-hoc, pairwise comparison with paired samples t-tests and a Bonferroni correction. The results show that the mean prediction error of the multiple-predictor system was statistically significantly lower than all three of the individual methods, at a confidence level of $p<0.05$ — with the exception of Data Set 10, in which the difference between the multiple-predictor system and the velocity-based position projection was not significant ($p=0.058$).

B. Discussion of Key Findings

As indicated by the results in Table II, the multiple-predictor system nearly always outperformed the three individual prediction methods (with the two exceptions stated above). Among the differences in means that were statistically significant, the overall mean error (across all datasets) of the multiple-predictor system was 18.5%, 28.9%, and 37.3% lower than that of the individual VBPP, TSC, and SP methods, respectively. This result provides strong support for the concept of combining multiple predictors for prediction of human motion, and for our implementation of this concept.

By manipulating the number and order of actions in our dataset variants, we generated a variety of representative scenarios in which the individual prediction methods achieved differing levels of performance. For example, in scenarios in which only four of the eight actions were taken, the time series classification method generally performed better. This trend can be observed from lower mean errors for Data Sets 4, 6, 8, 10, and 12 compared with Data Sets 3, 5, 7, 9, 11, respectively. This is likely due to the fact that there are fewer possible goal regions that are more physically separated from each other, making the learned motion classes more distinct and leading to better prediction performance.

Changing the sequence in which the actions were performed represented another manipulation of the original dataset. In scenarios in which the next action was drawn from a uniform distribution over the set of remaining actions, the sequence prediction method performed poorly compared with scenarios in which the tasks were always completed in the same sequence. This can be determined by comparing the mean prediction error of this method for Data Set 7 and Data Set 9, in which the random order of actions caused the sequence prediction method's mean prediction error to more than double. One can also observe this effect by comparing mean prediction errors as a function of look-ahead time for these data sets shown in Figure 2 and Figure 3.

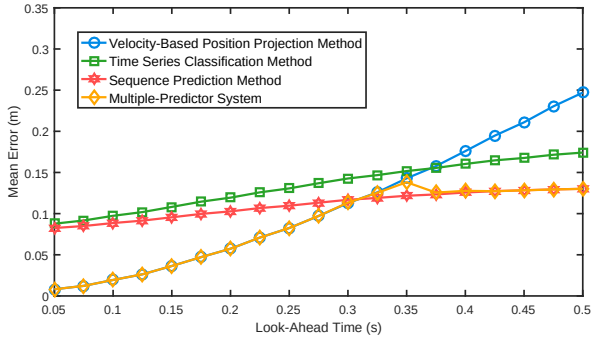

Fig. 2. Mean prediction errors as a function of look-ahead time for Data Set 7.

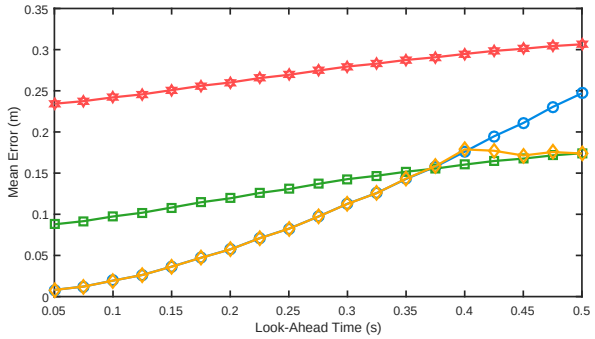

Fig. 3. Mean prediction errors as a function of look-ahead time for Data Set 9.

As expected, the other two individual methods, which do not reason on the sequence order, were not affected by the sequence randomization. Consequently, it is best not to rely upon the sequence prediction method in this case, and instead utilize the other two prediction methods.

Figure 3 indicates that the multiple-predictor system automatically learned this concept and constructed the combined predictor with the use of the time series classification and velocity-based methods. When the order of actions was consistent, on the other hand, our method constructed a combined predictor out of the sequence prediction and velocity-based methods, as depicted in Figure 2.

Besides down-selecting to two of the three methods appropriately, another interesting behavior of the multiple-predictor system is the way in which it combined the selected methods. Note from the plots that the velocity-based prediction method performed very well at low values of look-ahead time (ΔT), but that performance degraded quite rapidly as look-ahead time increased. This makes sense, since at larger look-ahead time values, the person might change direction during the time interval in question, resulting in poor prediction and higher mean error.

The time series classification and series prediction methods also exhibited a trend toward decreased performance at higher values of ΔT, but their mean error grew at a slower

rate. Furthermore, in cases in which the multiple-predictor system selected these two methods, the mean error at low values of ΔT was higher for these methods than for the velocity-based method, while this trend was reversed at high values of ΔT. One possible explanation for the relatively poorer performance at low values of ΔT is the variance in the example trajectories: If variance is high enough, the mean trajectory can be quite far from the trajectory being predicted.

While the relatively high error at low values of ΔT is not ideal, the time series and sequence prediction methods' performance degrades slowly with increasing values of ΔT. This is likely due to the fact that when these prediction methods correctly classify the goal the person is reaching toward, the mean trajectory of the motion class yields a much better prediction further into the future than simply assuming the person will continue to move their hand in a straight line.

