Predicting when to laugh with structured classification

Bilal Piot, Olivier Pietquin, Matthieu Geist

PREDICTING WHEN TO LAUGH WITH STRUCTURED CLASSIFICATION

Bilal Piot[1], *Olivier Pietquin*[2], *Matthieu Geist*[1]

[1] SUPELEC IMS-MaLIS research group and UMI 2958 (GeorgiaTech - CNRS)
[2]University Lille 1, LIFL (UMR 8022 CNRS/Lille 1) - SequeL team
[1]firstname.lastname@supelec.fr, [2]olivier.pietquin@univ-lille1.fr

ABSTRACT

Today, Embodied Conversational Agents (ECAs) are emerging as natural media to interact with machines. Applications are numerous and ECAs can reduce the technological gap between people by providing user-friendly interfaces. Yet, ECAs are still unable to produce social signals appropriately during their interaction with humans, which tends to make the interaction less instinctive. Especially, very little attention has been paid to the use of laughter in human-avatar interactions despite the crucial role played by laughter in human-human interaction. In this paper, a method for predicting the most appropriate moment for laughing for an ECA is proposed. Imitation learning via a structured classification algorithm is used in this purpose and is shown to produce a behavior similar to humans' on a practical application: the yes/no game.

***Index Terms*—** Laughter; Imitation Learning; Structured Classification

1. INTRODUCTION

Building efficient and user-friendly human-machine interfaces is a key challenge for the future of computer science, enabling a large public to interact with complex systems and reducing the technological gap between people. In the last decade, Embodied Conversational Agents (ECAs) emerged as such interfaces. Yet, their behaviour is still perceived as quite unnatural to users. One of the reasons of this bad perception is the inability of ECAs to make a proper use of social signals, although there exists some research on this topic [1]. Among these signals, laughter is a prominent feature used by humans during interactions. Yet, very little attention has been paid to enable ECAs with laughter capabilities until recently [2].

Enabling ECAs with laughter capabilities is not only about being able to synthesize audio-visual laughter signals [3, 4]. It is also concerned by an appropriate management of laughter during the interaction. There is thus a need for a laughter-enabled interaction manager, able to decide when to laugh so that it is appropriate in the conversation. This being said, it remains uneasy to define what is an appropriate moment to laugh.

More formally, the task of the laughter-enabled interaction Manager (IM) is to take decisions about whether to laugh or not. These decisions have to be taken according to the interaction context which can be inferred from laughter, speech and smile detection modules (detecting social signals emitted by the users) implemented in the ECA but also by the task context (for example, if the human is playing a game with the ECA, what is the status of the game). Formally, the IM is thus a module implementing a mapping between contexts (or states noted $s \in S$) and decisions (or actions noted $a \in A$). Let's call this mapping a policy, noted $\pi(s) = a$. This mapping is quite difficult to learn from real data as the laughs are quite rare and very different from one user to another.

In this paper, we describe the research results for learning such a mapping from data, recorded during some human-human interactions, so as to implement, in the IM, a behavior similar to the one of a human. An imitation learning method is thus adopted. Especially, structured classification is investigated and proven to efficiently learn a behavior similar to human users where the similarity between human and algorithms is measured via a new criterion called $Naturalness$ and defined in Sec. 5. In addition, we use a technique of boosting for the structured classification algorithm which makes it a non-parametric algorithm. This avoids the choice of meta-parameters. Finally, we test different imitation algorithms on data sets of real laughs in a natural interaction context which is the yes/no game described in Sec. 4.

2. IMITATION LEARNING

Describing the optimal behavior of the avatar is a very tricky task. It would require the perfect knowledge of rules prevailing to the generation of laughter by humans. Interpreting sources of laughter or predicting laughter from a cognitive or psychology perspective is non-trivial. Therefore, a data-driven method has been preferred here. Especially, learning by imitation seems the best suited framework to learn the IM policy. Indeed, humans are implementing such a policy and they can provide examples of natural behaviors.

