Imitation Learning Applied to Embodied Conversational Agents

Bilal Piot, Matthieu Geist, Olivier Pietquin

Imitation Learning Applied to Embodied Conversational Agents

Bilal Piot
Univ. Lille-CRIStAL (UMR 9189)
SequeL team
bilal.piot@univ-lille3.fr

Matthieu Geist
CentraleSupelec-MaLIS research group
UMI GeorgiaTech-CNRS (UMI 2958)
matthieu.geist@centralesupelec.fr

Olivier Pietquin
Univ. Lille-CRIStAL (UMR 9189)
SequeL team
Institut Universitaire de France (IUF)
olivier.pietquin@univ-lille1.fr

Abstract

Embodied Conversational Agents (ECAs) are emerging as a key component to allow human interact with machines. Applications are numerous and ECAs can reduce the aversion to interact with a machine 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, methods for predicting when and how to laugh during an interaction for an ECA are proposed. Different Imitation Learning (also known as Apprenticeship Learning) algorithms are used in this purpose and a regularized classification algorithm is shown to produce good behavior on real data.

1 Introduction

An important challenge for the future of computer science is to build efficient and user-friendly human-machine interfaces. This will enable a large public to interact with complex systems and reduce considerably the technological gap between people. In the last decade, Embodied Conversational Agents (ECAs), also called avatars, emerged as such interfaces. However, their behavior appear quite unnatural to most users. One possible explanation of this bad perception is the inability of ECAs to make a proper use of social signals, even though there exists some research on this subject (Schröder et al., 2012). One of these signals is laughter, which is a prominent feature used by human during interactions. Yet, very little attention has been paid to enable ECAs with laughter capabilities until recently (Niewiadomski et al., 2013; Piot et al., 2014b).

Enabling ECAs with laughter capabilities is not only about being able to synthesize audio-visual laughter signals (Niewiadomski et al., 2012; Urbain et al., 2013). It is also concerned by an appropriate management of laughter during the interaction, which is a sequential decision-making problem. Thus, there is a need for a laughter-enabled interaction manager, able to decide when and how 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 and to choose the good type of laugh.

More formally, the task of the laughter-enabled Interaction Manager (IM) is to take decisions about whether to laugh or not and how. 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). This is a sequential decision-making problem. Thus, the IM is 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. Imitation Learning (IL), also known as Apprenticeship Learning (AL), methods are considered. Indeed, during some human-human interactions, one human can be considered as an expert which actions should be imitated by the IM. This can be framed as a Learning from Demonstrations (LfD) problem as LfD is a paradigm in which an agent (called the apprentice) learns to behave in a dynamical environment from demonstrations of another agent (named the expert). IL is a framework that

Appearing in Proceedings of the 4^{th} Workshop on Machine Learning for Interactive Systems (MLIS) 2015, Lille, France. JMLR: W&CP volume 43

offers methods that solve the LfD problem. Most IL methods are classification-based and do no take into account the underlying dynamics of the environment (the human-human interaction in our case), which can be problematic. Indeed, choosing when and how to laugh may depend on previous states of the interaction and thus the dynamics of the interaction may be an important aspect to take into consideration. However, a recent algorithm called Regularized Classification for Apprenticeship Learning (RCAL) (Piot et al., 2014a) uses the underlying information of dynamics existing in the data. Thus, pure and regularized classification (Taskar et al., 2005; Piot et al., 2014a) methods are compared and regularized classification is shown to efficiently learn a behavior on data sets of real laughs in a natural interaction context in Sec. 4

The remainder of the paper is organized as follows. First, we present the LfD paradigm and a large-margin classification method in Sec. 2. Then, we use the large margin method to derive the RCAL algorithm (Piot et al., 2014a) in Sec. 3. Finally, we compare several IL methods and RCAL on some real human-human interactions data.

