J Multimodal User Interfaces (2014) 8:231–241
DOI 10.1007/s12193-013-0132-1

ORIGINAL PAPER

Iterative perceptual learning for social behavior synthesis

Iwan de Kok · Ronald Poppe · Dirk Heylen

Received: 13 July 2012 / Accepted: 5 November 2013 / Published online: 16 November 2013

Abstract We introduce Iterative Perceptual Learning (IPL), a novel approach to learn computational models for social behavior synthesis from corpora of human–human interactions. IPL combines perceptual evaluation with iterative model refinement. Human observers rate the appropriateness of synthesized behaviors in the context of a conversation. These ratings are used to refine the machine learning models that predict the social signal timings. As the ratings correspond to those moments in the conversation where the production of a specific behavior is inappropriate, we regard features extracted at these moments as negative samples for the training of a classifier. This is an advantage over the traditional corpus-based approach to extract negative samples at random non-positive moments. We perform a comparison between IPL and the traditional corpus-based approach on the timing of backchannels for a listener in speaker–listener dialogs. While both models perform similarly in terms of precision and recall scores, there is a tendency that the backchannels generated with IPL are rated as more appropriate. We additionally investigate the effect of the amount of available training data and the variation of training data on the outcome of the models.

This publication was supported by the Dutch national program COMMIT and the EU FP7 project SSPNet. We would like to thank the anonymous reviewers for their constructive feedback, which helped us to improve the paper.

I. de Kok · R. Poppe (✉) · D. Heylen
Human Media Interaction Group, University of Twente,
P.O. Box 217, 7500 AE Enschede, The Netherlands
e-mail: r.w.poppe@utwente.nl

I. de Kok
e-mail: i.a.dekok@utwente.nl

D. Heylen
e-mail: d.k.j.heylen@utwente.nl

Keywords Social behavior synthesis · Machine learning · Perceptual evaluation · Backchannel

1 Introduction

In this paper, we address the learning of computational models for the synthesis of human behavior in conversational settings. We target the setting where a human interacts verbally and nonverbally with an intelligent virtual agent (IVA). The aim is to make this human–machine interaction as close as possible to natural human–human interaction. From a machine perspective, this requires that appropriate responsive behavior is displayed to the human (see Fig. 1(top)). A common approach to endow IVAs with this ability is to learn conditional responsive behavior patterns from a corpus of human–human dialogs [20]. The verbal and nonverbal behavior of a dialog partner is continuously encoded in feature vectors of, e.g., speech activity, gaze direction or body movement. In addition, discrete social behaviors are identified in time. Examples are smiles as a reaction to observed facial movements or backchannels as a reaction to a speaker's speech and gaze. The task of the synthesis model (i.e., classifier) is to associate (probability) scores for the synthesis of these behaviors to feature instances of the dialog partner's behavior.

The application of this corpus-based learning approach for human behavior synthesis is widespread [20], but suffers from two main drawbacks. First, the evaluation of the synthesized behavior is typically measured by comparing it to the behavior performed in the corpus. While this is an objective measure, it does not take into account the *optionality* (or individuality) of social behavior. We argue that social behavior different from that in the corpus can also be appropriate. However, objective measures will discredit such alterna-

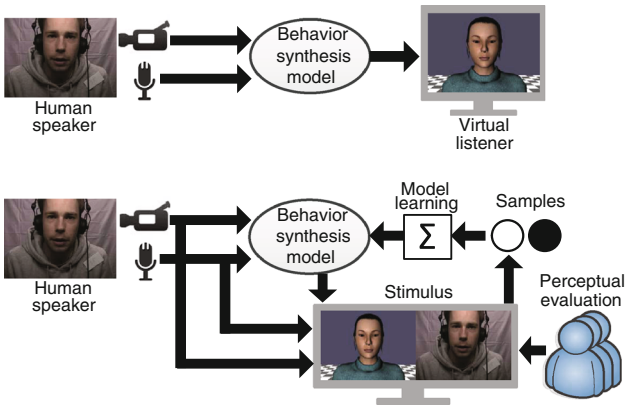

Fig. 1 Schematic overview of social behavior synthesis for an artificial listing agent (*top*) and the setting of our IPL framework (*bottom*)

tive behavior. This might eventually hinder the generalization ability of the behavior synthesis model.

Second, a classifier is typically trained with feature instances extracted slightly before the occurrence of a social behavior. These instances are considered *positive* samples. Typically, random feature instances that do not overlap with these positive samples are used as *negative* samples. However, while a social behavior was not performed in the actual dialog, there is no guarantee that it would be perceived as inappropriate if it had been performed. Consequently, some of the negative samples could also be regarded as positive samples. Having ambiguously labeled samples typically hinders the learning of a classification model.

In this paper, we describe a novel approach that addresses these drawbacks. Instead of relying on objective measures, we obtain subjective ratings regarding the appropriateness of the synthesized behavior. We use these ratings not only to evaluate the quality of the behavior but also as samples to iteratively re-train the classifier. This approach is a variant of active learning [28], where human raters provide the labeling of samples that have been generated by a machine learning classifier. We focus on obtaining negative samples that correspond to moments where the production of a specific social signal is inappropriate. We have termed our approach *Iterative Perceptual Learning* (IPL). Figure 1 shows a schematic overview of the IPL approach. IPL is general in the sense that it can be applied to the learning and synthesis of a broad range of social behaviors in dialogs. In addition, the approach is independent of the choice of machine learning classifier and features.

The contribution of this paper is a novel learning framework. We present a novel approach to learn synthesis models for social behaviors by combining perceptual evaluation and machine learning. We use subjective, perceptual ratings to measure the appropriateness of individual instances of social behavior. At the same time, we obtain samples (moments in the dialog) where the production of a specific social behavior is regarded as inappropriate. Given the availability of these negative samples, we train machine learning classifiers for the synthesis of the timings of social behaviors. By iteratively training and evaluating the resulting synthesized behavior, we refine the performance of the classifier by focusing the samples on those feature instances that are relevant.

