Advancing Affect Modeling via Preference Learning and Unsupervised Feature Extraction

Héctor P. Martínez
Center for Computer Games Research
IT University of Copenhagen

A thesis submitted for the degree of
Doctor of Philosophy
February 2013

Abstract

Recognizing and reacting to emotions are fundamental elements in communication among humans. Transferring these skills to computers is an exceptionally complex task, in part, due to the subjective nature of emotions and the subtle, context-dependent and disperse properties of their manifestations. This thesis investigates methods to uncover the mapping between emotions and their manifestations based on observations of humans experiencing specific affective states.

The first challenge is to annotate and, in turn, recognize the affective states experienced. While posing interesting computational difficulties, ordinal reports such as rankings and ratings can yield more reliable affect annotations than alternative tools. This thesis explores preference learning methods to automatically learn computational models from ordinal annotations of affect. In particular, an extensive collection of training strategies (error functions and training algorithms) for artificial neural networks are examined across synthetic and psycho-physiological datasets, and compared against support vector machines and Cohen's method. Results reveal the best training strategies for neural networks and suggest their superiority over the other examined methods.

The second challenge addressed in this thesis refers to the extraction of relevant information from physiological modalities. Deep learning is proposed as an automatic approach to extract input features for models of affect from physiological signals. Experiments on psycho-physiological datasets show that these methods, in combination with automatic feature selection, can reveal information that yields more accurate predictors of affect than typical hand-crafted feature extractors examined in Affective Computing research.

The third challenge arises from the complexity of hand-crafting feature extractors that combine information across dissimilar modalities of input. Frequent sequence mining is presented as a method to learn feature extractors that fuse physiological and contextual information. This method is evaluated in a game-based dataset and compared against ad-hoc extracted features. The evaluation reveals that this unsupervised method, combined with appropriate feature selection algorithms, yields more accurate predictors of affective player experiences than hand-crafted single-modality features.

In summary, this thesis proposes and validates a complete methodology for building models of affect from ordinal annotations with minimal expert-knowledge, advancing affect modeling towards an automated data-driven process. The generality of the thesis' key findings presented along with the limitations and the extensibility of the proposed components are discussed.

Acknowledgements

A long list of people have assisted me in the completion of this dissertation. I would not be able to sort the list precisely by importance as all the help was invaluable, but I can clearly place my advisor, Georgios N. Yannakakis, at the top. I would like to express my most sincere gratitude to him for his guidance over the past years, but most of all, for introducing me to the pleasures of research.

I am also grateful to Yoshua Bengio and everyone else at LISA lab for hosting me there and letting me participate of their fascinating research.

I would like to extend special thanks to Tobias Malhmann and Alessio Magro for putting at my disposal outstanding hardware resources; without their help, I could not have tested my ideas.

Thanks also to all my colleagues at the Center for Computer Games Research for all the stimulating discussions, but also for the fun times. Writing this dissertation would have not been possible outside that exciting working environment.

Loads of thanks to Patricia Trincado for being there for me all these years, but also for her help during the final weeks of writing, I would not have completed this thesis without her.

Last, but not least, I would like to thank my parents and my brother, nothing of all this would be possible without their love and support.

To Pati, Víctor, Carmen and Foro

Contents

List of Figures

List of Tables

List of Abbreviations

AC	affective computing
AE	auto-encoder
ANN	artificial neural network
BP	backpropagation
BV	blood volume
BVP	blood volume pulse
CAE	convolutional auto-encoder
CI	computational intelligence
CM	Cohen's model
CNN	convolutional neural network
DEAP	database for emotion analysis using physiological signals
DL	deep learning
DM	data mining
FS	automatic feature selection
GA	genetic algortihm
GFS	genetic feature selection
GSP	generalized sequence patterns
HR	heart rate
HRV	heart rate variability
MB	Maze-ball
ML	machine learning
MLP	multi-layer perceptron
NE	neuroevolution
OR	object ranking
PCA	principal component analysis
PL	preference learning
SC	skin conductance
SFS	sequential feature selection
SLP	single-layer perceptron
SM	frequent sequence mining
SVM	support vector machine

Chapter 1

Introduction

Emotions are fundamental elements of human intelligence linked to attention, perception and behavior (Goleman, 2006; Scherer, 2003; Frijda, 1986; Lopes et al., 2005). The voluntary expressions and unconscious manifestations of emotions give away clues on our state of mind. Predicting emotion from such manifestations and being able to act on that knowledge, is considered an essential skill for basic social interactions and a fundamental ability for eminent managers, lecturers and showmen. Nowadays we work with, learn from and get entertained by computers that ignore our emotions. Models that would enable machines to detect and respond to them, would augment our interaction with technology. In particular, machines that could detect and react to our stress, confusion or enjoyment levels would be able to improve our productivity at work, facilitate learning or maximize entertainment.

Unfortunately, more than 15 years after the early studies in *Affective Computing* (AC), (Picard, 1995) the problem of detecting and modeling emotions, or more generally *affect*, in the context of human-computer interaction (HCI) remains complex and largely unexplored. Research in the affect modeling field is focused, primarily, on the study and use of artificial intelligence (AI) techniques for the construction of computational models of affect. The key challenges one faces when attempting to create these models are inherent in the vague definitions and fuzzy boundaries of affect (Calvo and D'Mello, 2010), and in the modeling methodology followed. In this context, open research questions are still present in all key components of the modeling process. These include, first, the appropriateness of the modeling tool employed to map emotion manifestations and responses to annotated affective states; second, the processing of signals that express those manifestations (i.e. model input); and third, the way affect annotation (i.e. model output) is handled. This thesis touches upon all three key components (i.e. input, model, output) introducing new tools for affect modeling.

1.1 Motivation and Challenges

Visible signs of emotion often accompany our experiences with technology, for example a smile when a phone makes a good recommendation, a yawn when an online course presents the next exercise and a yell of anger when a game pulls off a seemingly impossible defeat. Loads of precious, accurate and automatic feedback falls on deaf ears while thrown at the machine. Models of affect would endow computers with the ability of assessing the subjective experience of users from these visible signs and additional non-visible bodily changes (e.g. changes in heart rate). This new communication channel between humans and technol-

ogy could be applied to enhance current testing methodologies: estimations of the affective responses of users to a new device or application would provide more interesting information than regular HCI performance metrics (e.g. selection speed (Natapov et al., 2009)) and bypass the disruption and tediousness of completing questionnaires. More advanced applications arise from the integration of these models as part of adaptation mechanisms that modify the content or behavior of the system based on the estimated affective state of the user, namely an *affecttive loop* (Höök, 2009; Fairclough, 2009).

Computer games in particular provide the ideal arena for the study of this loop since they can elicit a plethora of affective states (Perron, 2005) and their dynamic nature opens up a large number of possibilities for personalized and adaptive content (Gilleade et al., 2005; Hudlicka, 2009; Yannakakis and Togelius, 2011; Yannakakis and Paiva, 2013). However, a common critique for adaptation in games is that most players do not want to feel patronized by a game that lowers the difficulty according to the player's performance. Models of affect provide alternative heuristics (predictors of affect), such as boredom or engagement, that games can rely upon for adaptation. For each playing session, an affect-adaptive game could generate content tailored to the current player's mood and adapt different elements along the session to maximize enjoyment and minimize long periods of boredom. Additionally, a game aimed at eliciting emotions such as fear or even frustration, could use models of emotion to evaluate player's responses to different game elements and find the most effective stimuli for each player.

Moving from entertainment technologies to educational, affective adaptation holds the potential of boosting the efficacy of virtual tutors (McQuiggan et al., 2007). As emotions are strongly linked to attention, motivation and learning, a tutor able to minimize boredom and frustration and react to confusion would help students to learn faster than another tutor that only relies on performance measures. A final and highly relevant application for affect modeling is the implementation of tools for assisting people with difficulties to express or recognize emotions (Kaliouby et al., 2006). Models that rely on unconscious manifestations of emotion such as physiological signals, can help parents and educators of children living with such disabilities. Examples of these technologies are flourishing, and while a great promise has been shown via the exploration of basic mental-body mappings of affect — such as heightened skin conductance and arousal (Fletcher et al., 2010) or smiles and enjoyment (McDuff et al., 2012) — general and reliable models for more complex affective states are not yet available.

One of the main obstacles in the attempt to model affect is the difficulty of collecting reliable representative data to create the model. This difficulty is caused for a number of factors determined by the fundamental characteristics of affect: first, affective states are subjective and depend on a large number of factors, which makes impossible to devise an experimental protocol that accurately elicits, across a group of different individuals, a specific affective state; hence the need for annotation. Second, affective states are presented with different intensities and the boundaries among different states are blurry; thus the reliability and accuracy of annotated affect is compromized. Third, affective states are not instantaneous and last a variable amount of time; so modeling methods are required to process sequential information. Fourth, the manifestations of affect are complex and not completely understood; hence any designer-driven decision in the modeling process could hinder important effects. Fifth, the manifestations of affect can be ambiguous as there is no one-to-one mapping between the bodily and mental states of affect; thus, several modalities of inputs need to be considered simultaneously.