2 Background

Before introducing a regularized classification algorithm for IL (RCAL), it is necessary to recall some notions and definitions relatives to the IL paradigm. As IL is an implementation of the general LfD problem, which can be formalized properly thanks to the concept of Markov Decision Process (MDP), the remaining of the section is organized as follows. First, we provide some definitions relative to the concept of MDP, then we present the general problem of LfD and finally we show how a large-margin Multi-Class Classification (MCC) which is an IL method can solve the LfD problem.

2.1 Markov Decision Process

A finite MDP (Puterman, 1994) models the interactions of an agent evolving in a dynamic environment and is represented by a tuple $M = \{S, A, R, P, \gamma\}$ where $S = \{s_i\}_{1 \leq i \leq N_S}$ is the state space, $A = \{a_i\}_{1 \leq i \leq N_A}$ is the action space, $R \in \mathbb{R}^{S \times A}$ is the reward function (the local representation of the benefit of doing action a in state s), $\gamma \in]0, 1[$ is a discount factor and $P \in \Delta_S^{S \times A}$ is the Markovian dynamics represented as a function from $S \times A$ to Δ_S where Δ_S is the set of probability distributions over S. The Markovian dynamics P gives the probability, $P(s'|s, a)$, to reach s' by choosing the action a in the state s. A Markovian stationary and deterministic policy π is an element of A^S and defines the behavior of an agent. In order to quantify the quality of a policy π relatively to the reward R, we define the value function. For a given MDP $M = \{S, A, R, P, \gamma\}$ and a given policy $\pi \in A^S$, the value function $V_R^\pi \in \mathbb{R}^S$ is defined as $V_R^\pi(s) = \mathbb{E}_s^\pi[\sum_{t=0}^{+\infty} \gamma^t R(s_t, a_t)]$, where $\mathbb{E}_s^\pi$ is the expectation over the distribution of the admissible trajectories $(s_0, a_0, s_1, \dots)$ obtained by executing the policy π starting from $s_0 = s$. Moreover, the function $V_R^* \in \mathbb{R}^S$, defined as $V_R^* = \sup_{\pi \in A^S} V_R^\pi$, is called the optimal value function. A useful tool is, for a given $\pi \in A^S$, the action-value function $Q_R^\pi \in \mathbb{R}^{S \times A}$:

$$Q_R^\pi(s, a) = R(s, a) + \gamma \mathbb{E}_{P(.|s,a)}[V_R^\pi].$$

It represents the quality of the agent's behavior if it chooses the action a in the state s and then follows the policy π. Moreover, the function $Q_R^* \in \mathbb{R}^{S \times A}$ defined as: $Q_R^* = \sup_{\pi \in A^S} Q_R^\pi$ is called the optimal action-value function. In addition, we have that $Q_R^\pi(s, \pi(s)) = V_R^\pi(s)$ and that $\max_{a \in A} Q_R^*(s, a) = V_R^*(s)$ (Puterman, 1994). Thus, we have $\forall s \in S, \forall a \in A$:

$$R(s, a) = Q_R^*(s, a) - \gamma \sum_{s' \in S} P(s'|s, a) \max_{a \in A} Q_R^*(s', a). \quad (1)$$

Eq. (1) links the reward R to the optimal action-value function Q_R^*. This equation will be useful in the sequel, as one could obtain R from Q_R^* by a simple calculus.

2.2 Learning from Demonstrations

LfD is a paradigm in which an agent (called the apprentice) learns to behave in a dynamical environment from demonstrations of another agent (named the expert). To address this problem, we place ourselves in the MDP framework which is used to describe dynamical systems as a set of states, actions and transitions. In this framework, the learnt behavior takes the form of a policy. More precisely, using MDPs, solving the LfD problem consists in finding the policy of the expert agent in states unvisited by the expert.

The expert agent is supposed to act optimally (with respect to the unknown reward function) and the apprentice can only observe the expert policy π_E via sampled transitions of π_E. Moreover, we suppose that the apprentice has some information about the dynamics which he could have collected by previous interactions. The aim of the apprentice is of course to find a policy π_A which is as good as the expert policy with respect to the unknown reward.