We evaluate the IPL approach for the synthesis of backchannel timings in speaker–listener dialogs. IPL is compared to the common corpus-based approach where negative samples are obtained from the pool of non-positive samples. Our experiment contributes to the understanding of the strengths and weaknesses of both approaches. Based on several hours of dialog, we analyze the influence of the negative samples and the amount of available data on both the objectively and subjectively measured quality of the synthesized listening behavior.

The remainder of this paper is organized as follows. We first discuss related work on learning social behavior synthesis models. We introduce the IPL approach in Sect. 3. In Sects. 4 and 5, we describe, respectively, the setup and the results of an experiment on the synthesis of backchannel timings. We conclude in Sect. 6.

2 Related work

The field of social signal processing [31] addresses computational approaches towards the automatic understanding, modeling and generation of human social behavior in artificial agents and robots. In this work, we focus on the synthesis of nonverbal behavioral cues. Previous work on this topic has addressed, among others, the decisions of when to produce backchannels [13,18,21], eye gaze [24], smiles [3] or head gestures during speech [19].

These synthesis models are typically based either on hand-crafted rules [24,27] or on machine learning algorithms [21]. Both give a (probability) score for the production of a social signal at a selected moment, given feature instances obtained from observations of the conversational partners. Due to the real-time nature of interactions, the methods use shallow features in the sense that they are non-lexical and are derived directly from the audio or video signal. Hand-crafted rules are usually intuitive and can be based on known patterns in human social behavior, such as observed mutual gaze [1,15] or rising or falling pitch of the speaker [10,32]. Specifying these rules based on shallow features is not trivial. Therefore, recent work has increasingly addressed employing machine learning algorithms to learn behavior synthesis models. For the decisions of when to produce backchannels, which we will address in detail in Sect. 4, different machine learning classifiers have been explored, including decision trees [22], Hidden Markov Models [23] and Conditional Random Fields [21].

Machine learning models are trained by providing samples (i.e., feature vectors) to a learning algorithm. For social behavior modeling, positive samples correspond to feature instances extracted at, or slightly before, moments in a dialog where the production of a specific behavior is appropriate. The dominant approach to obtain these samples is to record a corpus of human–human interactions in a similar conversational setting and to identify the moments in time where a specific behavior is displayed [20]. In general, the number of such moments is relatively small and there are probably many appropriate moments where no social behavior has been produced. This is due to differences in behavior between individuals (e.g., in the number and timing), which is a consequence of the optional nature of social signals.

Negative samples are usually extracted at random moments within the conversation with the constraint that they do not overlap with positive samples. As a result of the optional nature, these negative samples could be extracted at moments in time where the production of a social signal is appropriate, but not present in the corpus. The classifier will therefore try to label these moments as inappropriate, which is likely to reduce the quality of the classifications. One way to handle this is to assign a weight to each sample, and determine the overall performance of a classification model as a weighted sum of all sample (mis)classifications. However, assigning such weights is difficult, unless we look at the overlap in time with other samples. In this case, positive samples that overlap with negative samples could be assigned a zero weight, or the other way around. Here, we circumvent this issue by not sampling these points altogether, which should make the training of the classification model more efficient.

Currently, the optional nature of social signals is also not reflected in the evaluation practice of machine learning models that generate their timings. In general, the quality of a behavior synthesis model is evaluated in terms of precision and recall of the generated social behaviors compared to those performed by the actual subjects in the corpus. Any deviation from the actually performed behavior results in lower scores. This is an undesired effect as there is no guarantee that the generated listening behavior is also *perceived* as less appropriate.

In sum, one of the key challenges in social behavior synthesis is to obtain appropriate positive and negative samples. This will help in learning behavior synthesis models that are better able to generalize. In addition, it allows for the perceptual evaluation of the synthesized social behavior. Several studies have addressed this challenge. To obtain more samples, De Kok and Heylen [18] recorded three listeners that interacted in parallel with the same speaker. The result of their Parallel Listener Consensus approach is a larger pool of positive samples compared to the setting where only a single listener interacted with the speaker. In addition, by analyzing when multiple listeners produced a social signal, moments in time could be identified where this production is more likely to occur. The method also allows for the investigation of the variation in timing and differences between listeners.

To overcome the complex recording setting of [18], Huang et al. [12] introduced Parasocial Consensus Sampling (PCS). With this method, human observers watch a video of a conversational partner and act as if they were in the conversation. Every time they would produce a social signal, they are to press a button. Despite the fact that the observers are not part of the conversation and pressing a button seems artificial, the results of PCS in terms of the quantity and timing of social signals was comparable to those produced by the actual subjects in the corpus. For social behavior synthesis, increased generalization was observed when considering as positive samples only the moments in time where the majority of the human observers indicated they would produce a social signal.

Both of the above methods address obtaining more positive samples, which reduces the moments in time where negative samples can be extracted. Still, there is no guarantee that a negative sample corresponds to a moment in time where the production of a social signal is inappropriate. To this end, Poppe et al. [26] had human observers watch a video of a speaker and an animation of a listener side-by-side. The listener was an IVA that produced specific social signals at predetermined moments in time. Motivated by the observation that humans are sensitive to flaws in animated social behavior, the human observers were instructed to press a button when they judged the produced social behavior as inappropriate. This approach was used as a subjective, perceptual evaluation measure for synthesized social behavior. However, it can also be used to obtain negative samples as we do in this research.

3 Iterative perceptual learning

We target a dyadic conversational setting where we aim at generating appropriate social signals for an IVA in real-time, based on the observed social behavior of a human conversational partner. We consider discrete social signals that (1) are performed as a reaction to the observable behavior of the conversational partner and (2) are more or less optional in nature. We further assume that the observations can be described as feature vectors. This allows us to use machine learning techniques that output a (probability) score for the production of a social signal based on a feature vector instance. These assumptions are common for learning social signal models [31]. Examples of this application setting are the animation of head movement as a reaction to the speech of the conversational partner, or the synthesis of a backchannel as a reaction to a speaker's speech and gaze. The latter example will be discussed in Sect. 4.