This thesis introduces a generic methodology towards more accurate and reliable models

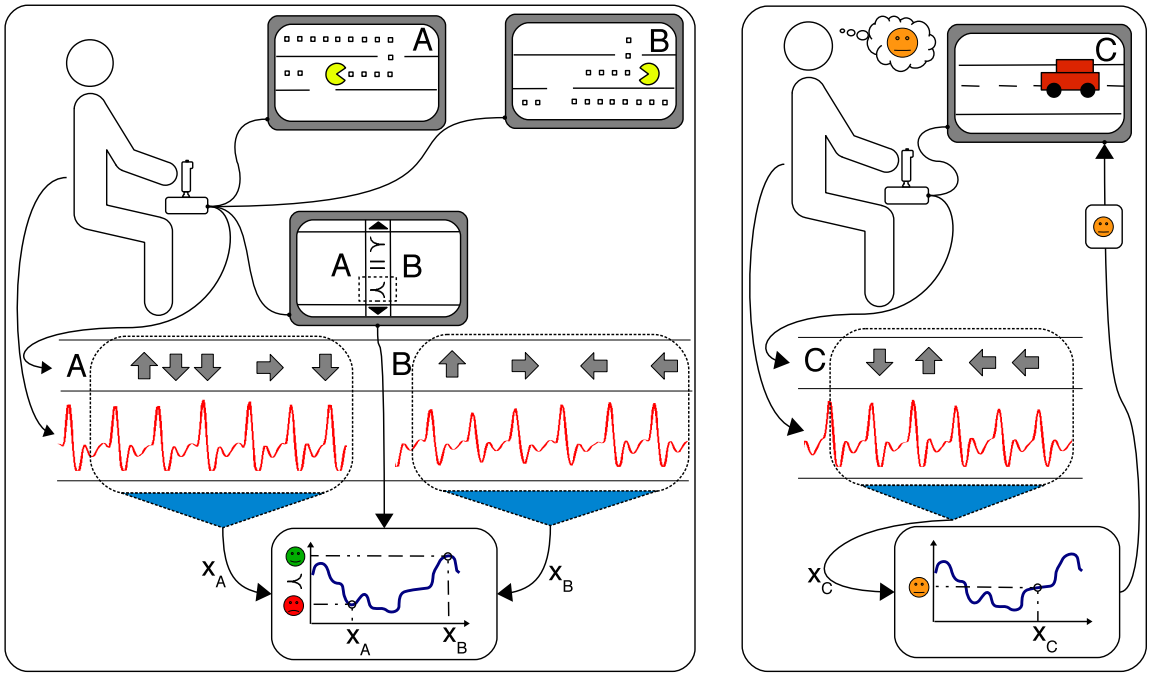
of affect by proposing and evaluating methods that overcome the above-mentioned difficulties. In particular, first, several *preference learning* methods (PL) are proposed for constructing models of affect from ordinal reports. These reports create a more challenging and less investigated machine learning problem but they facilitate more reliable annotations of affect intensity, as they only require comparisons among affective experiences instead of absolute numerical estimations of the intensity of each experience. Hence the proposed methods aim at creating more accurate models from more reliable data. Second, a deep learning method is introduced for extracting novel complex physiological features that can feed the input of the models of affect which are built through preference learning. This method processes physiological signals automatically introducing minimal human-biases, thus potentially revealing sequential patterns not represented by ad-hoc designed metrics. Finally, a sequence mining method is presented for synthesizing the temporal relation among input modalities; this method conforms an automatic procedure for fusing physiological and contextual information facilitating multimodal models of affect. The introduced methodology is tested on a number of synthetic and user survey datasets that contain affect annotations. In the next two sections, the problem of affect modeling is formulated more concretely and the contributions of this thesis are specified in more detail.

1.2 Problem Formulation

Despite the extensive amount of research in affective phenomena, researchers have not yet reached consensus on a precise definition of affect. Affect is generally used as a term to encompass dissimilar mental states related to emotion (affective constructs) such as personality traits, preferences, attitudes and moods (Davidson, 1994; Davidson et al., 1994). Compared to the other constructs, emotions are typically depicted as short episodes that span for seconds or minutes (Scherer, 2000; Frijda, 1993). Modeling affect in this thesis is restricted to modeling mental states elicited by brief experiences including not only *traditional* emotions (e.g. frustration and anxiety) but also other constructs related to affect and cognition, and fundamental to entertainment (e.g. "fun" and perceived challenge). For the sake of readability, all these mental constructs are referred to as *affective states* throughout this dissertation.

The approach to affect modeling investigated here is based on two points in which most theorists agree upon (Picard, 1997; Scherer, 2000). First, it is widely accepted that emotions have a mental and a bodily component. The former refers to the subjective feeling and the cognitive processes that occur during an emotional episode while the latter refers to motor expressions (e.g. facial expressions) and physiological changes (e.g. increase of heart rate). The second point of consensus among the researchers in the field is that emotions normally occur as a response to internal or external stimuli or events that are significant to the organism.

Based on these two points of agreement, a model of affect can be defined as a predictor of the mental component of emotion that relies on the (measurable) bodily component and the external stimuli that triggers the emotional response. In computational terms, this model can be further defined as a function that processes several parallel streams of multimodal data (e.g. video feed of the face, heart rate signal and events occurring in the computer interface) and predicts a quantifiable representation of the affective state (e.g. 0.9 probability of the user being frustrated). This idea is exemplified in Figure 1.1b through a model that estimates the affective state of a video-game player relying on certain



(a) Training from pairwise self-report

(b) Evaluation

Figure 1.1: Problem formulation: a model of affect estimates the affective state of a user based on information about her physiological state and her interaction with the system. A *feature extraction* phase is required to reduce the high-dimensionality of the input signals and provide suitable inputs to the model. Training data is collected from users interacting with the system and reporting the target affective states by sorting several sessions (e.g. A and B) by the intensity of the state felt (e.g. B was more exciting than A, $A \prec B$). The model is trained by adjusting its parameters to make predictions that satisfy the training reports (e.g. a higher prediction of excitement for B with respect to A). The trained model can then be applied to estimate in real-time the affective state of new users.

metric ($\mathbf{x}$) that combines physiological signals and the sequence of events occurring in the game. The central problem addressed in this dissertation is the application of artificial and computational intelligence methods to automatically infer that model and that metric from a set of observations of the target affective state.

Collecting reliable observations of the affective state is key to creating a valid and accurate model. As there are no available systematic approaches for objectively measuring affective states it, inevitably, requires humans to make the annotations through self-reports, reports as an expert observer or through the design of the experimental protocol (e.g. custom-design a frustrating experience). For affect modeling, the annotations are usually reduced to binary (the affective state is felt or not) or numerical values (intensity of the affective state) matching well the most popular AI tasks, namely *classification* and *regression*. Algorithms designed for these tasks attempt to find the one-to-one mapping between each experience (data sample) and its corresponding target output (affect annotation). While annotating binary labels oversimplifies the task by neglecting affect intensities, annotating exact numerical intensities of affective states is a rather challenging (and to a degree, naive) task. Note that user ratings follow unknown subjective scales that vary across users and

along time (Viswanathan, 1993; Costner, 1969), and therefore they should not be treated as real-values. A compromise is found by treating ratings as ordinal labels or by using ordinal annotations such as rankings (e.g. the affective state was felt stronger during the first than the second experience) that represent different relative levels of intensity without introducing unreliable numerical values.

As illustrated in Figure 1.1a, the process for creating models of affect in this dissertation is rendered as learning the mapping between objective measurable cues and ordinal annotations of different emotional experiences. Three fundamental challenges can be distinguished in this process:

- The first key challenge emerges from the switch from nominal and real-valued annotations to ordinal annotations (rankings, pairwise preferences or ratings as ordinal values): the computational learning task moves from one of classification or regression to *object ranking* (OR) in which each data sample no longer maps to a given label but to a position within an ordering of the data samples. This problem has been studied within the machine learning sub-field of preference learning (Fürnkranz and Hüllermeier, 2010a); however, **affect datasets feature a number of particular characteristics that create challenges for standard PL methods**. These include small sample size, noisy inputs, significant differences across groups of data samples and noisy outputs. These characteristics arise from the data collection experiments using specialized sensing hardware, which leads to time consuming sessions, experimental biases and data artifacts, and participant-specific baseline recordings; in addition of course to the challenging task of annotation affective states.

  This challenge motivates the first objective of this thesis which consists of identifying the suitability of dissimilar PL techniques for affect modeling and beyond. In particular, we explore a series of *artificial neural network* (ANN) variants, *support vector machines* (SVM) and *Cohen's method* (CM).

- A second challenge faced when learning models of affect is dealing with the **large dimensionality of the streams of input data and the multiple entangled factors of variation independent of affect**. Consider, for instance, that a monitored episode of frustration lasts for 60 seconds, a physiological sensor is recording a signal at 32 Hz and one rating for the whole experience is reported; in this example one single annotation of frustration is related to $60 \times 32 = 1920$ physiological recordings. Additionally, if an internal body change related to e.g. digestion alters the physiological signal, the effect would be entangled with the manifestations of frustration. Finding a function that maps that amount of data directly to an estimation of the affective state is practically impossible, especially considering that affect datasets generally contain a small number of data samples. Traditionally, the dimensionality of the inputs is reduced by extracting several characteristic attributes of the input streams (*feature extraction*) and feed them to the models. While this phase is beneficial for affect modeling, it limits the creativity of feature design to the expert (i.e. the AC researcher) resulting in potentially inappropriate affect detectors that might not be able to capture the manifestations of affect embedded in the raw input signals. To overcome the limitations of human-designed features (ad-hoc features), this thesis introduces a method based on deep architectures for automatic discovery of features that can yield more accurate models of affect.

  This method renders the second objective of this thesis consisting of evaluating the

efficacy of deep learning methods for reducing the dimensionality of input signals. In particular, we focus on the reduction of key physiological signals, namely skin conductance (SC), heart rate (HR), blood volume (BV) and blood volume pulse (BVP). In addition, we test whether more accurate models of affect can be created by replacing manual ad-hoc features with deep learned features. A sub-goal of this objective is to test the efficacy of automatic feature selection (FS) for finding the deep learned features that are relevant for affect, hence creating more accurate and expressive models of affect.

- A third challenge arises from the **complexity of effectively hand-crafting multimodal features**. While there exist well-stablished ad-hoc features for individual physiological signals (e.g. average heart inter-beat amplitude), features that combine information across modalities are not common. This usually results in models of affect that combine the different input modalities using unimodal features (i.e. feature level fusion (Pantic and Rothkrantz, 2003)), potentially missing relevant interactions among modalities. In addition, when context information is considered, complex ad-hoc features can rarely be reused due to large differences on the definition of actions and events across dissimilar applications or systems. This thesis introduces the application of *frequent sequence mining* (SM) algorithms for automatic extraction of features from multiple modalities including context.