More precisely, we suppose that we have a fixed data-set of expert sampled transitions $D_E = (s_i, \pi_E(s_i), s'_i)_{\{1 \leq i \leq N_E\}}$ where $s_i \sim \nu_E \in \Delta_S$ and $s'_i \sim P(.|s_i, \pi_E(s_i))$. In addition, we suppose that the apprentice has some information about the dynamics via a fixed data-set of sampled transitions $D_P = (s_j, a_j, s'_j)_{\{1 \leq j \leq N_P\}}$ where $s_j \sim \nu_P \in \Delta_S$ and $s'_j \sim P(.|s_j, a_j)$. We have $D_E \subset D_P$ and no particular assumptions are made considering the choice of the action a_j or the distributions ν_E and ν_P which can be considered unknown. Those requirements (used for example by Klein et al. (2012); Boularias et al. (2011)) are not strong and can be fulfilled by real-life applications.

One can argue that having a data-set of sampled transitions D_P is a strong assumption. However, the presented algorithms can be run with $D_P = D_E$ which is the case shown in the experiments (see Sec. 4).

2.3 The large margin approach for Classification

To tackle the problem of LfD, it is possible to reduce it to an MCC problem (Pomerleau, 1989; Ratliff et al., 2007; Ross and Bagnell, 2010; Syed and Schapire, 2010). The goal of MCC is, given a training set $D = (x_i \in X, y_i \in Y)_{\{1\leq i\leq N\}}$ where X is a compact set of inputs and Y a finite set of labels, to find a decision rule $g \in Y^X$ that generalizes the relation between inputs and labels. Ratliff et al. (2007) use a large margin approach which is a score-based MCC where the decision rule $g \in Y^X$ is obtained via a score function $q \in \mathbb{R}^{X\times Y}$ such that $\forall x \in X, g(x) \in \text{argmax}_{y\in Y} q(x, y)$. The large margin approach consists, given the training set D, in solving the following optimization problem:

$$q^* = \underset{q\in\mathbb{R}^{X\times Y}}{\text{argmin}}\, J(q), \tag{2}$$

$$J(q) = \frac{1}{N}\sum_{i=1}^{N} \max_{y\in Y}\{q(x_i, y) + l(x_i, y_i, y)\} - q(x_i, y_i),$$

where $l \in \mathbb{R}_+^{X\times Y\times Y}$ is called the margin function. If this function 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 (Ratliff et al., 2007). Instead of requiring only that the example label is scored higher than all other labels, one requires it to be better than each label y by an amount given by the margin function. Another way to improve generalization is to restrain the search of q^* to an hypothesis space $\mathfrak{H} \subset \mathbb{R}^{X\times Y}$. However, it introduces a bias.

Applying the large margin approach to the LfD problem is straightforward. From the set of expert trajectories D_E, we extract the set of expert state-action couples $\tilde{D}_E = (s_i, \pi_E(s_i))_{\{1\leq i\leq N_E\in\mathbb{N}^*\}}$ and we try to solve:

$$q^* = \underset{q\in\mathfrak{H}\subset\mathbb{R}^{S\times A}}{\text{argmin}}\, J(q) \tag{3}$$

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

The policy outputted by this algorithm would be $\pi_A(s) \in \text{argmax}_{a\in A}\, \hat{q}(s, a)$ where $\hat{q}$ is the output of the minimization. The advantages of this method are its simplicity and the possibility to use a boosting technique (Ratliff et al., 2007) to solve the optimization problem given by Eq. (3). However, this is a pure classification technique which does not take into account the dynamics information contained in the sets D_E and D_P. In the following section, we present an algorithm, RCAL (Piot et al., 2014a), based on the large margin approach which uses the dynamics information by adding an original regularization term to $J(q)$.

3 Regularized Classification

In this section, RCAL , a non-parametric AL algorithm using the information contains in the dynamics, is introduced.