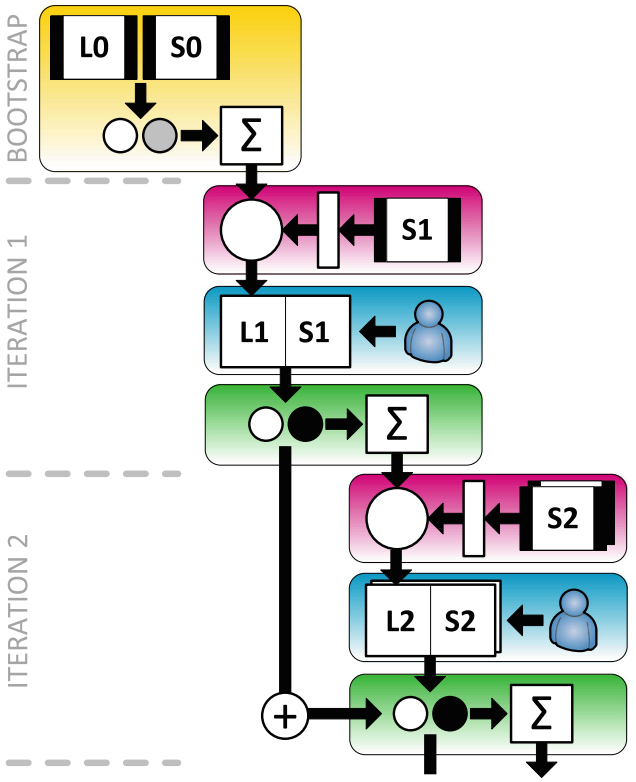

Fig. 2 Schematic representation of the Iterative Perceptual Learning framework. The generation, evaluation and learning stage are shown in *pink, blue and green*, respectively. *L* and *S* correspond to videos of the listener and speaker, respectively. The *numbers* correspond to the current iteration. The *summation sign* represents model learning, the *plus sign* stands for the addition of two sets of samples. *Small white and black circles* denote positive and negative samples, respectively. *Small gray circles* correspond to the feature instances sampled at random at non-positive moments. Please refer to the text for details. Best viewed in color

In this research, we learn social behavior synthesis models in an iterative, incremental manner. The basis is a machine learning model which we will treat as a black box. At each iteration, we train the model given the available positive and negative samples. As we cannot obtain negative samples from the corpus directly, we resort to an active learning approach. Here, human raters provide the labeling of samples that have been generated by the machine learning classifier [28]. Given that these samples are focused, the amount of labeling needed is typically lower compared to the labeling of all data samples. We apply our framework iteratively, so more training samples become available. We expect that we will generate social signals at more appropriate moments. Still, some of these instances will be perceived as inappropriate and these end up as negative samples for the next iteration. In our approach, we focus on obtaining such negative samples. We use a virtual, computer-animated, copy of the conversant and animate social signals according to a trained classifier. We then have human observers rate the (in)appropriateness

Fig. 3 Example stimulus presented to the participants during the evaluation

of the displayed social signals in the context of the conversation. Based on these ratings, we obtain negative samples which are used to train the models in the next iteration. Due to an increased number of available samples, both positive and negative, we expect that the models are progressively more accurate. The subjective ratings double as perceptual evaluation measures. This allows us to determine, at each iteration, the subjective quality of the generated listening behavior. This information allows us to determine when the learning saturates, so we can stop the training.

A schematic representation of the IPL framework appears in Fig. 2. It shows a bootstrap phase followed by two iterations. Each iteration consists of a generation, evaluation and learning stage, respectively. We discuss these in the following sections. We also address the bootstrapping of the approach. We consider a dialog with a sender and a receiver. Social behaviors will be synthesized for the receiver in response to features extracted from the behavior of the sender. However, the framework is general and can be used for learning any computational model for the synthesis of discrete nonverbal behaviors.

3.1 Generation

An iteration starts with the generation of the stimuli. Each stimulus is a video of the sender placed side-by-side with an animation of the receiver. See Fig. 3 for an example. There are three steps involved in the generation stage (the pink areas in Fig. 2): feature extraction, feature classification and stimulus generation.

The sender is observed, for example using microphone or camera. From these recordings, we obtain feature vectors at each time step. The sample rate is typically high to reduce the latency. Features can be audio features such as pitch and intensity, video features such as amount of movement or head orientation, or any combination of features.

We then classify each feature vector with the classifier that was trained in the learning step of the previous iteration, see Sect. 3.3 for details. This results in a numerical output, for example a probability score. We assume here that higher scores correspond to moments in time where the production

of a social signal is more appropriate. Given an entire video, we thus obtain a sequence of scores, one at each time instant.

The next step is to convert this sequence of scores into a set of discrete social signal timings. To this end, we can apply a threshold or select the moments corresponding to the top n scores. Additional constraints such as minimum time between two social signal timings, or a minimum or maximum number of social signals per minute can be enforced at this stage as well. Given the determined timings of the social signals, the resulting behavior is animated on an IVA. Here, the occurrence of a signal is translated into an animation, e.g. a head movement or backchannel. Finally, we place the animation of the receiver side-by-side with the video of the sender and make sure both are synchronized in time.

3.2 Evaluation

In the evaluation stage (blue areas in Fig. 2), human observers rate the inappropriateness of the animated social signals. Similar to [16,26], human raters watch the stimuli and press a button (the *yuck* button) whenever they consider an animated social signal of the receiver inappropriate.

After watching and rating a stimulus (i.e., a video of the speaker and an animation of the listener), the raters' yucks are matched to the animated social signals. When several raters watch the same stimuli, their yucks can be aggregated. This results in a percentage of raters that judge a certain social signal instance as inappropriate. These numbers can be thresholded to filter out accidental mis-presses and determine which social signal instances are to be considered negative samples. The instances that received no or only a few yucks can be regarded as positive samples, in addition to the social signals performed by the human listener in the recorded conversation.

3.3 Learning

A trained machine learning model is the result of the learning stage (green areas in Fig. 2). In this stage, all positive and negative samples are used to train the classifier. As mentioned before, the specifics of the classifier are not important at this point. In each iteration (except for the first, as we discuss below), the positive and negative samples are added to those of the previous iteration. There is thus an increasing number of samples available for training at each subsequent iteration.