  The corresponding objective consists of evaluating the efficacy of sequence mining methods for automatically fusing multiple input modalities. In particular, we investigate the fusion of physiological and contextual information, and test whether more accurate models of affect can be created by replacing manual ad-hoc feature extraction with sequence mining. Similarly to the second objective, a sub-goal of this third objective consists of testing the efficacy of automatic feature selection mechanisms for boosting the accuracy of models of affect based on automatically extracted multimodal features.

In summary, the main goal of this thesis is to introduce appropriate methods and tools towards more reliable and accurate computational models of affect. The three main research questions that are raised towards achieving that goal and addressed in this thesis are as follows:

- **What are the most adequate computational methods to create models of affect from ordinal annotations?** Experiments in Chapter 5 tackle this question by evaluating which training algorithms and error functions create the most accurate artificial neural network for predicting synthetic functions and affect datasets; in addition, support vector machines and Cohen's method are applied to the same tasks and compared against the ANN models.

- **Can automatic feature extraction reveal relevant components from physiological signals that lead to more accurate models of affect than ad-hoc features?** Chapter 6 addresses this question by analysing features learned from physiological datasets using *convolutional neural networks* (CNNs), and comparing the prediction accuracy of affect models built on these features and on standard ad-hoc features.

- **Can automatic feature extraction capture the interrelationships among input modalities and lead to more accurate models of affect than ad-hoc**

single-modality features? Chapter 7 helps to answer this question by extracting multimodal features from physiology and context signals using the *generalized sequential patterns* (GSP) algorithm; prediction models based on these features are compared against models fed by standard ad-hoc features in order to reveal the relative prediction power of the sequence-mining multimodal features.

The next section outlines the contributions of this thesis which arise by answering the above-mentioned research questions.

1.3 Novelty and Contributions

The main contributions of this thesis are aligned along the process of modeling affect from annotated user data. In particular, this thesis brings methods from the fields of data mining (DM), artificial intelligence and computational intelligence (CI) to address three different challenges within the affect modeling process:

- **Methods for modeling emotion from ordinal annotations**: a set of tools for training models of affect using ordinal data (rankings or ratings). The methods are evaluated on synthetic data representing common patterns and characteristics of affect datasets. In addition, the methods are validated in two real datasets containing self-reports of affect and several input modalities.
- **Methods for automatically extracting physiological features**: we propose an algorithm based on convolutional neural networks and *auto-encoders* (AE) for unsupervised extraction of features of blood volume, blood volume pulse, skin conductance and heart rate. This is the first reported study of such methods on physiological signals. Also, for the first time deep learning is used to create models of affective experiences.
- **Methods for automatically fusing physiological modalities and context**: a frequent sequence mining algorithm is introduced as a method for extracting features from asynchronous and simultaneous signals. In particular, the method is utilized to extract patterns from discrete skin conductance and blood volume pulse events and the sequence of actions arising from the interaction between user and system (keystrokes and actions in a video-game). This technique has not been used before in the context of human-computer interaction or as a feature extraction method for affect modeling.
- **Methods for automatic feature selection**: the efficacy of two automatic feature selection algorithms is validated across a number of ad-hoc and automatically extracted physiological and multimodal feature sets in two affect datasets.

Additionally, this thesis also contributes to the the fields of machine learning and computer games research:

- **Methods for object ranking**: a set of algorithms and tools for training artificial neural networks using ordinal data is compiled combining existing training algorithms and error functions scattered in the machine learning and preference learning literature. New error functions are also introduced to complete the spectrum of possible

approaches to neural network-based object ranking. This is the first thorough empirical study of artificial neural networks across training algorithms and a large set of error functions for a preference learning task. Additionally, support vector machines and Cohen's method are evaluated to validate the efficacy of ANN approaches against dissimilar well-established preference learning algorithms.

- **Methods for player modeling**: collectively, the thesis contributes to the field of computer games (arguably, the richest human computer interaction sub-domain) and player experience as it introduces methodologies for emotion annotation, feature extraction and selection, and affect modeling that yield models of affect of high prediction accuracies.

1.4 Publications

The work conducted in this thesis has resulted in a number of technical peer-reviewed publications in journals and conference proceedings that are listed below:

Journal Publications

- Martínez H.P., Yannakakis, G.N. - *Artificial Neural Networks for Object Ranking: an Empirical Comparison* (under preparation)
- Martínez, H.P., Bengio, Y., Yannakakis, G.N. - *Learning Deep Physiological Models of Affect*, IEEE Computational Intelligence Magazine, 9:1:2033, 2013
- Yannakakis, G. N., Martínez, H. P., Jhala, A. - *Towards Affective Camera Control in Games*, User Modeling and User-Adapted Interaction, 20:313-340, 2010

Conference Proceedings and Workshop Papers (peer-reviewed)

- Martínez, H.P., Yannakakis, G.N. - *Mining Multimodal Sequential Patterns: A Case Study on Affect Detection*, Proceedings of the International Conference on Multimodal Interaction, ICMI, Alicante, November, 2011 (Outstanding student paper award)
- Martínez, H.P., Garbarino, M., Yannakakis, G.N. - *Generic Physiological Features as Predictors of Player Experience*, Proceedings of the international conference on Affective Computing and Intelligent Interaction, ACII, Memphis, October, 2011
- Martínez, H.P., Yannakakis, G.N. - *Analysing the Relevance of Experience Partitions to the Prediction of Players Self-Reports of Affect*, Proceedings of the international conference on Affective Computing and Intelligent Interaction, Emotion in Games workshop, EMOGames, Memphis, October, 2011
- Martínez, H.P., Yannakakis, G.N. - *Genetic Search Feature Selection for Affective Modelling: a Case Study on Reported Preferences*, Proceedings of the international workshop on Affective Interaction in Natural Environments, AFFINE, Turin, October, 2010
- Martínez, H. P., Hullett, K., Yannakakis, G. N. - *Extending Neuro-evolutionary Preference Learning through Player Modelling*, Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG, Copenhagen, August, 2010

- Schwartz, M., Martínez, H.P., Yannakakis, G.N., Jhala, A. - *Investigating the interplay between camera viewpoints, game information, and challenge*, Proceedings of the international conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE, Palo Alto, October, 2009
- Martínez, H.P., Jhala, A., Yannakakis, G.N. - *Analyzing the impact of camera viewpoint on player psychophysiology*, Proceedings of the international conference on Affective Computing and Intelligent Interaction, ACII, Amsterdam, September, 2009

1.5 Summary of Thesis

The remaining of this thesis is organized as follows:

Chapter 2 reviews the state-of-the-art in affect modeling, surveys studies from other fields related to the methods proposed in this dissertation, and summarizes the work done in several application domains.

Chapter 3 details the key phases of the method employed for affect modeling including the motivation for every phase. This includes the data collection (input modalities and annotation schemes), the feature extraction (unsupervised methods), the automatic feature selection and the preference modeling phases.

Chapter 4 introduces the two experimental surveys that generate the affect datasets used to validate the proposed methods. Additionally, the synthetic datasets used for the validation of the preference learning algorithms are presented.

Chapter 5 compares the efficacy of several preference learning algorithms based on artificial neural networks, support vector machines and Cohen's method and identifies the most adequate configurations for affect modeling and beyond. The algorithms are tested on both synthetic and real affect datasets.

Chapter 6 investigates the feasibility of automatic extraction of physiological features using deep learning and evaluates its suitability for affect modeling.

Chapter 7 introduces the use of sequence frequent mining for unsupervised feature extraction and showcases its advantages and limitations for affect modeling.

Chapter 8 summarizes the main findings extracted from the previous chapters and discusses their limitations and extensibility.

1.6 Summary

Computational models of affect have the potential to enhance and transform human computer interaction. Affective states are not easily elicited and delimited, while their measurable manifestations are ambiguous and exist in highly-dimensional, multimodal signals. This thesis contributes to the advancement of current affect modeling methodologies with new methods that build models on ordinal annotations of affective experiences and reduce and fuse multiple input modalities automatically without relying on hand-crafted feature extraction and selection solutions.

Chapter 2

Related Work

In this chapter, we survey work related to this thesis from different perspectives. The first section surveys a complete set of methods that have been applied to affect modeling. Advances on the technical aspect of this work within other research areas are analysed in the second section. Finally, the last section offers a brief review of affect detection applications, where the methods examined in this thesis can be applied.

2.1 Affect Modeling

Emotions and affect are mental and bodily processes that can be inferred by a human observer from a combination of contextual, behavioral and physiological cues. Part of the complexity of affect modeling emerges from the challenges of finding objective and measurable signals that carry affective information (e.g. body posture, speech and skin conductance) and designing methodologies to collect and label emotional experiences effectively (e.g. induce specific emotions by exposing participants to a set of images). Although the contributions of this thesis are mostly concerned with computational aspects of creating physiological detectors of affect, the signals and the affective target values collected shape the modeling task and, thus, influence the efficacy and applicability of dissimilar computational methods. Consequently, this section gives an overview of the field beyond the input modalities and emotion annotation schemes examined in the case studies. In particular, this section surveys studies representative of the four phases of the methodology proposed in this thesis for affect modeling: 1) eliciting and collecting observations of changes in affective states (Section 2.1.1) 2) defining feature sets to extract relevant bits of information from objective data signals (i.e. feature extraction; Section 2.1.2), 3) reducing the dimensionality of the feature sets (Section 2.1.3), and 4) creating models that map a feature set into a given affective target value (i.e. training models of affect; Section 2.1.4). For more extensive surveys of affect modeling methods, the reader is referred to several review articles Calvo and D'Mello (2010); Kleinsmith and Bianchi-Berthouze (2012); Zeng et al. (2009); Pantic and Rothkrantz (2003); Kivikangas et al. (2010); Wu et al. (2012).

2.1.1 Data Collection

Affect elicitation

Many different strategies have been proposed for the induction of affective states in humans. These strategies can be separated into three categories. The first consists of instructing

participants to act or pose affective states. This is a common practise in facial and body expression studies (Kleinsmith and Bianchi-Berthouze, 2012) but is unsuitable for other modalities such as physiology as participants can not produce on-request the typical physiological response to a given affective state. Furthermore, one can easily speculate that models built on such data will not be that reliable in real settings (e.g. when using any computer application) where emotional displays are not as exaggerated.