First, it is important to remark that the large-margin classification problem described by Eq. (2) tries to find a function minimizing an empirical criterion obtained from sparse data in a given set of functions called the hypothesis space. This is, in general, an ill-posed problem (infinite number of solutions when the hypothesis space is rich enough) and a way to solve it is the regularization theory of Tikhonov and Arsenin (1979), which adds a regularization term that can be interpreted as a constraint on the hypothesis space. Evgeniou et al. (2000) show how the work of Vapnik (1998) set the foundations for a general theory which justifies regularization in order to learn from sparse data. Indeed, the basic idea of Vapnik's theory is that the search for the best function must be constrained to a *small* (in terms of complexity) hypothesis space. If the hypothesis space is too *large*, a function that exactly fits the data could be found but with a poor generalization capability (this phenomenon is known as over-fitting). Evgeniou et al. (2000) show that the choice of the regularization parameter λ corresponds to the choice of an hypothesis space: if λ is *small* the hypothesis space is large and vice-versa. In the large margin framework, a general and natural way to introduce regularization is to add to $J(q)$ a regularization term $\lambda W(q)$ and to consider the following optimization problem:

$$q^* = \underset{q\in\mathfrak{H}\subset\mathbb{R}^{S\times A}}{\text{argmin}}\, J_W(q) = \underset{q\in\mathbb{R}^{S\times A}}{\text{argmin}}\, \left(J(q) + \lambda W(q)\right).$$

where $\lambda \in \mathbb{R}_+^*$ and W is a continuous function from $\mathbb{R}^{S\times A}$ to $\mathbb{R}_+$.

In order to introduce the dynamics information contained in the data-set D_P, Piot et al. (2014a) assume that the unknown reward function for which the expert is optimal is sparse. This assumption helps to choose an appropriate regularization term which constrains the hypothesis space, hence reduces the variance of the method. To do so, Piot et al. (2014a) remark that a good score function must verify:

$$\forall s \in S, \pi_E(s) \in \underset{a\in A}{\text{argmax}}\, q(s, a),$$

which means that there exists a reward function $R_q \in \mathbb{R}^{S\times A}$ for which π_E is optimal and such that $q(s, a) = Q_R^*(s, a)$ (see Sec. 2.1). This reward is given via the inverse Bellman equation (Eq. (1)) $\forall s \in S, \forall a \in A$:

$$R_q(s, a) = q(s, a) - \gamma \sum_{s'\in S} P(s'|s, a) \max_{a\in A} q(s', a).$$

As the reward R_q is assumed to be sparse, a natural choice for $W(q)$ is $\|R_q\|_{1,\nu_P}$ where $\nu_P \in \Delta_{S\times A}$ is the distribution from where the data are generated. However, as $P(.|s, a)$ is unknown, it is not possible to compute $R_q(s, a)$. Thus, Piot et al. (2014a) rather consider, for each

transition in D_P, the unbiased estimate of $R_q(s_j, a_j)$ noted $\hat{R}_q(j)$ which is can be obtained from data:

$$\hat{R}_q(j) = q(s_j, a_j) - \gamma \max_{a \in A} q(s'_j, a).$$

Therefore, in order to introduce the dynamics information contained in D_P, the choice of the regularization term W is:

$$W(q) = \frac{1}{N_P} \sum_{j=1}^{N_P} |\hat{R}_q(j)| = \frac{1}{N_P} \sum_{j=1}^{N_P} |q(s_j, a_j) - \gamma \max_{a \in A} q(s'_j, a)|,$$

Even if $W(q)$ is not an unbiased estimate of $\|R_q\|_{\nu_P,1}$, it is shown by Piot et al. (2014a) that the constraints imposed by the regularization term $W(q)$ are even stronger that the ones imposed by $\|R_q\|_{\nu_P,1}$.