3.4 Bootstrap

As we do not have access to negative samples in the first iteration, we bootstrap the process by learning a model from a limited amount of training data with negative samples extracted at random moments where no positive samples occur. This is the exact same approach as is typical for corpus-based learning [20]. After the generation and evaluation phases, we then obtain positive and negative samples, which are then used at each following iteration. The initial samples of the bootstrap phase are discarded after the first iteration, see also Fig. 2.

4 IPL for the timing of backchannels

To illustrate the use of the IPL framework for social behavior synthesis, we target the scenario of a face-to-face conversation between a speaker and a listener. In this setting, the listener is to signal continued attention, interest and understanding to the speaker [6], for example with a nod, a short vocalization ("uh-huh") or a smile. These social signals are commonly referred to as backchannels [35] or listener responses [34]. Our aim is to learn computational models to synthesize listening behavior, conditioned on the observed behavior of a human speaker [11]. Specifically, we predict the timing of backchannels in these speaker–listener dialogs.

We present an experiment in which we learned a backchannel prediction model for the listener using the (*IPL*) approach and compare this to the (*baseline*) with a classifier learned using the standard corpus-based approach. We evaluate the influence of several factors on both the objective and perceptual quality of the models. In the following, we will explain the data on which the models are learned and evaluated. Subsequently, the two models and experimental setup are presented. The results and discussion of the experiment appear in Sect. 5.

4.1 Corpus

We used the MultiLis corpus [17] for the training and evaluation of our synthesis models. The corpus consists of Dutch-spoken mediated human–human interactions between pairs of subjects. In the first interaction, one subject assumed the role of speaker and one subject was assigned the role of listener. In a second interaction, the roles were switched. In total, 32 subjects (29 male, 3 female, mean age 25) participated in 32 recordings, with a total duration of 131 min.

The speakers were instructed to either summarize a short video or to provide the instructions of a recipe they had just studied. Listeners had to remember as many details as possible. Subjects interacted through a remote videoconferencing system. The camera was placed behind an interrogation mirror on which the other subject was projected. This allowed subjects to look directly at the camera and this created the feeling of eye contact. In addition, this setting allowed us to analyze gaze.

4.2 Feature preprocessing

From the audio and the video, we extracted three types of features: *acoustic* (112 features), *speaking* (1 feature) and

looking (1 feature). Acoustic features have been used extensively. Backchannels are typically produced after the completion of a speaker's sentence or grammatical clause [7]. Often, the pitch rises or drops at these moments [2,32]. In addition, these endings are often followed by a short pause [4,30]. Several authors have found that the production of backchannels is also cued by a short moment of mutual gaze [1,8].

From each speaker's audio channel, we extracted acoustic features pitch, intensity and the first 12 mel-frequency cepstrum coefficients (MFCC) every 10 ms using OpenEAR [9]. We expect that these features are informative of the acoustic properties of the speech. The search for a set of features that can be extracted in real-time and yields good results for the prediction of backchannel timings remains an open research question that will not be addressed in this research. The high framerate ensured that the latency was minimal. Pitch detection is typically noisy and can fail for a few frames during speech. To solve this issue, we linearly interpolated the pitch values for gaps smaller than 8 frames, which is in line with [32]. Between subjects, acoustic signals can vary significantly. For instance, pitch is higher in females than in males, people speak with different volume and/or had the microphone closer to their mouth. We normalized these signals to account for these differences between speakers by converting each signal into the z-score equivalent. The means and standard deviations needed for calculating the z-score were obtained from the first 10 s of each session, which were excluded from the training data.

As we assume that a classifier is applied to each frame of data independently (see Sect. 4.4.3), we need to capture the temporal aspect to some extent. To this end, we calculated the mean and the slope of each signal over a period of 50, 100, 200 and 500 ms prior to the onset of a backchannel. As such, we expect that the behavior that cues backchannels is captured. The slope was calculated by fitting a first-order polynomial to the signal.

The *speaking* feature indicates if and for how long the speaker is talking and is extracted using the SHoUT automatic speech recognizer [14]. The *looking* feature indicates if and for how long the speaker is looking at the listener and is based on the manual annotations provided with the MultiLis corpus. Both signals are initially binary. To represent sequentiality, we calculated the relative offset to the moment where the speaker starts talking, respectively starts looking at the listener. Specifically, this offset is positive during the speaker's speech and negative in a pause. For example, a value of 5 means that the speaker started an utterance 5 frames before. A value of -5 is assigned when the speaker stopped talking 5 frames before. As speaking and not speaking are obviously correlated, we decided to use a single feature to represent the offset. For looking, a similar processing was applied.

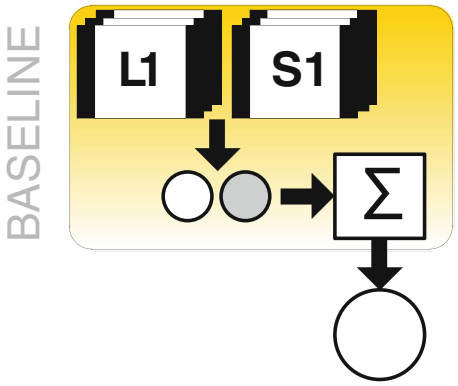

Fig. 4 Schematic representation of the Baseline approach. Please refer to the text for details

In summary, we extracted 14 acoustic signals, calculated their z-scores and obtained their means and slopes for four different window lengths. This resulted in a total of 112 acoustic features. In addition, we used one speaking feature and one looking feature. We concatenated all features into a 114-dimensional vector per time instance.

4.3 Baseline model

The baseline model (see Fig. 4) represents the common corpus-based approach for social behavior synthesis. Feature vectors together with their ground truth labels are presented to a classifier. The classifier learns a computational model that approximates the ground truth labels. In our experiment, we used the Support Vector Machine (SVM). SVMs are commonly used in social signal processing and associate a score to each input feature vector. We are interested in the relative performance of both approaches and do not focus on obtaining an optimally performing model. Therefore, we used the default settings of the libSVM library [5] without optimization of the parameters involved. These settings are an RBF kernel with $c = 1$ and $\gamma = 1/|x|$, where $|x|$ is the dimensionality of the input vector. Note that machine learning models that take into account the temporal nature of the input (e.g., generative models such as hidden Markov models, or discriminative models such as conditional random fields) are more suitable, but typically require more training data. We therefore used the well-understood SVM in our experiment.