These attributes create a natural "cross-over point" at which one method becomes superior to the other as a function of look-ahead time. The plots in Figures 2 and 3 indicate that the multiple-predictor system was able to learn and exploit this concept automatically.

The automatic adjustment in response to action order and long- versus short-term accuracy showcases the ability of the multiple-predictor system to learn from the given training data how best to combine the complementary strengths of several predictors, greatly improving the robustness and generalizability of human motion prediction compared with reliance upon a single prediction method. Our method achieved this goal quite well, with the mean error of the multiple-predictor system closely tracking the mean errors of the best performing methods at each look-ahead time. The only points at which our method did not select the best performing method was at the transition point between two methods, which is likely due to slight differences between the training and testing data during the leave-one-out cross-validation.

VII. CONCLUSION

In this work, we presented a novel method of human motion prediction that utilizes a combination of individual predictors to form a multiple-predictor system. We showed that our approach outperforms the individual prediction methods in terms of mean prediction error for a variety of scenarios generated from a human-robot interaction data set. Our results indicate that the multiple-predictor system was able to adapt to the various conditions of these scenarios and automatically learn which prediction methods should be used at the various look-ahead time values, highlighting the improved robustness and generalizability of our approach to variations in the number and order of actions.

As generalizability to motion type is important as well, one avenue of future work involves evaluating the performance of the MPS on ambulatory motions, such as walking. Another potential future direction involves extending our predictor fusion method to continuously adapt the learned weights of the individual algorithms as new predictions are made. As the PW algorithm we used for model selection is an online learning algorithm, our implementation lends itself well to this extension. We would also like to incorporate a metric of individual method confidence into the model selection step, as well as add biasing toward the production of continuous predicted trajectories. Finally, we intend to incorporate the MPS with an online planning approach to investigate the impact of improved prediction on the quality of human-robot interaction.

REFERENCES

[1] W. Knight, "Smart robots can now work right next to auto workers," *MIT Technology Review*, vol. 17, 2013.

[2] B. Graf, M. Hans, and R. D. Schraft, "Care-o-bot iidevelopment of a next generation robotic home assistant," *Autonomous robots*, vol. 16, no. 2, pp. 193–205, 2004.

[3] T. Fong, M. Micire, T. Morse, E. Park, C. Provencher, V. To, D. Wheeler, D. Mittman, R. J. Torres, and E. Smith, "Smart spheres: a telerobotic free-flyer for intravehicular activities in space," in *Proc. AIAA Space*, vol. 13, 2013.

[4] P. A. Lasota and J. A. Shah, "Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human-Robot Collaboration," *Human Factors*, vol. 57, pp. 21–33, Jan. 2015.

[5] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proceedings of IROS*, pp. 299–306, Ieee, Nov. 2013.

[6] C. Pérez-D'Arpino and J. A. Shah, "Fast Target Prediction of Human Reaching Motion for Cooperative Human-Robot Manipulation Tasks using Time Series Classification," in *Proceedings of ICRA*, 2015.

[7] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *Proc. of IROS*, pp. 3931–3936, 2009.

[8] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using Inverse Optimal Control and iterative re-planning," in *Proceedings of ICRA*, pp. 885–892, 2015.

[9] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *Proceedings of ICRA*, pp. 464–469, IEEE, May 2010.

[10] H. S. Koppula and A. Saxena, "Anticipating Human Activities using Object Affordances for Reactive Robotic Response," in *Proceedings of RSS*, 2013.

[11] Y. Jiang and A. Saxena, "Modeling High-Dimensional Humans for Activity Anticipation using Gaussian Process Latent CRFs," 2014.

[12] W. Takano, H. Imagawa, and Y. Nakamura, "Prediction of human behaviors in the future through symbolic inference," in *Proceedings of ICRA*, pp. 1970–1975, 2011.

[13] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-Based Prediction of Trajectories for Socially Compliant Navigation," in *Proceedings of RSS*, (Sydney, Australia), 2012.

[14] S. Xiao, Z. Wang, and J. Folkesson, "Unsupervised robot learning to predict person motion," in *Proceedings of ICRA*, pp. 691–696, 2015.

[15] V. V. Unhelkar, P. Claudia, L. Stirling, and J. A. Shah, "Human-Robot Co-Navigation using Anticipatory Indicators of Human Walking Motion," in *Proceedings of ICRA*, 2015.

[16] J. Elfring, R. van de Molengraft, and M. Steinbuch, "Learning intentions for improved human motion prediction," *Robotics and Autonomous Systems*, vol. 62, pp. 591–602, Apr. 2014.

[17] P. F. Dominey, G. Metta, F. Nori, and L. Natale, "Anticipation and initiative in human-humanoid interaction," in *International Conference on Humanoid Robots*, pp. 693–699, 2008.

[18] B. Letham, C. Rudin, and D. Madigan, "Sequential event prediction," *Machine Learning*, vol. 93, pp. 357–380, nov 2013.

[19] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.

[20] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures.," *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[21] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, aug 1997.

[22] J. Kivinen and M. K. Warmuth, "Averaging Expert Predictions," pp. 153–167, Springer Berlin Heidelberg, 1999.

[23] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*, vol. 1. Cambridge University Press Cambridge, 2007.