Formally, in the learning by imitation framework, an artificial learning agent (here the IM) learns to behave optimally by observing some expert agent demonstrating the task. The expert is implementing an optimal policy noted π^E and the demonstrations provide a set of examples $\{s_i, a_i = \pi^E(s_i)\}_{\{1\leq i\leq N\}}$. The problem is thus to learn a policy $\hat{\pi}$ such that $\forall s, \hat{\pi}(s) \approx \pi^E(s)$ from the set of demonstrations.

One way to address the problem of imitation learning is to reduce it to a Multi-Class Classification (MCC) problem [5, 6, 7, 8]. The goal of MCC is, given a training set $D = (s_i \in S, a_i \in A)_{\{1\leq i\leq N\}}$ where S is a compact set of inputs (generally a compact set of $\mathbb{R}^n$) and A a finite set of labels, to find a decision rule $\pi \in A^S$ that generalizes the relation between inputs and labels. More formally, it consists in finding a decision rule $\pi \in H$, where $H \subset \mathbb{R}^{S\times A}$ is called the hypothesis space, that tries to minimize the following empirical risk:

$$T(\pi) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{\pi(s_i)\neq a_i\}}.$$

where $\mathbf{1}_{\{a\neq b\}} = 1$ if $a \neq b$ and 0 otherwise.

A large literature already exists about the MCC problem. Well known methods such as Classification Trees [9], K-Nearest Neighbors (KNN) [10] and Support Vector Machines (SVM) [11, 12] are widely used and statistically studied. In [5], the authors use an artificial neural network to learn a driving policy for a robotic vehicle. Neural nets are also used in [7] to learn to play video games (although the method is more generic and could use other MCC methods). KNN's where used in [2] in a similar application as the one described in this paper. In [6], structured classification [13] is used to learn a grasping control policy for a robotic arm.

3. STRUCTURED CLASSIFICATION FOR IMITATION LEARNING

In [6], the authors use a large margin approach which allows adding some prior (or structure) via a margin function in the classification method. That is why it is considered as a structured classification method. The large margin approach is a score-based MCC where the decision rule $\pi \in A^S$ is obtained via a score function $q \in \mathbb{R}^{S\times A}$ such that $\forall s \in S, \pi(s) \in \operatorname{argmax}_{a\in A} q(s,a)$. The large margin approach consists, given the training set D, in solving the following optimization problem:

$$q^* = \operatorname*{argmin}_{q\in\mathbb{R}^{S\times A}} J(q), \tag{1}$$

$$J(q) = \frac{1}{N}\sum_{i=1}^{N} \max_{a\in A}\{q(s_i,a) + l(s_i,a_i,a)\} - q(s_i,a_i),$$

where $l \in \mathbb{R}_+^{S\times A\times A}$ is called the margin function. If it is zero, minimizing $J(q)$ attempts to find a score function q^* for which the example labels are scored higher than all other labels. Choosing a nonzero margin function improves generalization [6]. Instead of requiring only that the demonstrated label is scored higher than all other labels, it requires it to be better than each label a by an amount given by the margin function. Thus, the margin function allows deciding which samples are required to be well classified by putting an important margin on this particular example compared to the others. The policy outputted by this algorithm would be $\pi(s) \in \operatorname{argmax}_{a\in A} \hat{q}(s,a)$ where $\hat{q}$ is the output of the minimization of $J(q)$. The advantages of this method are its simplicity and the possibility to change the margin that allows us to adapt to specific characteristics of the problem. In addition, in [14], the authors use a boosting technique to solve the optimization problem given by Eq. (1) which is advantageous. A boosting method is an interesting optimization technique: it minimizes directly the criterion in Eq.1 without the step of choosing features. As presented in [15], a boosting algorithm is a projected sub-gradient descent [16] of a convex functional (here J is convex relatively to the variable q) in a specific functions space (here $\mathbb{R}^{S\times A}$) which has to be a Hilbert space. Boosting algorithms use a projection step on a restriction set of functions when optimizing over functions space, because the functions representing the gradient are often computationally difficult to manipulate and do not generalize well to new inputs [15]. In boosting literature, the restriction set corresponds directly to the set of hypotheses generated by a weak learner. In our experiments, we choose as restriction set the set of classification trees with two classes.