Samples are labeled either positive or negative. Positive samples correspond to annotated backchannels in the corpus. Feature vectors were extracted from the window prior to the onset of the backchannel. The negative samples were randomly selected from moments where no backchannel was annotated in the corpus. Due to the optionality of backchannels, they possibly included false negatives. Typically, there is only a small number of positive samples available in a corpus. To increase the amount of training data and to make

the models less dependent on single frames, we selected four additional frames around the positive frame. We sampled these frames from a normalized Gaussian distribution with a σ such that 95 % of the samples falls within 250 ms of the positive sample. While this interval is somewhat arbitrary, corpus analysis revealed that this margin is typical for the production of backchannels [30]. Finally, we made sure that we selected an equal number of negative samples.

After training the SVM, we applied it to each input vector, taken at 10 millisecond intervals, to obtain backchannel timing predictions. We used the numerical decision values, which can be regarded as confidence scores for the synthesis of a backchannel. By sequencing these decision values over time, we obtained curves representing the appropriateness to provide a backchannel. To remove artifacts due to the potentially highly non-linear output of the SVM, we smoothed these curves with a 10 frame moving average. After this filtering, we considered the highest peaks in this curve to correspond to the most likely moments to predict a backchannel. A threshold was used to determine at which peaks a backchannel should be synthesized for the listener, similar to [21].

4.4 Iterative perceptual learning model

The IPL model is learned according to the framework presented in Sect. 3. We will explain the design decisions for each of the steps generation, evaluation and learning.

4.4.1 Generation

For each stimulus video, we used the Elckerlyc platform [33] to synthesize head nods as backchannels for the listener at the timings predicted by the trained SVM. To control for the number of backchannels, we determined the mean backchannel rate over all interactions in the corpus, which was approximately 7.7 per minute. We decided to generate 25 % extra backchannels (corresponding to a rate of 9.6) with the aim of potentially collecting more negative samples to be used in subsequent iterations. Based on these numbers, we determined the value of the threshold for the peak selection per stimulus video. The only restriction applied was that two backchannels could not be within 2 s from each other. Stimuli were obtained by putting side-by-side the video of the actual speaker and the animation of the virtual listener, see Fig. 3.

4.4.2 Evaluation

Each stimulus was evaluated perceptually by a number of participants in the experiment. Participants had to press the yuck button whenever they perceived an individual backchannel from the virtual listener as inappropriate. To account for response time, we matched these presses to the last preceding backchannels that occurred within 5000 ms of the onset. We determined for each synthesized backchannel the number of yucks, which we used as a measure of inappropriateness of the backchannel.

4.4.3 Learning

We used the exact same machine learning classifier as for the baseline model. The only difference was the way the negative samples were selected. Instead of randomly selecting negative samples, we used the timings of the generated backchannels which were yucked during the evaluation of the previous iteration as negative samples. Again, we balanced the number of positive samples and number of negative samples. The number of positive samples was multiplied by five, in line with the baseline model. Next, we calculated the sampling factor for the negative samples. We determined this factor by dividing the number of positive samples by the number of individual yucks. Backchannels that were yucked multiple times, were added as multiple negative samples. The sampling of both the positive and negative samples was performed in the same way as in the baseline model, using a normalized Gaussian distribution.

4.5 Experiment

The experiment for the prediction of backchannel timings compares IPL to the baseline approach. For IPL, we used four iterations after bootstrapping. At each iteration of the IPL model, we learned a model using the baseline approach to allow for comparison between the two approaches. After each phase, we evaluated the results of the IPL and baseline models using both objective and subjective measures.

4.5.1 Stimuli

Participants of the experiment were shown a video of a speaker from the MultiLis corpus side-by-side with an animated listener, see Fig. 3. The virtual listener nodded her head when the synthesis model predicted a backchannel. Other behaviors such as head movement, posture shifts, facial expressions and eye blinks were not animated to prevent these factors to contribute to the perception. As a result, the synthesized listening behavior was completely controlled, but rather minimal. For each interaction in a set we created an animation of the virtual listener based on the IPL model and a virtual listener based on the baseline model. The mean duration of a stimulus video was approximately 4 min, depending on the interaction between the actual speaker and listener in the corpus.

Table 1 Overview of the sets used in each iteration and each phase of the IPL process

Phase	Sets for training	Number of interactions	Sets for evaluation	Number of interactions
Bootstrap	Bootstrap set	1	Set 1	1
Iteration 1	Set 1	1	Set 2	2
Iteration 2	Sets 1, 2	3	Set 3	3
Iteration 3	Sets 1, 2, 3	6	Set 4	6
Iteration 4	Sets 1, 2, 3, 4	12	Test set	6

4.5.2 Procedure

The experiment consisted of five phases. We started with a bootstrap phase, followed by four iterations of IPL. In the bootstrap phase, a model was learned on a single interaction. This model was then evaluated perceptually on one other interaction, see the first row in Table 1.

For all but the first iteration, positive and negative samples obtained from all previous iterations were used to learn the IPL model. Given that the negative samples were selected at random in the bootstrap phase, all samples of this phase were discarded for the first iteration. Given that the evaluation results for the IPL model doubled as negative samples for model learning, there were more positive and negative samples available to learn the IPL model in each subsequent iteration (see also Fig. 2). In addition, we used a larger set of stimuli for evaluation. An overview of the number of stimuli used for learning and evaluation is given in Table 1.

To compare the performance of IPL with the baseline approach, we also perceptually evaluated the performance of the baseline approach after each iteration. We learned models on the same interactions according to Table 1, but with negative samples selected randomly without overlapping with positive samples, as explained before. As both models were trained on the same interactions and rated by the same participants, we can make a fair comparison. To this end, we perceptually evaluated the resulting IPL and baseline models on a test set of six interactions. The data of the test was never used for training.