A second approach for one to be able to elicit affect consists of instructing participants to remember significant real-life events attached to certain emotional experiences. This method has been used in several studies (AlZoubi et al., 2009; Levenson, 2007; Picard et al., 2001; Calvo et al., 2009 among others); however, it has been argued that the exact experience of an emotion (including its physiological manifestations) cannot be re-experienced by mentally reenacting a past situation (Robinson and Clore, 2002; Galin, 1994; James, 1890). Thus, it appears that this is not an appropriate method to collect psycho-physiological data.

The third approach to affect elicitation consists of designing different experimental conditions aimed at eliciting dissimilar affective responses. The conditions could target specific affective states or simply produce dissimilar experiences. A popular method within this category relies on slide presentation of pictures labeled with levels of arousal and valence (IAPS Lang et al., 1999). A set of these pictures selected from different areas of the arousal-valence space are presented for few seconds, one after another potentially evoking dissimilar affective states (e.g. Alzoubi et al., 2011; Schaaff and Schultz, 2009). Other forms of passive stimuli (i.e. participants do not interact) are audio and video clips that can be presented in the same fashion as images (e.g. Koelstra et al., 2012; Bradley and Lang, 2000; Janssen et al., 2009). Stimulation time is usually longer for such multimedia content, increasing from 10 seconds that images are typically shown to several minutes that a complete song or scene can last. On the other hand, they have a relatively higher degree of ecology as emotions in real life often occur in response to dynamic external visual and auditory stimulation (Levenson, 2007). Similarly, a wide variety of interactive tasks have also been used to elicit emotions such as driving (Healey and Picard, 2005; Fernandez and Picard, 2003) and playing video-games (Martínez et al., 2013; Kapoor et al., 2007). In these tasks, the interaction information can be used as input to the affect model which can yield richer models.

For this thesis we selected two affect datasets collected using audio-visual stimuli in a lab as we believe that these user studies offer a good balance between the intensity of the emotions elicited and control over experimental variables and environment. In the first study several participants watch a sequence of music video-clips whereas in the second study the participants play a series of computer games.

Affect annotation

The key limitation of modeling affect at large is the inability to access directly the subjective feelings of the target experiment participant. In order to access that information with the highest possible accuracy, three basic *psychometric* approaches have been proposed: expert annotation, the subtractive method and self-reports. The first approach consists of one or several trained experts annotating the affective states that the participant is feeling during the experience (e.g. Devillers and Vidrascu, 2006; Bailenson et al., 2008; Karpouzis et al., 2013; D'Mello and Graesser, 2009). The main drawbacks of this approach are the high time-consumption, the inability of reporting *on the fly* (the observers will normally work over video recordings) and the need of expert annotators (Sanderson and Fisher, 1994).

Additionally, this method relies on visible expressions of emotion that the participant may (unintentionally) hide, fake or exaggerate. On the other hand, the main advantage of this method is that it does not disrupt participants' experiences. An increasingly popular variant of this technique consists of asking a large number of untrained observers over the web (i.e. crowdsourcing (Tsai et al., 2012; McDuff et al., 2012)). While this variant might reduce the cost and time of annotation, it may also reduce the quality of the data as it is highly complicated to control for the expertise and good will of the annotators (Mohammad and Turney, 2012).

The second approach consists of designing several tasks with the aim of eliciting different affective states or different intensities of the same states (e.g. a stressful task and a non-stressful task (Healey and Picard, 2005)). It is assumed that every participant experiences the emotions that the task has been designed to evoke. This approach is not very appropriate for rather abstract and complex states such as "fun" since it is e.g. non trivial to create a game that would be engaging for every participant. Otherwise, this method eliminates the need to re-evaluate the experiences which includes the time and budget overhead of expert observers and the disruption of questionnaires.

The third approach consists of asking directly the participant what she is feeling or has felt. This approach works only under the assumption that participants are aware of their affective state and able to remember it. According to Robinson and Clore (2002), participants would be able only to access their feelings directly if they report the emotion while happening. For instance, Kapoor et al. (2007) used a game on their experiments that featured a button to allow participants to report when they began to feel frustrated. However, this and other methods for reporting affect on-line (e.g. think-aloud) are disruptive of the experience (Nielsen et al., 2002). On the other hand, if the emotion is reported at the end of the experience, participants would need to reconstruct the affective experience by recalling relevant thoughts and event-specific details. Note, that the fact that the participant needs to rely on memories and belief, does not imply that reports of emotion are not reliable and; in fact, post-experience self-reports can potentially be the most reliable source of affective information (Clore, 1994; Diener, 2000; Watson, 2000) despite problems such as order effects (Chan, 1991). In addition, self-reports present a number of practical advantages over expert annotators that have made them popular among AC studies (e.g. Koelstra et al., 2012; Hernandez et al., 2011): they are a quick (to answer one question will take few seconds), on-line (answers are known immediately after or during the experience) and cheap (forced-choice questionnaires can be processed automatically).

Regardless of the approach, the affective state information can be given as an absolute or relative measure. Absolute values or ratings are regularly used to evaluate user experience along some dimension such as arousal or frustration. Participants are typically asked to express how they felt in a scale varying between two extremes as for instance in Likert-scales (Likert, 1932). The scale is usually characterized by numbers or qualitative adjective (e.g. "extremely" and "poor") indicating the direction of the scale and the intensity at each point (Watson and Clark, 1999). It is also frequent to replace those markers by visual representations as for example in the self-assessment manikins (Morris, 1995) for reporting arousal, valence and dominance (Mehrabian, 1995). Alternatively, Scherer (2005) proposed a 2-dimensional questionnaire that lays out several 5-point scales with different directions, aligned with the positions of different affective states in the arousal-valence space (Russell, 1980).

Relative measures or rankings/preferences are common in marketing studies under the name of *A/B testing* (Kohavi et al., 2009). In this context, pairwise self-reports can be used

to collect participants' preferences over two variants (A and B) of the same product. In the context of affect annotation, rankings and preference questionnaires require the participant to order a set of experiences by intensity of affective state felt, (e.g. the first game felt more frustrating than the second). However, these tools have only been explored scarcely (e.g. in Tognetti et al., 2010a; Yannakakis et al., 2010; Pedersen et al., 2010) in favour of ratings, despite of a number of known subjective biases (Viswanathan, 1993; Costner, 1969) and stronger order effects (Yannakakis and Hallam, 2011) of rating reports.

Both affect datasets used in this thesis include post-experience self-reports, DEAP using ratings and Maze-Ball using pairwise preferences, showcasing the applicability of the proposed methodology to both types of affect reports.

Measurable manifestations of affect

This thesis focuses on two modalities of input, namely physiology and (game) context. Physiology has been extensively investigated in relation to affect (Andreassi, 2000; Calvo et al., 2009 among many others). While a relation between physiology and affect is by now undeniable, the exact mapping is not yet known; the methodology proposed in this thesis provides new tools for advancing further the investigation of this relation. As for context information within the game domain, the number of studies is more reduced, yet they have shown that this modality is extremely valuable for affect detection, in particular when fused with other modalities such as physiology (e.g. McQuiggan et al., 2007; Martínez and Yannakakis, 2011b; Ravaja et al., 2005). In addition, this is the least obtrusive and cheap modality as it does not require any physical contact with the participant and can be recorded within the virtual environment (game or other application).

Other modalities that have been explored in related work but not touched upon in this thesis include facial expressions (Kapoor et al., 2007; Arroyo et al., 2009; Grafsgaard et al., 2011; Busso et al., 2004; Zeng et al., 2009), muscle activation (typically face) (Conati and Maclaren, 2009; Dennerlein et al., 2003), body movement and posture (Asteriadis et al., 2009; van den Hoogen et al., 2008; Kapoor et al., 2007; D'Mello and Graesser, 2009; Bianchi-Berthouze and Lisetti, 2002), speech (Vogt and André, 2005; Kannetis and Potamianos, 2009; Juslin and Scherer, 2005; Johnstone and Scherer, 2000; Banse and Scherer, 1996), brain interfaces (Rebolledo-Mendez et al., 2009; AlZoubi et al., 2009) and eye movement (Asteriadis et al., 2009). The methodology that we present is potentially applicable to all these modalities but we restrict our experiments to physiology and context as we believe they are more appropriate for the future development of affective technologies. Note that camera-based modalities (facial expressions, body posture and eye movement) require a well-lit environment often not present in home settings (e.g. when playing video-games) and they can be seen by some users as privacy hazards (as the user is continuously recorded). As for brain interfaces and muscle activation, the sensors are currently more invasive than physiological sensors that can be embedded in comfortable wrist bands[1], making physiological sensors easier to wear both in home settings and *in the wild*. Finally, speech is also a easy-to-access modality that do not require obtrusive sensors, but on the other hand it does not provide continuous data as users may be quiet during parts or the whole experience, e.g. some gamers are silent while playing games alone.

[1]Accessed November 14, 2012 `http://www.empatica.com/`

2.1.2 Feature Extraction

In the context of affect detection, feature extraction is referred to as the process of transforming the raw signals captured by the hardware (e.g. a skin conductance sensor, a microphone, or a camera) into a set of inputs suitable for a computational predictor of affect. The most common features extracted from unidimensional continuous signals are simple statistical features calculated on the time or frequency domains of the raw or the normalized signals (see Picard et al., 2001; Ververidis and Kotropoulos, 2004 among others). Examples of these features are average heart rate, maximum skin conductance and variance of the amplitude of a speech signal. More complex features inspired by signal processing methods have also been proposed by several authors. For instance, Giakoumis et al. (2011) proposed features extracted from physiological signals using Legendre and Krawtchouk polynomials while Yannakakis et al. (2008) used the approximate entropy (Pincus, 1991) and the parameters of linear, quadratic and exponential regression models fitted to a heart rate signal. Unidimensional discrete signals — i.e. temporal sequences of discrete labels, typically *events* such as clicking a mouse button or blinking an eye — are usually transformed with similar ad-hoc statistical features such as counts. The focus of this thesis is on unsupervised methods that can automatically derive features from the data, opposed to a fixed set of features that represent arbitrary characteristics of the signals.