4. EXPERIMENTAL SETUP

The yes/no game is one of the possible scenarios of an interaction between humans and avatars where laughter is involved. In this game, players must respond to questions without saying "yes" or "no". The experiment scenario we present in this article is illustrated in Fig. 2. Two users are sitting on one side of a table while a virtual agent projected on a large screen is placed on the opposite side of the table. The users start to play the yes/no game, one asking questions (e.g., "what's your nationality?", "are you sure?"), this user is named U_1,

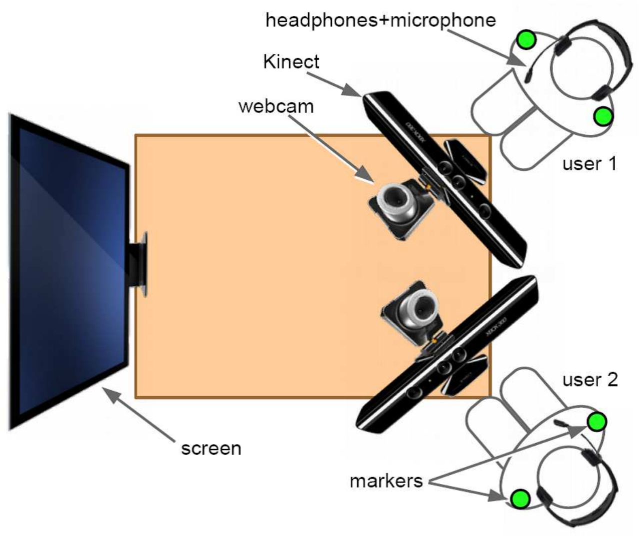

Fig. 1. Experimental setup.

and the other one answering trying to avoid to say "yes" or "no" (e.g., "I'm not sure" or "definitely"), this user is named U_2. The avatar, named A, participates to the interaction by laughing and asking questions. Of course, U_1 and A try to make U_2 to say "yes" or "no" and thus try to induce a loss of self-control. At any point, laughter can occur for any participant. The avatar has to generate laughter at appropriate moments given its perception of the context.

As shown in Fig. 2, detection of humans' laughter is performed through body (Kinect and body markers), face (Kinect) and speech (head mounted microphones) analysis [17]. Several recognition algorithms are executed in real-time to determine users' expressivity of motion.

In order to train our avatar by an imitation learning algorithm, several experiments are first recorded, where the avatar (symbolized by a screen in Fig. 2) is replaced by a human playing the role of the avatar (this is the expert we want to imitate). The same detection material as for the two other participants is used for the human playing the role of the avatar. Thanks to those recordings an expert data set $D = \{s_i, a_i = \pi^E(s_i)\}_{\{1\leq i\leq N\}}$ is generated which is the input of an imitation learning algorithm. Indeed, for each user (U_1 and U_2), the recognition algorithms are able to extract each 0.5s 4 features which are real values between 0 and 1. The 4 features are the probability of speech, the probability of laughter, the intensity of laughter and the probability of smile. Moreover, another feature, which represents the context of the game, is added by annotation of the recordings: 0 when the game is currently ongoing and 1 when it ends (that is when U_2 said "yes" or "no" or that some time-out occurred). Thus each 0.5s, we are provided 9 features (4 features for U_1, 4 features for U_2 and the context) that represents the state of the game s_i. Finally, by annotations of the recordings, we provide each 0.5s a binary information (1 or 0) giving the decision of the expert (a_i): `laugh`/`no_laugh` (so it is a 2 actions decision process). A sample (s_i, a_i) where $a_i = 0$ corresponds to a `no_laugh` sample and a sample (s_i, a_i) where $a_i = 1$ corresponds to a `laugh` sample. In addition, we also collect, by annotations, the binary `laugh`/`no_laugh` information for U_1 and U_2 : $(a_i^{U_1}, a_i^{U_2})_{\{1\leq i\leq N\}}$. Now that we have the expert data set, it is possible to use it as an input to different imitation learning algorithms.

Fig. 2. Real Demo.