Participants of the experiment were shown stimuli through a webpage. It was explained to them that they would be participating in an experiment to determine the quality of synthesized listening behavior. After entering their name, gender and age, the participants were presented a set of (at most) 6 stimuli. They were asked to press the spacebar each time the virtual listener performed a backchannel they judged as inappropriate. In principle, this required them to simultaneously monitor the behavior of the speaker and that of the listener. In practice, and in line with [26], this did not appear to be problematic. Participants could replay the stimulus from the start, which would discard all previously issued yucks for that stimulus. Each participant was shown the same interaction twice: once with the virtual listener based on IPL, once based on the baseline model. The order of the stimuli was varied systematically. The within-subject design allowed us to evaluate the difference between the two models pair-wise. This is essential as there are typically differences in the amount of yucks between participants. An experiment session lasted around 30 min.

4.5.3 Participants

Each stimulus was rated by five participants. Set four and the test set contained six interactions (12 stimuli), so we decided to split these sets into two. Including the evaluation on the test set for iterations 1–4, this gave us 13 groups of stimuli. Consequently, we required 65 participants to rate the stimuli, 25 for the evaluation of sets 1–4 and 40 for the evaluation of the test sets. Participants were recruited among colleagues and students. Several persons participated more than once. As we used a within-subjects design, this does not bias the comparison between the models. Of the 65 trials, 8 and 57 were completed by females and males, respectively (mean age 28, min. 18, max. 47).

4.5.4 Evaluation measures

We used both objective and subjective performance measures. For the objective measure, we compared the predicted timing of the backchannels with those performed by the actual listener in the MultiLis corpus, as is common for corpus-based learning. A backchannel was regarded as matching as it was predicted within 500 ms (before or after) a backchannel produced in the corpus. We calculated the precision p and recall r. Precision is the amount of matches amongst all predicted signals, recall is the amount of matches amongst all relevant instances in the corpus. We combined these into a single score by taking the F_1 measure, a weighted harmonic mean of the two: $F_1 = \frac{2 \times p \times r}{p + r}$. For the subjective measure, we used the yucks collected in the perceptual evaluation. We calculated the percentage of backchannels that did not receive any yucks. In addition, we calculated the average number of yucks per backchannel.

5 Results and discussion

Each stimulus was rated by five human observers. In total, this amounted to 24 h of annotated dialog. First, we compare the performance of both models on the test set after the final iteration. On the objective measure, both approaches perform the same with F_1 scores of 0.323. Direct comparison with other works is hindered by differences in the conversation type, models used and the margins for which pre-

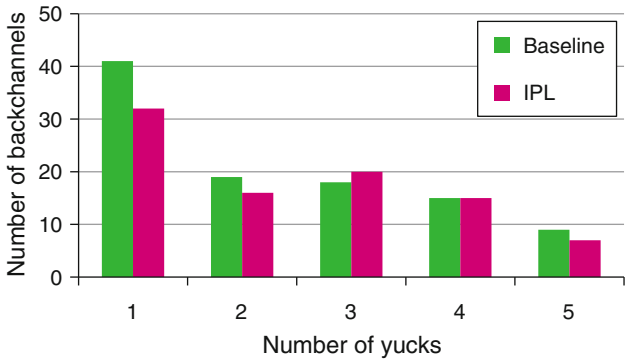

Fig. 5 Frequency histogram for number of yucks per synthesized backchannel on the test set, for baseline and IPL models after iteration four

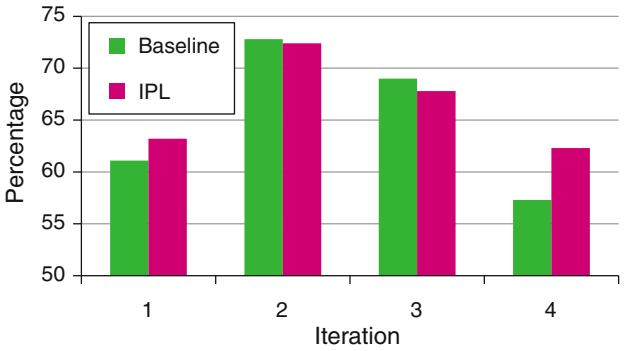

Fig. 6 Percentage of synthesized backchannels that did not receive any yucks on the test set, for the baseline and IPL models after each iteration

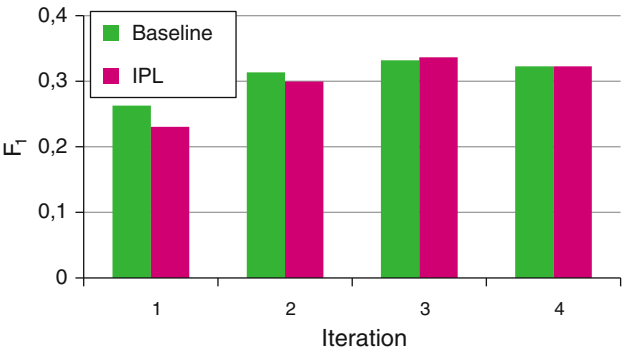

Fig. 7 F_1 measure on the test set, for the baseline and IPL models after each iteration

dicted backchannels are considered matching. In a comparable setting, De Kok et al. [18] achieved an F_1 score of 0.265.

The subjective measures show a slightly different effect. In total, 239 backchannels were generated with each of the models. The number of yucks obtained from five participants per stimulus is lower for IPL than for the baseline (219 and 238, respectively). A pair-wise t-test shows a tendency that the number of yucks per stimulus is lower for IPL, although this difference is not significant ($t(5) = -1.516$, $p = 0.09$). A larger number of stimuli could have made this difference more apparent. On average, a backchannel synthesized with IPL received 0.92 yucks from all participants, whereas this number was 1.0 for a backchannel generated from the baseline model. A breakdown of the number of yucks per backchannel is given in Fig. 5. Most of the generated backchannels received a modest number of yucks.

The number of backchannels that did not receive any yucks is higher for the IPL model, 149 (62.3 %) compared to 137 (57.3 %) for the baseline model. This finding is important as none of the participants judged these backchannels as inappropriate. Ideally, this would be the case for all backchannels generated by a synthesis algorithm. In conclusion, both models generate behavior that approximates that of the listener in the corpus in terms of co-occurring backchannels, but the behavior generated based on the IPL model is perceived as slightly more natural. In the following, we look at the amount of available data on the subjective and objective performance, and at the variation of the performance on different sets.