Related work on recognition of affect based on signals with more than one dimension boils down to recognition of affect from images or videos of body movements, posture or facial expressions. In most studies, a series of relevant points of the face or body are first detected (e.g. right mouth corner and right elbow) and tracked along frames. Second, the tracked points are aggregated into discrete *Action Units* (Ekman and Friesen, 1978), gestures (Caridakis et al., 2011) (e.g. lip stretch or head nod) or continuous statistical features (e.g. body contraction index) which are then used to predict the affective state of the user (Kleinsmith and Bianchi-Berthouze, 2012). Both above-mentioned feature extraction steps are, by definition, supervised learning problems as the points to be tracked and action units to be identified have been defined a priori. While these problems have been investigated extensively under the name of facial expression or gesture recognition this thesis will not survey them broadly as the focus is on methods for automatically discovering new or unknown features in an unsupervised manner.

Deep neural network architectures such as convolutional neural networks are popular techniques for object recognition in images (LeCun and Bengio, 1995; Farabet et al., 2013) and have also been applied for facial expression recognition. In (Matsugu et al., 2003), CNNs were used to detect predefined features such as eyes and mouth which later were used to detect smiles. Contrary to the work presented in this thesis, in that study each of the layers of the CNN was trained independently using backpropagation, i.e. labeled data was available for training each level. More recently, Rifai et al. (2012) successfully applied a variant of auto-encoders (Bengio et al., 2007) and convolutional networks, namely contractive convolutional neural networks, to learn features from images of faces and predict the displayed emotion. The key differences of the application of DL in this thesis with that study reside on the nature of the dataset and the method used. While Rifai et al. (2012) used a large dataset (over $100,000$ samples; $4,178$ of them were labeled with an emotion-class) of static images displaying posed emotions, this thesis uses small datasets (224 and 880 samples, labeled with pairwise orders) with a set of physiological signal time-series recorded along an emotional experience. The reduced size of these datasets (which is on the same magnitude as datasets used in related psycho-physiological studies such as Kapoor et al.,

2007; Tognetti et al., 2010b) does not allow the extraction of large feature sets (e.g. 9,000 features in (Rifai et al., 2012)) as it would lead to affect models of poor generalizability. Furthermore, while the use of CNNs to process images is extensive, to the best of the author's knowledge, CNNs have not been applied before to process (or as a means to fuse) physiological signals. While extracting information from an image of a posed emotion is not an easy task, the image typically contains a sufficient number of distinct cues to identify the emotion. On the other hand, the relation between physiological signals and affect is more ambiguous and subtle, thereby finding distinct physiological components relevant for affect is arguably a harder task.

2.1.3 Feature Selection

As in many other machine learning applications, in affect detection it is common to apply dimensionality reduction techniques to the complete set of features extracted. A wide variety of feature selection methods have been used in the literature including sequential forward (Lee and Narayanan, 2005; Ververidis and Kotropoulos, 2004; He et al., 2009), sequential floating forward (Picard et al., 2001; Vyzas and Picard, 1998; Schuller et al., 2005), sequential backwards (Wagner et al., 2005; Giakoumis et al., 2012), n-best individuals (Yannakakis and Hallam, 2007), perceptron (Pedersen et al., 2010) and genetic (Tognetti et al., 2010a) feature selection. This thesis is not focused on the feature selection phase, we utilize sequential forward and genetic search to demonstrate how the proposed feature extraction mechanisms interact with quick local-search and slower global-search algorithms.

Fisher's projection (Krzanowski, 1977) and principal component analysis (PCA) (Wold et al., 1987) have been also widely used as dimensionality reducers on different modalities of human input for affect modeling (e.g. see Kim et al., 2004; Vyzas and Picard, 1998; Schuller et al., 2005; Lee and Narayanan, 2005; Busso et al., 2004; Charfuelan and Schröder, 2011 among others). An auto-encoder can be viewed as a non-linear generalization of PCA Hinton and Salakhutdinov (2006); however, while PCA has been applied in AC to transpose sets of manually extracted features into low-dimensional spaces, in this thesis auto-encoders are training CNNs to transpose subsets of the raw input signals into a *learned* set of features. With the application of DL in the AC domain we expect that relevant information for prediction can be extracted more effectively using dimensionality reduction methods directly on the raw physiological signals than on a set of designer-selected extracted features.

2.1.4 Model Creation

Before delving into the details of computational models of affect, it is worth mentioning that a large body of research in affect modeling does not make use of CI and ML tools whatsoever. Instead, correlation analyses between affective target values and a set of statistical features are often used as the key methodology in a vast number of publications (Nacke and Lindley, 2008; Mandryk et al., 2006; Rani et al., 2005; Hazlett, 2006; Drachen et al., 2010; Hazlett, 2006; Ravaja et al., 2005 among others). The outcome of this methodology is a statistical analysis of certain input signal characteristics with respect to a specific condition or feature (e.g. average heart rate is linearly correlated with self-reports of fun). On the other hand, the outcome of the methods used in this thesis consists of both linear and non-linear function approximators (or models) that map a number of input features to an output (affect estimation).

The selection of a method to create that model is strongly influenced by the dynamics of

the features (stationary or sequential) and the format in which training examples are given (continuous values, class labels or ordinal labels). Hidden Markov models (Grafsgaard et al., 2011; Fernandez and Picard, 2003), dynamic Bayesian networks (Kaliouby and Robinson, 2005; Fernandez and Picard, 2003) and recurrent neural networks (Kobayashi and Hara, 1993) have been applied for constructing affect detectors that rely on features which change dynamically. In the methodology presented here, automatic methods are used to reduce the resolution of temporal signals down to a set of features that can be fed to simple stateless models; these models allow typically for a simpler training and interpretation than complex continuous sequential predictors. To create models of affect based on stationary features, a vast set of *off-the-shelf* machine learning methods have been applied, irrespective of the specific emotions and modalities involved. These include linear discriminant analysis (Giakoumis et al., 2012), multi-layer perceptrons (Bailenson et al., 2008; Wagner et al., 2005), k-nearest neighbours (Lee and Narayanan, 2005; AlZoubi et al., 2009; Nasoz et al., 2004), support vector machines (Alzoubi et al., 2011; Kim et al., 2004; Garber-Barron and Si, 2012; Soleymani et al., 2012), decision trees (Heraz and Frasson, 2007; McQuiggan et al., 2007; Mcquiggan et al., 2008), Bayesian networks (Gunes and Piccardi, 2007), Gaussian processes (Kapoor et al., 2007) and fuzzy-rules (Mandryk and Atkins, 2007). On the other hand, From this large selection of stateless models, we focus our investigation on multi-layer perceptrons (MLPs) because they can approximate any continuous function, characteristic that we believe is key to investigate the unknown mappings of affect. In addition, the complexity of MLPs can be kept to levels in which is possible to interpret the learned function. We compare MLPs against support vector machines which offer a reliable baseline given their popularity in a large number of domains.

Different variants of the aforementioned methods have to be used depending on the nature of the prediction target. In all above-mentioned studies, the prediction targets are either class labels (e.g. frustrated and happy) or ratings. Class labels are treated as nominal variables and ratings are also typically transformed into nominal variables (e.g. in a scale from 1 to 5 of stress, values above or below 3 correspond to the user at stress or not at all, respectively (Hernandez et al., 2011)). It is also common to treat continuous ratings as real-valued variables (e.g. Nicolaou et al., 2011) although this is a questionable practice since the subjective biases in human ratings make them ordinal variables. Alternatively, it is possible to treat ratings as ordinal variables or collect the prediction targets as rankings (e.g. the first experience is more frustrating than the second). In this thesis we focus on methods that train computational models using this type of data. These methods, known as *preference learning* methods, allow us to avoid binning together ordinal labels and to work with comparative questionnaires which provide more reliable reports of affect data compared to ratings (Yannakakis and Hallam, 2011).

Preference learning methods and comparative (rank) questionnaires have been scarcely explored in the AC literature, despite their well-known advantages. To the best of the authors knowledge, PL methods have not been used before within AC studies to model ratings. Applications to model ranks can be found in Tognetti et al. (2010a) which applies linear discriminant analysis to learn models of preferences over game experiences based on physiological statistical features and comparative pairwise self-reports (i.e. participants played pairs of games and ranked games according to preference). On the same basis, Yannakakis et al. (2008) and Yannakakis and Hallam (2008) trained multi-layer perceptrons via genetic algorithms (i.e. *neuroevolutionary preference learning*) to learn models for self-reported fun using physiological and behavioral data, and pairwise self-reports. As mentioned before, this thesis focus on artificial neural networks (MLPs to be more precise), and more specifi-

cally, it analyzes the impact of different error functions as this is the element in the training process that enables learning models from ordinal data.

2.2 Data Modeling

Data modeling consists of finding the underlying mathematical model that best describes a set of observations. This thesis addresses the problem of finding the mapping between physiological and behavioral data and a number of affective states of users. For that purpose, methods are borrowed from three independent but overlapping research areas: machine learning, computational intelligence and data mining.

2.2.1 Machine Learning

Machine learning (Mitchell et al., 1997) is a branch of artificial intelligence concerned with computational methods for *automatic learning* from data. This involves, for example, methods for autonomous agents to develop strategies to solve a given task. Related to data modeling, ML also offers algorithms to find unknown relations among the variables or features in a given dataset; these algorithms can be categorized in two groups: *supervised* and *unsupervised* learning.

Supervised learning

In supervised learning, a function or model is learned from a set of training examples. These examples associate a set of input features with a set of target outputs. The learning process synthesizes the mapping between the inputs and the output. Depending on the nature of the output, supervised learning algorithms can be classified as *regression* if the output is a continuous value, *classification* if the output is an item from a finite set (class) and *preference learning* if the output is an ordered set, ordinal class (rank) or ordinal relation. This thesis focus on the third type of supervised learning, preference learning, and in particular on learning from ordinal relations, a setting also known as *object ranking* (Fürnkranz and Hüllermeier, 2010b). The basis of the algorithms used in this thesis, however, has been developed for classification and regression tasks and, therefore, we offer an overview of such methods in the following section.