5. RESULTS

In this section, we present the results obtained by applying different imitation learning algorithms to the expert data set. We use 4 different algorithms, 3 classical classification algorithms, which are KNN, Classification Tree and SVM, and the large-margin algorithm presented in Section 3. The KNN algorithm was previously used in [2] where $K = 1$, here we do the same choice in order to compare to other methods. The SVM algorithm uses a Gaussian kernel with a standard deviation $\sigma = 1$ and the Classification Tree is a pruned binary classification tree. For the large margin approach, we choose a margin with a particular structure that favors the `no_laugh` samples more than the `laugh` samples so as to only synthesized laughter when it is really appropriate. Indeed, laughing at inappropriate moments seems awkward for humans and it is important to avoid that while not laughing is not too problematic in this application. Thus, we choose the following margin structure:

$$\begin{aligned}
&l(s_i, a_i, a) = 0 \text{ if } a = a_i, \\
&l(s_i, a_i, a) = 6 \text{ if } a \neq a_i \text{ and } a_i = 0, \text{ (no laugh)} \\
&l(s_i, a_i, a) = 1 \text{ if } a \neq a_i \text{ and } a_i = 1, \text{ (laugh)} .
\end{aligned}$$

Eighteen minutes of recordings were collected in three sessions where the game was played several times (at least twice by recordings). This provided an expert data set $D = \{s_i, a_i = \pi^E(s_i)\}_{\{1\leq i\leq N\}}$ of 2285 examples (that is the number of 0.5s frames). The 4 algorithms were trained on this data set. In order to compare the performances of the algorithms, we use a P-fold cross validation. In P-fold cross-validation, the original data D is partitioned into P equal size sub-samples $D = (D_p)_{\{1\leq p\leq P\}}$, where $D_p = \{s_{j,p}, a_{j,p} = \pi^E(s_{j,p})\}_{\{1\leq j\leq N_p\}}$ and $\sum_{p=1}^{P} N_p = N$. Of the P sub-samples, a single sub-sample is retained as the validation data for testing the algorithm, and the remaining $P-1$ sub-samples are used as training data. The cross-validation process is then repeated P times (the folds), with each of the P sub-samples used exactly once as the validation data. The P results from the folds then can be averaged to produce a single estimation. For each

Algorithms	Global	`laugh`	`no_laugh`
Large Margin	**0.6871**	0.3256	**0.8081**
SVM	0.6723	0.3166	0.7893
KNN	0.5440	**0.6347**	0.5173
Tree	0.5570	0.5166	0.5732

Table 1. Classification rates.

sub-sample D_p and each algorithm `algo`[1], we define the policy $\pi_p^{\texttt{algo}} \in A^S$ learned on the remaining $P-1$ sub-samples. In addition, we define, for each sub-sample D_p, the number of `laugh` samples $N_p^{\texttt{laugh}} = \sum_{j=1}^{N_p} \mathbf{1}_{\{a_{j,p}=1\}}$ and the number of `no laugh` samples $N_p^{\texttt{no\_laugh}} = N_p - N_p^{\texttt{laugh}}$. Several quality evaluation criteria were used for each algorithm. The first criterion is the mean over the P folds of the global classification rate:

$$\frac{1}{P}\sum_{p=1}^{P}\frac{1}{N_p}\sum_{j=1}^{N_p}\mathbf{1}_{\{\pi_p^{\texttt{algo}}(s_{j,p})=a_{j,p}\}}.$$

The second criterion is the mean over the P folds of the classification rate on `laugh` samples:

$$\frac{1}{P}\sum_{p=1}^{P}\frac{1}{N_p^{\texttt{laugh}}}\sum_{j=1}^{N_p}\mathbf{1}_{\{\pi_p^{\texttt{algo}}(s_{j,p})=a_{j,p}\}}\mathbf{1}_{\{a_{j,p}=1\}}.$$

The third criterion is the mean over the P folds of the classification rate on the `no_laugh` samples:

$$\frac{1}{P}\sum_{p=1}^{P}\frac{1}{N_p^{\texttt{no\_laugh}}}\sum_{j=1}^{N_p}\mathbf{1}_{\{\pi_p^{\texttt{algo}}(s_{j,p})=a_{j,p}\}}\mathbf{1}_{\{a_{j,p}=0\}}.$$

We choose those different criteria in order to see the quality of each algorithm on the `laugh` samples and the `no_laugh` samples because those two classes are not well balanced (basically there is 5 times more `no_laugh` samples than `laugh` samples). In Table 1, we have the results of the different algorithms in terms of classification rates with $P = 5$.