Thus, the algorithm RCAL consists in solving the following optimization problem:

$$q^* = \underset{q \in \mathfrak{H} \subset \mathbb{R}^{S \times A}}{\operatorname{argmin}} \; J_W(q)$$

$$J_W(q) = J(q) + \frac{\lambda}{N_P} \sum_{j=1}^{N_P} |\hat{R}_q(j)|.$$

Then, the policy outputted by RCAL is $\pi_A(s) \in \operatorname{argmax}_{a \in A} \hat{q}(s, a)$ where $\hat{q}$ is the output of the minimization of J_W. In order to minimize the criterion $J_W(q)$, a boosting technique, initialized by a minimizer of J_q for instance, is used (see the work of Piot et al. (2014a) for more details) in our experiments. Such a technique is interesting as it avoids to choose features which is often problem dependent. However, there is no guarantee of convergence.

4 An ECA deciding when and how to laugh

In order to build a laugh-aware agent, we searched for an interaction scenario which is realistic and simple enough to be set up. Thus, we opted for the one used by Niewiadomski et al. (2013) that implies telepresence. This scenario involves two subjects watching a funny stimulus. These two persons do not share the same physical space: they watch the same content simultaneously on two separate displays. However, they can see each other reactions in a small window placed on the top of the displayed content. This scenario corresponds to very common situations in real life when someone wants to share interesting content over the web. Such interactions are recorded thanks to cameras and microphones in order to build an ECA. More precisely, features such as laugh intensity, speech probability, smile probability extracted by analysis components are computed each 400 ms from the first person's recordings and form the state s (more details are provided by Niewiadomski et al. (2013)). In addition, from the recordings of the second person (playing the role of the expert) is extracted the type of laugh which corresponds to the action to be imitated by the ECA and form the expert action $\pi_E(s)$. They are 4 types of laugh (thus 4 actions): strong laugh, normal laugh, quiet laugh and silence. Our data base named D_E is composed of several expert trajectories with a total of 2378 state-action couples.

Thus, we have a data-set D_E on which we can apply any IL method and more particularly RCAL (with $D_E = D_P$ and $\lambda = 0.1$). Niewiadomski et al. (2013) use a k-Nearest Neighbors (k-NN with $k = 1$) algorithm (Cover and Hart, 1967). Here, we compare the results obtained via an averaged K-fold cross validation ($K = 5$) between RCAL, Classif (RCAL with $\lambda = 0$, that is the method of Ratliff et al. (2006) presented in Sec. 2.3), k-NN and a classification tree (Breiman et al., 1984) on the data set D_E.

Algorithms	Global perfomance	Good laugh chosen	Performance on Silence
RCAL	**0.7722**	**0.3147**	0.8872
Classif	0.7671	0.2316	**0.8934**
k-NN	0.7381	0.2681	0.8521
Tree	0.7533	0.2914	0.8683

In our data-set, the laughs and silences are unbalanced (approximately 80% of silence for 20% of laughs). That is why, it is a difficult task to classify correctly the laughs. We observe that the introduction of the regularization improves drastically the performance between RCAL with $\lambda = 0.1$ and the Classif algorithm (RCAL with $\lambda = 0$). Thus, taking into account the underlying dynamics helps to improve the performance. RCAL has also a better performance compared to the k-NN algorithm and the classification tree.

5 Conclusion and Perspectives

In this paper, a method for learning when and how an avatar (ECA) should laugh during an interaction with humans was presented. It is based on data-driven IL algorithms and especially on structured and regularized classification methods. It is shown, in a telepresence scenario, that RCAL outperformed other classification methods. Compared to previous experimentations (Niewiadomski et al., 2013; Piot et al., 2014b), this method provides better results as it takes into account the underlying dynamics of the interaction.

Here, LfD is reduced to an MCC problem. Yet, LfD can also be solved by other methods such as Inverse Reinforcement Learning (IRL) (Russell, 1998; Klein et al., 2012). Actually, IRL has been shown to work better for some types of problems (Piot et al., 2013) and has already been used to imitate human users in the case of spoken dialogue systems (Chandramohan et al., 2011). Therefore, it seems natural to extend this work to IRL 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 (Pietquin and Dutoit, 2006; Pietquin and Hastie, 2013).