Classification and regression: both artificial neural networks and support vector machines are popular and well-known methods in pattern recognition (Bishop, 1995; Schölkopf and Smola, 2001). Support vector machines were proposed by Cortes and Vapnik (1995) and have since then been applied on numerous domains — e.g. text categorization (Joachims, 1998), spam classification (Drucker et al., 1999) and gene selection for cancer classification (Guyon et al., 2002). The original SVM model was presented for binary classification only but extensions for regression have been proposed (Drucker et al., 1997) as well as different optimized versions of the training algorithm (Osuna et al., 1997; Joachims, 2006).

The first artificial neural network models were proposed by McCulloch and Pitts (1943) and Rosenblatt (1958) but discredited due to the limited representational power of the basic model (only linear functions) and lack of general training algorithm for more complex models (Minsky and Seymour, 1969). ANNs re-emerged years later after Werbos (1974) and Rumelhart et al. (1986) popularized a gradient-descent method known as *backpropagation* (BP) to train feed-forward neural networks. This training method applied on feed-forward

fully connected networks known as multi-layer perceptrons has dominated the study of ANNs in ML since then. Backpropagation training of MLPs has been applied to an endless list of tasks from the prediction of thunderstorms (Gardner and Dorling, 1998) and bridge damage detection (Pandey and Barai, 1995) to affect classification (Bailenson et al., 2008). Although, in theory, an MLP can approximate any function (Hornik et al., 1989), in practice backpropagation is limited by the size of the MLP and therefore the complexity of the approximated functions. *Time-delay* neural networks (TDNN) (Waibel et al., 1989) represent a variation of feed-froward networks whose restricted structure makes them suitable for pattern recognition on 1-dimensional signals. The main application of TDNN is sound classification tasks such as music genre classification (Hamel et al., 2011) and phoneme recognition (Hampshire et al., 1990) but several studies have also applied them on electro-encephalogram signals to predict epileptic seizures (Mirowski et al., 2008) or recognize event-related potentials (Cecotti and Graser, 2011). Convolutional neural networks (LeCun et al., 1989) represent a generalization of the same model to data of any number of dimensions and have been applied successfully to several computer vision tasks (LeCun et al., 1998; Szarvas et al., 2005; Nebauer, 1998). In this thesis CNNs are used for the first time to extract features from physiological signals that are then fed to MLPs, SVMs and Cohen's models that predict user affective states.

Object ranking: popular families of object ranking OR algorithms include Gaussian processes (Chu and Ghahramani, 2005; Nielsen et al., 2011; Abbasnejad et al., 2011), SVMs (Joachims, 2002; Radlinski and Joachims, 2005; Bahamonde et al., 2007) and ANNs (Pedersen et al., 2009). Most work in the field of object ranking focuses on the application of one family of methods to new domains or suggests modifications of existing algorithms for performance improvement. This thesis instead examines several modifications to ANN training and presents an empirical comparison which includes SVMs and Cohen's method across dissimilar datasets. Kamishima et al. (2005) reported a similar empirical work where they compared two different SVM approaches, Cohen's method and Empirical Rank Regression (ERR) across several sets of synthetic and real data. That work, however, did not consider any ANN based methods which are central to this thesis, and did not make use of any affect dataset.

Neural networks were first used to solve an order learning problem by Tesauro (1989). Tesauro was concerned with computer players that could learn to play backgammon and he proposed a method to train a MLP using pairwise preferences. The network received as inputs the final state of the game board after each one of two alternative moves that the player had to choose from. Backpropagation with a standard error function, namely sum of squared errors (SSE), was used to learn which one of the two moves was optimal. In this thesis we test several alternative error functions that allow artificial neural networks training with a single object (or sample) as input.

Caruana et al. (1996) introduced a different neural network approach, namely *RankProp*, which utilizes a 2-phase modification of backpropagation to learn a binary dataset: at the first phase of each epoch a real-valued rank is estimated for each object; at the second phase standard backpropagation is applied with SSE as the error function to learn those ranks. Ranks are estimated by ordering the training samples , first, based on their categorical binary label and, second, based on the output values that the current ANN yields for each one of them. The normalized position in the order is then used as a target value for the regression phase. Note that this is not an object ranking task per se — as the final goal is

binary classification — but it can easily be extended using ordinal labels during the ordering step. This approach assumes that the ordinal labels correspond to a global ordering. In (Caruana et al., 1996) the target function is *risk of death* and one can safely assume that all dead patients had a higher risk of death than patients alive; however, this assumption does not hold true for any domain. In particular, when we model user preferences given as ratings in a e.g. 5-point Likert scale from 'poor' to 'very good' (Likert, 1932), personality biases, cultural background, temperament, and interests may have significant effects on the global ordering as 'Average' for one person may indicate a higher preference than 'good' for another, or for the same person an object shown twice may be first rated as 'average' and then as 'good' due to order and inconsistency effects (Yannakakis and Hallam, 2011) (see Section 3.1.1 for more details). User ordinal reports are central in this thesis and consequently we examine error functions that bypass the need for assumptions about global consistency of affect annotations.

Crammer and Singer (2001) proposed a method called *PRank* to train single layer perceptrons on a set of objects with explicit numeric ranks. They reported experiments on a synthetic and on a real dataset and showed that their approach could outperform a multi-class classification and standard regression algorithms for training perceptrons. In this thesis on the other hand, models are not trained to predict the exact numerical rating values; instead the aim is to train models that predict relative orderings of objects. Furthermore, the algorithms described here can be used to train arbitrary network topologies.

Burges et al. (2005) proposed one of the error functions used in this thesis (cross-entropy) and compared it against RankProp, PRank and Online Aggregate Prank-Bayes Point Machine (OAP-BPM) (Harrington, 2003) across several synthetic and real datasets. They showed that both single layer perceptrons (SLPs) and MLPs trained using their error function could outperform those other methods. The rank margin error function (also examined in this thesis) was used in (Bai et al., 2010; Grangier and Bengio, 2005) to train quadratic models for document retrieval tasks and compared across several real datasets against other methods designed for that same task (e.g. OKAPI (Robertson et al., 1995) and Latent Semantic Indexing (Hofmann, 1999)). The regularized least-squares function, also evaluated in this dissertation, was introduced by Pahikkala et al. (2009) and compared against SVMs on several datasets. In this thesis, those error functions are further validated on different synthetic and real datasets and their effectiveness is also tested when combined with a global optimization algorithm.

Yannakakis et al. (2009, 2010) studied the sigmoid error function to train artificial neural networks using genetic algorithms on dissimilar real datasets and compared them against large margins (Fiechter and Rogers, 2000), meta-large margins and Gaussian processes (Williams and Rasmussen, 1996). This thesis offers the first attempt to compare that method against local search algorithms (backpropagation) and other error functions for neural network training.

To the best of our knowledge, this dissertation is the first study that extensively compares various SVM kernels and artificial neural networks in the context of object ranking. Furthermore, none of the studies surveyed above investigated the effect of error functions and error function margins on OR model performance, which are central for using ANNs for object ranking.

Unsupervised learning

Unsupervised learning deals with problems such as finding distinct groups (clusters) of data samples and learning the probability distribution that generates a dataset. In this thesis unsupervised learning is applied to generate feature extractors that process large data samples transforming them into suitable input sets for supervised learning. To that end, *auto-encoders* (Bengio et al., 2007) are used to *pretrain* the layers of a convolutional neural network that reduces physiological signals into a manageable set of features (see Section 3.2.1). Pretraining in neural networks can be interpreted as a method to find a good initial configuration that facilitates supervised learning. This phase has allowed researchers to train efficiently large hierarchical models — referred to as *deep architectures* — which otherwise yield poor results (Bengio, 2009). This method was first proposed by Hinton et al. (2006) who pretrained layered feed-forward neural networks using restricted Boltzmann machines (Freund and Haussler, 1994). Since then a myriad of studies have proposed improvements and alternative methods for pretraining deep architectures including several variants of auto-encoders (Rifai et al., 2011; Vincent et al., 2010, 2008). The main application domain of these methods is computer vision tasks such as object and scene recognition (Kiros and Szepesvari, 2012; Ballan et al., 2012; Nair and Hinton, 2009) where they nowadays represent the state-of-the-art. This thesis does not contribute to theoretical advancements on these methods but applies auto-encoders to an unexplored domain: affect recognition from physiological signals.

2.2.2 Computational Intelligence

Computational intelligence focuses on nature-inspired algorithms such as artificial neural networks and genetic algorithms (GAs) (Goldberg, 1989) which can solve computational problems that cannot be easily formulated mathematically. Genetic algorithms consist of a global search algorithm that uses operators inspired by theories of natural evolution. GAs are used in this thesis to train ANNs as an alternative method to backpropagation. The combination of GAs and ANNs, known as *neuroevolution* (NE) (Moriarty and Miikkulainen, 1997), is a powerful tool that has generated solutions to learning problems where backpropagation is intractable. Although in this thesis we employ the basic form of neuroevolution that only adjusts the weights of a neural network, more advanced methods to evolve also the network topology are available (Stanley and Miikkulainen, 2002).

2.2.3 Data Mining

The data mining field covers methods applied to discover patterns in large datasets of data. These include, for example, the ML and CI methods presented above but may also include sequence mining methods such as *frequent sequence mining*. This method is used in this thesis to extract features from multiple modalities of user data for affect prediction.

Sequence classification using frequent patterns

Frequent sequential patterns are typically mined to detect interesting common trends in the data and discover association rules, correlations and other relationships (Han and Kamber, 2006) but they can be applied also for the classification and clustering of sequences.

Protein homology detection and classification is one of the most popular sequence mining tasks. In (Ben-Hur and Brutlag, 2003), a set of relevant subsequences (motifs) are predefined

and each protein is represented as a vector of attributes; each attribute represents the number of occurrences of a motif in the represented protein (generally, each motif will either occur once or not occur). In (Ferreira and Azevedo, 2005) a naïve Bayes classifier identifies the most probable family to which a protein belongs to using as inputs the number and average length of the frequent subsequences shared within each of the protein families.