5.1 Effect of amount of data

Typically, as more data becomes available for training, one would expect that the performance of the resulting learned model improves. Models typically generalize better when trained on a wider variety of positive and negative samples. In addition to the IPL model, we learned a corresponding baseline model trained on the same conversations of the corpus. Both models were trained on the same positive samples but with different negative samples. After each iteration, we tested both models on the test set. The results of the evaluation are shown in Fig. 6 for the percentage of backchannels that did not receive a yuck. A couple of observations can be made. First, the performance is not monotonically increasing for an increasing number of available training interactions. Even though these numbers are obtained on the same set of interactions, they are not completely comparable as they are obtained from ratings of different participants. A within-subject design for iterations is needed to analyze whether there is a significant trend. Still, as model (IPL or baseline) was a within-subject factor, we can compare the results pair-wise. From Fig. 6, it becomes clear that the IPL models learned after the first and fourth iterations are perceived as more appropriate than those from the baseline approach.

For the objective F_1 measure, a positive trend can be observed for the amount of training data (see Fig. 7). However, in the final iteration, the scores are lower for both models. Apart from the first iteration, differences between the two models are small.

Table 2 The F_1 measures obtained for IPL/baseline models and evaluated on different sets

Model	IPL/baseline evaluated on			
	Set 2	Set 3	Set 4	Test
Iteration 1	0.165/0.154	0.206/0.222	0.152/0.216	0.230/0.263
Iteration 2	–	0.222/0.222	0.158/0.191	0.300/0.313
Iteration 3	–	–	0.189/0.222	0.336/0.332
Iteration 4	–	–	–	0.323/0.323

In sum, we found no evidence that an increasing amount of training data leads to better models. This might be due to two causes. First, the features used might not be sufficiently informative to clearly differentiate between appropriate and inappropriate moments to produce a backchannel. In our experiment, this might cause the performance of the models to saturate quickly. Second, there is typically a substantial variation between listeners in the amount and timing of backchannels [17]. We will investigate this in the following section.

5.2 Effect of variation in training set

To gain more insight into the variation in backchannel placement between training sets, we evaluated models trained after each iteration on all training sets of subsequent iterations. These tests are explicitly not part of the common IPL or baseline procedure, as we would be using test data for training. We calculated the F_1 measures for all combinations. Results are summarized in Table 2.

All models perform worse on set four. The listener's backchannel behavior or the backchannel-inviting behavior of the speaker in this set might differ from that in other sets. We expect this to cause the performance to drop in the final iteration. This can be explained as follows. Both models aim at learning a model for predicting backchannel opportunities, applicable to every speaker and listener. But individuals differ in their interaction styles and the models are not capable of attuning to each individual. During training, they converge to the behavior of an average speaker and an average listener. Apparently, the models are better at generalizing to the behavior of the first three sets, whereas the interactions in set four might deviate more from the average behavior.

6 Conclusion and future work

We introduced Iterative Perceptual Learning (IPL), a novel approach for learning computational models for social behavior synthesis. IPL takes an active learning approach by iteratively learning classifiers based on subjective, perceptual evaluations. Human observers rate the quality of synthesized behavior, based on the output of trained models. These ratings are given at the level of individual synthesized behaviors. Specifically, observers press a button to indicate that the behavior is inappropriate in the context of the conversation. By analyzing the ratings of several observers, we can measure the appropriateness of individual behavior instances. This subjective measure complements traditional objective measures such as precision and recall. In addition, the perceptual ratings are used to obtain negative samples for the subsequent training of the classifier. As such, the behavior synthesis model is refined iteratively, which allows us to tune our models to social behavior that is rated as appropriate.

We have demonstrated the application of IPL in a case study on the timing of backchannels in speaker–listener dialogs. We compared IPL to the traditional corpus-based approach. While both models performed similarly in terms of precision and recall, the results of the IPL model were rated as perceptually more appropriate. However, this difference was only marginally significant, but mainly due to the higher number of IPL backchannels without any negative ratings. Differences between IPL and the baseline approach were small and varied between sets of stimuli.

We expect that the features were not sufficiently informative to differentiate between appropriate and inappropriate moments to produce a backchannel. This might have caused the learning of the models to saturate quickly. Future work should address taking into account a larger amount of context and possibly other modalities (e.g., body motion and facial expressions). Furthermore, we consider the use of semantic and lexical features such as those utilized in [29] for the prediction of backchannel opportunities.

The SVM model might not have been the most suitable machine learning classifier as it is not a sequential model. Future work should address the use of temporal classifiers, especially those that can be attuned to different interaction styles.

For the sake of experimental control, our virtual listener only performed nods. No other behaviors were animated. We should investigate more realistic behavior. The experiment described in this paper used stimuli that consisted of a video of a speaker and an animation of a listener, which had to be observed simultaneously. Given the limited means for expression, this was not problematic. We expect that it will be more difficult to judge more elaborate and complex behavior of a virtual listener. Therefore, we propose to use an online setting, in which the observer is also the speaker. This would guarantee that the behavior of the listener is contingent with that of the virtual listener, while the observer (speaker) does not have to monitor her own behavior. Recently, we proposed the Switching Wizard of Oz, an experimental paradigm for the online evaluation of social behavior synthesis algorithms [25].