References

A. Boularias, J. Kober, and J. Peters. Relative entropy inverse reinforcement learning. In *Proc. of AISTATS*, 2011.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. 1984.

S. Chandramohan, M. Geist, F. Lefèvre, and O. 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)*, pages 1025–1028, Florence, Italy, August 2011.

T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13 (1):21–27, 1967. ISSN 0018-9448. doi: 10.1109/TIT.1967.1053964.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

E. Klein, M. Geist, B. Piot, and O. Pietquin. Inverse reinforcement learning through structured classification. In *Proc. of NIPS*, 2012.

R. Niewiadomski, S. Pammi, A. Sharma, J. Hofmann, Tracey, R. T. Cruz, and B. Qu. Visual laughter synthesis: Initial approaches. In *Proceedings of the Interdisciplinary Workshop on Laughter and other Non-Verbal Vocalisations*, pages 10–11, Dublin, Ireland, October 2012.

R. Niewiadomski, J. Hofmann, T. Urbain, T. Platt, J. Wagner, J. Piot, H. Cakmak, S. Pammi, T. Baur, S. Dupont, M. Geist, F. Lingenfelser, G. McKeown, O. Pietquin, and W. Ruch. Laugh-aware virtual agent and its impact on user amusement. In *Proc. of AAMAS*, 2013.

O. Pietquin and T. Dutoit. A Probabilistic Framework for Dialog Simulation and Optimal Strategy Learning. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):589–599, March 2006. doi: 10.1109/TSA.2005.855836.

O. Pietquin and H. Hastie. A survey on metrics for the evaluation of user simulations. *Knowledge Engineering Review*, 28(01):59–73, February 2013. doi: 10.1017/S0269888912000343.

B. Piot, M. Geist, and O. Pietquin. Learning from demonstrations: Is it worth estimating a reward function? In *Proc. of ECML*, 2013.

B. Piot, M. Geist, and O. Pietquin. Boosted and reward-regularized classification for apprenticeship learning. In *Proc. of AAMAS*, 2014a.

B. Piot, O. Pietquin, and M. Geist. Predicting when to laugh with structured classification. In *Proc. of Interspeech*, 2014b.

D. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. Technical report, DTIC Document, 1989.

M. Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 1994. ISBN 0471619779.

N. Ratliff, J. Bagnell, and M. Zinkevich. Maximum margin planning. In *Proc. of ICML*, 2006.

N. Ratliff, J. Bagnell, and S. Srinivasa. Imitation learning for locomotion and manipulation. In *Proc. of IEEE-RAS International Conference on Humanoid Robots*, 2007.

S. Ross and J. Bagnell. Efficient reductions for imitation learning. In *Proc. of AISTATS*, 2010.

S. Russell. Learning agents for uncertain environments. In *Proc. of COLT*, 1998.

M. Schröder, E. Bevacqua, R. Cowie, F. Eyben, H. Gunes, D. Heylen, M. ter Maat, G. McKeown, S. Pammi, M. Pantic, C. Pelachaud, B. Schuller, E. de Sevin, M. Valstar, and M. Wöllmer. Building autonomous sensitive artificial listeners. *IEEE Transactions on Affective Computing*, 3(2):165–183, 2012. ISSN 1949-3045.

U. Syed and R. Schapire. A reduction from apprenticeship learning to classification. In *Proc. of NIPS*, 2010.

B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *Proc. of ICML*, 2005.

A. Tikhonov and V. Arsenin. *Methods for solving ill-posed problems*, volume 15. Nauka, Moscow, 1979.

J. Urbain, H. Cakmak, and T. Dutoit. Evaluation of HMM-based laughter synthesis. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, pages 7835 – 7839, Vancouver, Canada, May 2013.

V. Vapnik. *Statistical learning theory*. Wiley, 1998.