The Large Margin has the best results for the global classification rate and the `no_laugh` rate. The structure of the margin favors the performance on the `no_laugh` samples and it is reflected in the results. KNN works well on the `laugh` samples which is also the case of the Classification Tree but has a really poor global performance. It seems that the avatar is too reactive (laughs too often) which can be problematic if the laughs happen on inappropriate moments: this behavior appears unnatural. In order to check if the good performance on laughs of KNN is due to the fact that it is too reactive, we computed the number of laughs produced for each fold and take the mean. Results are provided in Table 2.

The Classification Tree and the KNN avatar are too reactive which can explain their good performance on laughs but their behavior is not natural compared to the expert. The most natural behavior is the one produced by the Large Margin algorithm which laughs in the same proportion than the expert. So the classification rates are not appropriate measures to assess the algorithms according to this application.

For this reason, we came up with a measure for naturalness which indicates if the policy produced by the algorithm corresponds

[1]The variable `algo` can take the values KNN, SVM, LargeMargin and Tree.

Algorithms	Number of laughs in average
Expert	11.6
Large Margin	**15.4**
SVM	17.4
KNN	35.4
Tree	25.2

Table 2. Comparison of laughs numbers.

to the behavior of the expert. The idea is to compare if relatively to the two other users the human playing the avatar and the algorithm have the same behavior.

In order to see if there is a similarity between the behavior of the user playing the avatar A_{expert} and the one learnt by the algorithms $A_{\texttt{algo}}$, we check if the behavior of the expert A_{expert} compares to the users $((U_q)_{\{q=1,2\}})$ similarly to the way the avatar's behaviour $A_{\texttt{algo}}$ compares to the users $((U_q)_{\{q=1,2\}})$. The idea is to show that the avatar doesn't differ more from U_1 and U_2 than the expert does. To do so, for each user U_q and each sub-sample D_p, we define the number of `laugh` samples $N^{\text{laugh}}_{p,U_q} = \sum_{j=1}^{N_p} \mathbf{1}_{\{a^{U_q}_{j,p}=1\}}$ and the number of `no_laugh` samples $N^{\text{no\_laugh}}_{p,U_q} = N_p - N^{\text{laugh}}_{p,U_q}$. Three criterions were used: the global rate, the `laugh` rate and `no_laugh` rate. The global criterion rate$^1_{\texttt{avatar}}$ is the rate of agreement in terms of actions between one of the user and an `avatar`[2] sample by sample:

$$\frac{1}{2}\sum_{q=1}^{2}\frac{1}{P}\sum_{p=1}^{P}\frac{1}{N_p}\sum_{j=1}^{N_p}\mathbf{1}_{\{\pi_p^{\texttt{avatar}}(s_{j,p})=a^{U_q}_{j,p}\}},$$

where $\pi_p^{Expert}(s_{j,p}) = \pi^E(s_{j,p}) = a_{j,p}$. The `laugh` criterion rate$^2_{\texttt{avatar}}$ gives the rate of agreed laughs between the avatar and one of the users:

$$\frac{1}{2}\sum_{q=1}^{2}\frac{1}{P}\sum_{p=1}^{P}\frac{1}{N^{\text{laugh}}_{p,U_q}}\sum_{j=1}^{N_p}\mathbf{1}_{\{\pi_p^{\texttt{avatar}}(s_{j,p})=a^{U_q}_{j,p}\}}\mathbf{1}_{\{a^{U_q}_{j,p}=1\}}.$$

The `no_laugh` criterion rate$^3_{\texttt{avatar}}$ gives the rate of agreed no laughs between the avatar and one of the users:

$$\frac{1}{2}\sum_{q=1}^{2}\frac{1}{P}\sum_{p=1}^{P}\frac{1}{N^{\text{no\_laugh}}_{p,U_q}}\sum_{j=1}^{N_p}\mathbf{1}_{\{\pi_p^{\texttt{avatar}}(s_{j,p})=a^{U_q}_{j,p}\}}\mathbf{1}_{\{a^{U_q}_{j,p}=0\}}.$$