One of the many differences between the task of classifying proteins and processing multimodal signals is the temporal nature of the signals: in a protein string every pair of consecutive elements has a distance of 1 unit while in a signal the time distance between two elements is derived by the sampling rate. This difference has an impact on the structure of sequences (e.g. in multiple time-series sequences two events might occur simultaneously), the matching conditions of sequential patterns and the procedure to match sequential patterns (e.g. in non-temporal sequences the number of gaps between elements can be constrained whereas in a temporal sequence the time between elements is constrained instead).

Lesh et al. (1999) define an effective method to mine features for sequence classification. This method consists of mining all sequential frequent patterns and prune those that are either not distinctive of one of the target classes or correlated with a pattern already selected. This pruning stage is necessary since frequent mining can produce an enormous amount of features that cannot be efficiently handled by a classifier. In this thesis, a similar approach to (Lesh et al., 1999) is used but the set of frequent sequential patterns is reduced by automatic feature selection which searches for the combinations of sequences that are more relevant for predicting a target output (i.e. affective state in the case studies presented).

2.3 Application Domains

The methodology investigated in this thesis will allow researchers to create models of affect that can potentially lead to a better understanding of human emotions. While studying human affect is laudable on its own, there are also a number of practical applications for creating affect detectors. Models of affect can be used to express or manifest felt emotions to others; recipients of these manifestations could be for instance professionals helping autistic children or computer applications adapting and personalizing their content. This section offers a brief review of four main domains where affect detectors have been explored.

2.3.1 Adaptive Digital Games

Computer games, opposed to traditional music and video content, are highly interactive media that continuously react to the users' input. This interactivity can naturally accommodate mechanisms for on-line real-time adaptation of content aimed at manipulating player experience (Yannakakis and Togelius, 2011). Some commercial games such as *Left 4 dead* (Valve, 2008) and *Mario Kart 64* (Nintendo, 1996) include mechanisms to adapt the difficulty of the game but player experience is more complex that just challenge. Affect detection can enable adaptation mechanisms that target other aspects of the experience such as frustration (McQuiggan et al., 2007) and engagement (Gilleade et al., 2005; Hudlicka, 2009).

Affect detection from physiology in games has been explored in a large number of studies (Tijs et al., 2008; Nacke and Lindley, 2008; Mandryk et al., 2006; Mandryk and Atkins, 2007; Rani et al., 2005; Tognetti et al., 2010b; Drachen et al., 2010; McQuiggan et al., 2007 among others). Nevertheless, most of these studies use games as a test-bed to elicit emotions and

in none of them further work has been done towards affective adaptation. At the moment of writing, the only existing study that adapts the game content based on affect-physiological models can be found in (Yannakakis, 2009); in that study a model of *fun* based on heart rate features is used to adapt several parameters of a physical game for children. It is also worth mentioning that few games use physiological sensors, not to adapt the game based on an affect model, but as an active control. For instance, *The Journey of Wild Divine* (Wild Divine, 2001) is a game designed to teach relaxation exercises and it uses blood volume pulse and skin conductance to evaluate the performance of the player.

The second modality used in this thesis, game context, have been used in several studies to predict different affective states and other dissimilar mental states relevant for playing experiences (Pedersen et al., 2010; Shaker et al., 2010; Robison et al., 2009 among others); on the other hand, to the best of the authors' knowledge, this technology has not been yet exploited in commercial games.

Finally, the fusion of physiology and game metrics has been explored in a small number of studies, typically by analysing the physiological responses to game events (Conati and Maclaren, 2009; Hazlett, 2006; Ravaja et al., 2005) but also using physiological and gameplay statistical features (Mcquiggan et al., 2008; Martínez and Yannakakis, 2010). In addition to the other examples, the applicability of these models has not been yet explored, and no studies or commercial games using them exist.

In all, affect modeling has been explored in games but attempts to create affect-driven adaptation mechanisms are still scarce. The methodology presented in this thesis facilitates is well-suited to create models to support these mechanisms the development of these adaptation mechanisms, as it offers an automatic method to create accurate real-time affect detectors.

2.3.2 Adaptive Music and Video Applications

Audio-visual media is a powerful elicitor of emotions in humans (Lundqvist et al., 2009; Levenson, 2007; Nasoz et al., 2004). Just as with games, an entertainment application can use affect models to detect those emotions and drive user experience. Within music, the relation between emotion and physiology has been extensively studied (Lundqvist et al., 2009; Grewe et al., 2007; Khalfa et al., 2002 among others) but the application of affect models in this domain is still limited. Janssen et al. (2009) worked on *affective play-lists* which automatically select the next song to be played relying on predictions of the listeners' mood. A similar application can be envisaged for a video or movie recommender system that predicts which clips are more suited for the viewer. However, research is still focused on the affect detection phase (Soleymani et al., 2008; Koelstra et al., 2012; McDuff et al., 2011; Silveira et al., 2013; Fleureau et al., 2012) and physiological video-recommender systems have not been yet exploited. While context information is not as easily accessed in this domain, the methodology studied in this dissertation can be used to build physiological models of affect that could enable all the applications mentioned above.

2.3.3 Intelligent Tutoring Systems

Confusion, anxiety and frustration are cognitive and affective states with a direct impact on students' learning process and outcome (Picard et al., 2004; Schwarz, 2000). Consequently, affect detection is becoming increasingly important in the intelligent tutoring community (Robison et al., 2009). The basic idea is that if a virtual tutor is capable of detecting the

affective state of the student, it can react to it which would enhance the learning experience (e.g. minimizing frustration) and possibly improve the learning outcomes.

As for other application domains, research has mostly focus on the detection phase (Calvo and D'Mello, 2011), evaluating dissimilar methods to model student confusion (Grafsgaard et al., 2011; Hussain et al., 2011), frustration (Conati and Maclaren, 2009; McQuiggan et al., 2007) and attention (Qu et al., 2005). An example of virtual tutors that react to automatically detected affect can be found in (Robison et al., 2009). That study presents a game-based virtual environment that monitors the student's actions and provides empathic feedback according to the detected affective state. They showed that, for instance, feedback targeting to reduce detected frustration can indeed help students.

Even when tutoring systems are not built as a game, they produce a large amount of information similar to the game metrics (e.g. actions of the student); thus, all the methods that are evaluated throughout this thesis can be utilized in intelligent tutoring systems without any modification.

2.3.4 Health Technologies

Nowadays, a significant part of the worlds population is afflicted by depression and stress-related illnesses, which are directly connected to emotion and moods. Thus affect detection is key for the prevention and computer-based treatment of these affections. Among these illnesses, post-traumatic stress disorder (PTSD) has attracted a lot of attention. Pedersen et al. (2012) performed a representative study in this area; they created a game-based tool for treating PTSD based on exposure theory. Physiological signals were recorded from patients playing the game in order to develop future versions in which the game is personalized to the patient's level of stress.

Another application of affect detection to health technologies is related to syndromes such as autism that involve difficulties processing or expressing emotions. There have been a large body of studies in AC research towards developing tools to help parents, teachers and carers of children with autism (Picard, 2009; Liu et al., 2008; Kaliouby et al., 2006). These tools detect the affective state of the children and communicate it to themselves or others, enhancing communication.

An additional application to health technologies has been explored in relation to telemedicine. In this particular domain, emotion is not at the core of the treated illness but it is regarded as an important element in the doctor-patient communication. Detecting the affective state of the patient can help the doctor to better diagnose or simply better interact with the patient. This enhanced communication can improve patient's satisfaction and yield faster recoveries. Lisetti et al. (2003) developed such a system, in which the affective state of the patients were predicted from physiological signals and communicated to the doctor.

From a computational perspective, this domain is not different from others in which context information is typically not available and only physical modalities such as physiology can be recorded. In this respect, the methods presented on this thesis regarding physiological models of affect can aid the development of better affect-aware health technologies.

2.4 Summary

This thesis proposes a set of computational methods to create models of affect. This chapter surveyed the related work from three perspectives: methods used in affect modeling, research performed in other areas around the methods used in this thesis, and application

domains for the methods examined. The next chapter, outlines the general methodology proposed for modeling affect and describes the key components in detail.

Chapter 3

Methodology

As it was outlined in Chapter 1, this thesis addresses several challenges in affect modeling. In this chapter the set of computational methods proposed to address those challenges are described. These different methods can be integrated into a complete methodology for affect modeling which is depicted in Figure 3.1. The four phases illustrated in the figure represent the typical machine learning process where (1) data is collected, (2) data is transformed into a set of suitable inputs for a computational model (i.e. feature extraction), (3) irrelevant inputs are removed (i.e. feature selection), and (4) model is learned from the data. While this process is common in ML, alternative computational models not requiring the phases of feature extraction and selection also exist. In particular, hidden Markov models (Grafsgaard et al., 2011), dynamic Bayesian networks (Kaliouby and Robinson, 2005) and recurrent neural networks (Kobayashi and Hara, 1993) could potentially receive the raw signals as inputs and predict the affective state once the whole sequence of values is processed. However, in practice quantization or reductions similar to feature extraction are required when the length of the sequences is large (e.g. Jiang et al., 2011).