References

1. Bavelas JB, Coates L, Johnson T (2002) Listener responses as a collaborative process: the role of gaze. J Commun 52(3):566–580
2. Bertrand R, Ferré G, Blache P, Espesser R, Rauzy S (2007) Backchannels revisited from a multimodal perspective. In: Proceedings of auditory-visual speech processing. Hilvarenbeek, The Netherlands, pp 1–5
3. Bevacqua E, Hyniewska S, Pelachaud C (2010) Positive influence of smile backchannels in ECAs. In: Proceedings of the workshop on interacting with ECAs as virtual characters at the international joint conference on autonomous agents and multi-agent systems (AAMAS), Toronto, Canada
4. Cathcart N, Carletta J, Klein E (2003) A shallow model of backchannel continuers in spoken dialogue. In: Proceedings of the conference of the European chapter of the Association for Computational Linguistics, vol 1, Budapest, Hungary, pp 51–58
5. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27
6. Clark HH (1996) Using language. Cambridge University Press, Cambridge
7. Dittmann AT, Llewellyn LG (1967) The phonemic clause as a unit of speech decoding. J Personal Social Psychol 6(3):341–349
8. Duncan S Jr (1972) Some signals and rules for taking speaking turns in conversations. J Personal Social Psychol 23(2):283–292
9. Eyben F, Wöllmer M, Schuller B (2009) OpenEAR–introducing the Munich open-source emotion and affect recognition toolkit. In: Affective computing and intelligent interaction (ACII). Amsterdam, The Netherlands, pp 576–581
10. Gravano A, Hirschberg J (2009) Backchannel-inviting cues in task-oriented dialogue. In: Proceedings of interspeech. Brighton, UK, pp 1019–1022
11. Heylen D, Bevacqua E, Pelachaud C, Poggi I, Gratch J, Schröder M (2011) Generating listening behaviour. In: Cowie R, Pelachaud C, Petta P (eds) Emotion-oriented systems. Springer, Berlin, pp 321–347
12. Huang L, Morency LP, Gratch J (2010) Parasocial consensus sampling: combining multiple perspectives to learn virtual human behavior. In: Proceedings of autonomous agents and multi-agent systems (AAMAS), Toronto, Canada, pp 1265–1272
13. Huang L, Morency LP, Gratch J (2011) Virtual rapport 2.0. In: Proceedings of intelligent virtual agents (IVA). Reykjavík, Iceland, pp 68–79
14. Huijbregts M (2008) Segmentation, diarization and speech transcription: surprise data unraveled. Phd thesis, University of Twente
15. Kendon A (1967) Some functions of gaze direction in social interaction. Acta Psychol 26(1):22–63
16. de Kok I, Heylen D (2011) Appropriate and inappropriate timing of listener responses from multiple perspectives. In: Proceedings of intelligent virtual agents (IVA). Reykjavík, Iceland, pp 248–254
17. de Kok I, Heylen D (2011) The MultiLis corpus–dealing with individual differences of nonverbal listening behavior. In: Toward autonomous, adaptive, and context-aware multimodal interfaces: theoretical and practical issues. Springer, Berlin, pp 374–387
18. de Kok I, Ozkan D, Heylen D, Morency LP (2010) Learning and evaluating response prediction models using parallel listener consensus. In: Proceeding of international conference on multimodal interfaces and the workshop on machine learning for multimodal interaction (ICMI-MLMI). Beijing, China, p 3
19. Lee J, Neviarouskaya A, Prendinger H, Marsella S (2009) Learning models of speaker head nods with affective information. In: Proceedings of intelligent virtual agents (IVA). Amsterdam, The Netherlands, pp 1–6
20. Martin JC, Paggio P, Kuehnlein P, Stiefelhagen R, Pianesi F (2008) Introduction to the special issue on multimodal corpora for modeling human multimodal behavior. Lang Resour Eval 42(2):253–264
21. Morency LP, de Kok I, Gratch J (2010) A probabilistic multimodal approach for predicting listener backchannels. Auton Agents Multi-Agent Syst 20(1):80–84
22. Nishimura R, Kitaoka N, Nakagawa S (2007), A spoken dialog system for chat-like conversations considering response timing. In: Proceedings of text, speech and dialogue (TSD). Plzen, Czech Republic, pp 599–606
23. Okato Y, Kato K, Yamamoto M, Itahashi S (1996) Insertion of interjectory response based on prosodic information. In: Proceedings of the IEEE workshop on interactive voice technology for telecommunication applications. Basking Ridge, NJ, pp 85–88
24. Peters C, Pelachaud C, Bevacqua E, Mancini M, Poggi I (2005) A model of attention and interest using gaze behavior. In: Proceedings of intelligent virtual agents (IVA). Kos, Greece, pp 229–240
25. Poppe R, ter Maat M, Heylen D (2012) Online behavior evaluation with the Switching Wizard of Oz. In: Proceedings of intelligent virtual agents (IVA), Santa Cruz, CA, pp 486–488
26. Poppe R, Truong KP, Heylen D (2011) Backchannels: quantity, type and timing matters. In: Proceedings of intelligent virtual agents (IVA), pp. 228–239. Reykjavík, Iceland
27. Poppe R, Truong KP, Reidsma D, Heylen D (2010) Backchannel strategies for artificial listeners. In: Proceedings of intelligent virtual agents (IVA), pp 146–158
28. Tong S, Koller D (2001) Support vector machine active learning with applications to text classification. J Mach Learn Res 2:45–66
29. Traum D, DeVault D, Lee J, Wang Z, Marsella S (2012) Incremental dialogue understanding and feedback for multiparty, multimodal conversation. In: Proceedings of intelligent virtual agents (IVA), Santa Cruz, CA, pp 275–288
30. Truong KP, Poppe R, Kok I, Heylen D (2011) A multimodal analysis of vocal and visual backchannels in spontaneous dialogs. In: Proceedings of interspeech, Florence, Italy, pp 2973–2976
31. Vinciarelli A, Pantic M, Heylen D, Pelachaud C, Poggi I, D'Errico F, Schröder M (2012) Bridging the gap between social animal and unsocial machine: a survey of social signal processing. IEEE Trans Affect Comput 3(1):69–87
32. Ward N, Tsukahara W (2000) Prosodic features which cue back-channel responses in English and Japanese. J Pragmat 32(8):1177–1207
33. van Welbergen H, Reidsma D, Ruttkay ZM, Zwiers J (2010) Elckerlyc—a BML realizer for continuous, multimodal interaction with a virtual human. J Multimod User Interf 3(4):271–284
34. Xudong D (2009) Listener response. In: The pragmatics of interaction. John Benjamins Publishing, Amsterdam, pp 104–124
35. Yngve VH (1970) On getting a word in edgewise. In: Sixth regional meeting of the Chicago Linguistic Society, vol 6, pp 657–677