In order to have a single number representing the similarity between the expert avatar A and the avatars outputted by the algorithms , a new criterion, called $Naturalness$ $N_{\texttt{algo}}$, is defined as follows:

$$N_{\texttt{algo}} = \prod_{i=1}^{3}\frac{\min(\text{rate}^i_{\texttt{algo}}, \text{rate}^i_{Expert})}{\max(\text{rate}^i_{\texttt{algo}}, \text{rate}^i_{Expert})}$$

This criterion is thus a measure of the deviation between the behavior of the expert avatar and the behavior learnt by a given algorithm. If the $Naturalness$ is equal to 1, it means that the avatar has the same behavior as the expert relatively to the other users and if it is equal to zero, it means that the avatar has a completely different behavior than the expert.

Table 4 gives the results. The Large Margin method clearly outperforms the other ones, which means that its behavior relatively to the other users corresponds closely to the one of the expert. We see that the KNN and the Tree have poor $Naturalness$ as they laugh too much relatively to the other users which is not what the expert does.

[2]The variable `avatar` can take the values Expert, KNN, SVM, Large-Margin and Tree

Algorithms	Global rate	Laugh rate	No Laugh rate
Expert	0.7079	0.4503	0.7649
Large Margin	0.7139	0.4286	0.7287
SVM	0.7183	0.5136	0.7756
KNN	0.5096	0.8115	0.4407
Tree	0.5285	0.5858	0.5163

Table 3. Rates used for $Naturalness$.

Algorithms	$Naturalness$
Expert	1
Large Margin	**0.9222**
SVM	0.8762
KNN	0.2319
Tree	0.3874

Table 4. $Naturalness$.

6. CONCLUSION AND PERSPECTIVES

In this paper, a method for learning when an avatar should laugh during an interaction with humans was presented. It is based on a data-driven imitation learning algorithm and especially on structured classification method. The structured margin implied in this method is used to weight the importance of laughter compared to silence so as to generate a more natural behaviour and deal with the unbalanced nature of data. It is shown, in a yes/no game setting, that the method outperforms other classification methods in terms of overall similarity with a human. Compared to previous experimentations [2], this method objectivelly provides better results in terms of a newly introduced criterion.

Here, imitation learning is reduced to a multiclass classification problem. Yet, imitation learning can also be solved by other methods such as inverse reinforcement learning [18, 19]. Actually, this method has been shown to work better for some types of problems [20] and has already been used to imitate human users in the case of spoken dialogue systems [21]. Therefore, we plan to extend this work to inverse reinforcement learning in the near future. Also, this method could be used to generate new simulation techniques for optimizing human machine interaction managers in other applications such as spoken dialogue systems [22, 23].

7. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 270780.

8. REFERENCES

[1] Marc Schröder, Elisabetta Bevacqua, Roddy Cowie, Florian Eyben, Hatice Gunes, Dirk Heylen, Mark ter Maat, Garry McKeown, Satish Pammi, Maja Pantic, Catherine Pelachaud, Björn Schuller, Etienne de Sevin, Michel Valstar, and Martin Wöllmer, "Building autonomous sensitive artificial listeners," *IEEE Transactions on Affective Computing*, vol. 3, no. 2, pp. 165–183, 2012.

[2] Radoslaw Niewiadomski, Jennifer Hofmann, Jérome Urbain, Tracey Platt, Johannes Wagner, Bilal Piot, Huseyin Cakmak, Sathish Pammi, Tobias Baur, Stéphane Dupont, Matthieu Geist, Florian Lingenfelser, Gary McKeown, Olivier Pietquin, and Willibald Ruch, "Laugh-aware virtual agent and its impact on user amusement ," in *Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Saint Paul, USA, May 2013, pp. 619–626.

[3] Radoslaw Niewiadomski, Sathish Pammi, Abhishek Sharma, Jennifer Hofmann, Tracey, Richard Thomas Cruz, and Bingqing Qu, "Visual laughter synthesis: Initial approaches," in *Proceedings of the Interdisciplinary Workshop on Laughter and other Non-Verbal Vocalisations*, Dublin, Ireland, October 2012, pp. 10–11.

[4] Jérôme Urbain, Hüseyin Cakmak, and Thierry Dutoit, "Evaluation of HMM-based laughter synthesis," in *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada, May 2013, pp. 7835 – 7839.

[5] Dean Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Advances in Neural Information Processing Systems (NIPS 1988)*, Vancouver, Canada, December 1988, pp. 305–313.

[6] Nathan Ratliff, J. Andrew Bagnell, and Siddhartha S. Srinivasa, "Imitation learning for locomotion and manipulation," in *Proceedings of the 7th IEEE-RAS International Conference on Humanoid Robots*, Pittsburg, USA, November 2007, pp. 392–397.

[7] Stéphane Ross and J. Andrew Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, Sardinia, Italy, May 2010, vol. 9 of *JMLR Workshop and Conference Proceedings*, pp. 661–668.

[8] Umar Syed and Robert E. Schapire, "A reduction from apprenticeship learning to classification," in *Advances in Neural Information Processing Systems (NIPS 2010)*, Vancouver, Canada, December 2010, pp. 2253–2261.

[9] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone, *Classification and regression trees*, Chapman & Hall, Monterey, CA, 1984.

[10] Thomas M. Cover and Peter E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[11] Vladimir Vapnik, *Statistical learning theory*, Wiley, 1998.

[12] Yann Guermeur, "A generic model of multi-class support vector machine," *Journal of Intelligent Information and Database Systems*, vol. 6, no. 6, pp. 555–577, October 2012.

[13] Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin, "Learning structured prediction models: a large margin approach," in *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, New York, NY, USA, 2005, pp. 896–903.

[14] Nathan Ratliff, David Bradley, J Andrew (Drew) Bagnell, and Joel Chestnutt, "Boosting Structured Prediction for Imitation Learning," in *Advances in Neural Information Processing Systems (NIPS 2007)*, Vancouver, Canada, December 2007.

[15] Alexander Grubb and J.Andrew Bagnell, "Generalized boosting algorithms for convex optimization," in *Proceedings of the 28th International Conference on Machine Learning (ICML2011)*, 2011.

[16] Naum.Z. Shor, Krzysztof.C. Kiwiel, and Andrzej Ruszcaynski, *Minimization methods for non-differentiable functions*, Springer-Verlag, 1985.

[17] Johannes Wagner, Florian Lingenfelser, Tobias Baur, Ionut Damian, Felix Kistler, and Elisabeth André, "The social signal interpretation (SSI) framework: multimodal signal processing and recognition in real-time," in *Proceedings of the 21st ACM international conference on Multimedia (MM 2013)*, New York, NY, USA, October 2013, pp. 831–834.

[18] Stuart Russell, "Learning agents for uncertain environments (extended abstract)," in *Proceedings of the eleventh annual conference on Computational learning theory (COLT' 98)*, Madison, Wisconsin, USA, July 1998, pp. 101–103.

[19] Edouard Klein, Matthieu Geist, Bilal Piot, and Olivier Pietquin, "Inverse reinforcement learning through structured classification," in *Advances in Neural Information Processing Systems (NIPS 2012)*, South Lake Tahoe, USA, December 2012, pp. 1016–1024.

[20] Bilal Piot, Matthieu Geist, and Olivier Pietquin, "Learning from demonstrations: Is it worth estimating a reward function?," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2013)*, Prague (Czech Republic), September 2013, vol. 8188 of *Lecture Notes in Computer Science*, pp. 17–32, Springer.

[21] Senthilkumar Chandramohan, Matthieu Geist, Fabrice Lefèvre, and Olivier Pietquin, "User simulation in dialogue systems using inverse reinforcement learning," in *Proceedings of the 12th Annual Conference of the International Speech Communication Association (Interspeech 2011)*, Florence, Italy, August 2011, pp. 1025–1028.

[22] Olivier Pietquin and Thierry Dutoit, "A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 2, pp. 589–599, March 2006.

[23] Olivier Pietquin and Helen Hastie, "A survey on metrics for the evaluation of user simulations," *Knowledge Engineering Review*, vol. 28, no. 01, pp. 59–73, February 2013.