The first phase, *data collection*, is crucial as the whole process relies on it. Capturing real world phenomena into a format accessible for a computer is a complex task in many domains; capturing the affective state of a human is arguably amongst the hardest data collection tasks as objective measurements do not exist — a human is always required to label affective states in one way or another. This thesis does not investigate different data collection mechanisms but focus on modeling methods that rely on ordinal annotations of affect. In Section 3.1.1 we discuss the advantages and limitations of different protocols used to gather ordinal annotations in this user studies used in this thesis, namely ratings and rankings. In addition to the affect labels, the data sample must contain the information cues that will be provided as inputs of the model. This information is typically collected through cameras, force and movement sensors, and physiological sensors which generate temporal sequences of continuous values or images that span across the monitored experience. Additionally, if the goal of the model is to infer affect while a user interacts with a specific system, several internal variables of the system, referred to as context, can be recorded and provided to the model as continuous signals or as sequences of discrete labels — typically, events corresponding to actions or internal changes (e.g. mouse click or loading screen completed). This thesis focuses on physiological signals and context information; Section 3.1.2 surveys both input modalities in detail. Often, an additional phase is used to reduce the noise on the signals and remove artifacts, specially for physiological signals. We do not make use of this treatment because we aim at creating a methodology with as few steps as possible

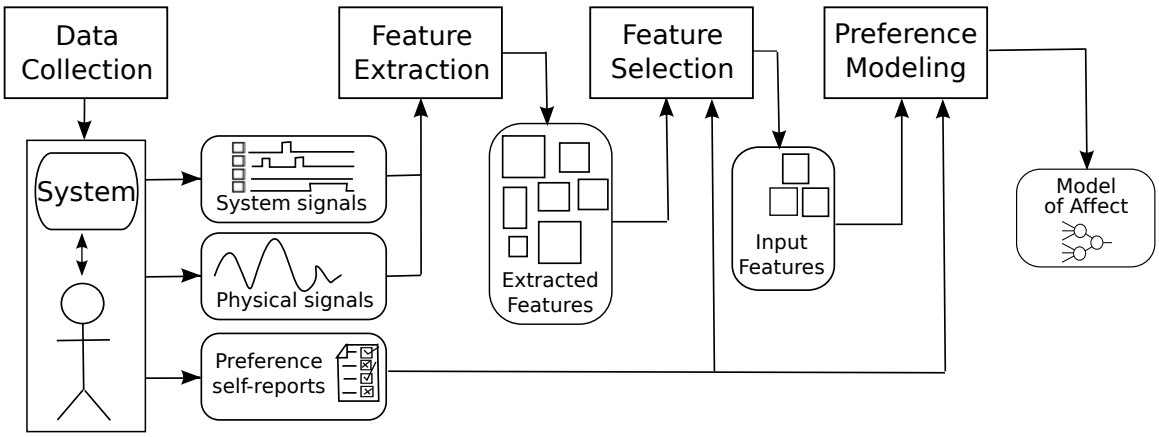


Figure 3.1: Methodology for affect modeling. Information from the user and the system is collected while the user interacts with the system. That information (time series or static data) is transformed during the feature extraction stage into a set of features. The set of features is reduced by means of automatic feature selection eliminating the least relevant ones. Finally, preference learning is used to learn the mapping between the set of selected features and the ordinal affect annotations (e.g. preference self-reports).

and as automated as possible. Therefore we validate that the feature extraction, feature selection and modeling phases can learn from untreated input signals.

The second phase, *feature extraction*, transforms the data collected from the sensors and the system into a set of features that can feed a computational model of affect. While this phase is typically performed through a number of *ad-hoc features* which are proposed by the researcher, this thesis proposes two methods to learn the features automatically from the collected data. The first method applies concepts from *deep learning* (Bengio et al., 2012) to learn features from different physiological signals. The second method, in a complete different manner, utilizes *frequent sequence mining* (Srikant and Agrawal, 1996) principles in order to extract features across dissimilar and discrete input modalities. Both methods are described in detail in Section 3.2.

The extraction phase often yields a number of features which is too large for ML algorithms to find meaningful and accurate mappings in the data, which motivates for the use of dimensionality reduction mechanisms. Automatic *feature selection* (Dash and Liu, 1997) is a popular category of methods that select the most relevant features according to some criterion, removing those that are (potentially) useless and redundant. In Section 3.3 the two FS approaches used in this dissertation are reviewed.

Finally, the mapping between the selected extracted features and the affect annotations is drawn by means of *preference learning* (Fürnkranz and Hüllermeier, 2010a) in the last phase of the methodology (see Figure 3.1). As the method relies solely on the data collected, the model does not depend on intermediate affect representations such as the arousal and valence dimensions (Russell, 1980) and maps extracted features directly into a numerical estimation of the target affective state. The general problem of PL is formalized in Section 3.4 along with the methods examined in this thesis, namely artificial neural networks, support vector machines and Cohen's method.

3.1 Data Collection

The process of creating models of affect by means of supervised learning relies upon a dataset that contains a mapping from measurable cues of users' context, behavior and physiological state to their affective states. Such a dataset is usually collected through controlled experiments where a group of participants is exposed to distinct experimental conditions aimed at eliciting different affective states. Through a number of sensors, cameras, questionnaires and other input devices, a sample of the prospect inputs and the expected outputs of the model is taken. The work described in this thesis did not include any data collection and instead we made use of two publicly available affect datasets that include physiological signals. Nevertheless, we describe, for the sake of completeness, the affect elicitation and *measurement* strategies used on these datasets (Section 3.1.1) and the characteristics of the input modalities recorded (Section 3.1.2).

3.1.1 Model's Output: Self-reports of Affect

The two datasets used in this thesis present a user study in which participants are exposed to audio-visual stimuli to elicit different affective states. In the first study, DEAP, a group of participants watches a sequence of 1-minute-long music video-clips whereas in the second study, Maze-Ball, another group plays several 90-second-long computer games. Both videos and games are able to elicit a wide range of affective states and they are the stimuli with a higher ecology within controlled settings (compared for instance to sliding images) as emotions in real life often occur in response to dynamic external visual and auditory stimulation (Levenson, 2007).

As generally, each individual can react differently to the same stimuli, it is required that the affective states felt during the experience are annotated. In both datasets used in this thesis, the experiences are annotated using post-experience self-reports. This approach works only under the assumption that participants are aware of their affective state and able to remember it. According to Robinson and Clore (2002), an affective state cannot be directly remembered, and the participant needs to rely on recalling relevant thoughts and event-specific details, and beliefs. Note, that the fact that the participant needs to rely on memories and beliefs, does not imply that reports of emotion are not reliable and; in fact, according to many authors post-experience self-reports can be the most reliable source of affective information (Clore, 1994; Diener, 2000; Watson, 2000) despite problems such as order effects (Chan, 1991). A limitation of post-experience reports is that, generally, only one annotation is available for the whole experience (the 1-minute video or 90-second game) but on the other hand they are not disruptive of the experience (as during-experience reporting protocols such as think-aloud (Nielsen et al., 2002)) and they are faster and cheaper than using expert annotators or crowdsourcing studies (Sanderson and Fisher, 1994).

Regardless of the approach, the affective state information can be given as an absolute or relative measure. Absolute values or ratings consist of choosing a value within a given interval to express the intensity of the affective state. The intensity may be expressed as a nominal value (e.g. the experience was frustrating or not) or using a more detailed scale (e.g. from very frustrating to not at all frustrating or from 1 to 100). The alternative approach, relative measures or rankings/preferences, require the participant to order a set of experiences by intensity of affective state felt, (e.g. the first game felt more frustrating than the second). At first glance, one may think that ratings are clearly superior because

(1) they require less experimental conditions (a comparison is not required) and, (2) they provide a numerical measure of the intensity. However, rating annotations offer ordinal information just like rankings due to the subjectivity of human reports of affect (Stevens et al., 1946). First, the difference among rating items is unknown, e.g. in a 5-point scale the difference between 1 and 2 is smaller that between 4 and 5 if the user is reluctant on using the final value of the scale. Second, ratings suffer from a number of reporting biases that makes the intensity value inaccurate (Viswanathan, 1993; Costner, 1969). These biases include variability of the scale across time (e.g. while rating 40 videos using a 5-point scale, the meaning of '2' in the first and the fortieth video may not be exactly the same), and variability of the scale across participants (e.g. in a 5-point scale a person may never use the extreme values 1 and 5 while other may only use those). Thus, ratings do not provide any advantage over rankings because (1) an appropriate use of ratings requires more than one experiment as the values are relative and not absolute, and (2) the information regarding the intensity cannot be easily utilized given that the subjective distance among items is unknown and varies along time and across participants. In addition, results reported on a recent study (Yannakakis and Hallam, 2011) suggest that ratings rankings do not only provide similar information, but also ratings can lead to significant order effects compared to rankings. More in detail, that study reported that in two different datasets, rating annotations presented significant recency effects (tendency to give higher ratings to later experiences) while pairwise preferences did not.

The modeling methods used in this thesis are designed for ordinal data independently of the reporting method; thus we validate them using both ratings and rankings. The first dataset used, Maze-Ball, was collected using pairwise preferences via a 4-alternative forced choice questionnaire (4-AFC). In a 2-AFC, the participant is asked to report whether a particular affective state (e.g. frustration and anxiety) was felt stronger during condition A or B. The 4-AFC adds two additional possible answers — that either the affective state was felt with equal strength in both conditions or that it was not felt in either of them — in order to minimize unreliable reports introduced when the two conditions do not induce affective states that are clearly different. The second dataset, DEAP, was collected using continuous ratings to annotate different dimensions of the experience (e.g. arousal and valence). Note that, as mentioned above, despite each rating is a continuous value, its meaning is solely ordinal. Therefore, we can represent the ratings as pairwise preferences without loss of information: if for every pair of data samples (videos) we create a pair in which the sample with a higher rating is preferred over the other sample, none of the ordinal information in the ratings is lost as it could be reconstructed from the pairwise preferences. However, in order to minimize the variability effects, we discard the following pairs:

- Pairs including ratings from two different participants are not included; thus, bypassing problems due to the variability across participants.
- Only pairs containing consecutive videos are included; thus, aggressively removing noise due to the variability along time of the rating scale.
- Pairs that present rating differences below one are considered unclear preferences, which are not used for learning. This threshold is based on the minimum distance in the visual scale displayed on the questionnaire; in this particular experiment, while participants reported a continuous number between 1.0 and 9.0, nine separate discrete items were displayed (rendering a minimum distance of one unit).

By removing pairs that potentially present noisy or even random data (e.g. when comparing ratings across participants) we are improving the quality of the ground truth; hence, automatic learning methods can potentially build more reliable models of affect.

3.1.2 Model Input: Affect Manifestations

Two sources of information are considered in this thesis as inputs to models of affect, namely physiological signals and context information.

Physiology

Research on psycho-physiology (Andreassi, 2000) has largely studied the unconscious physiological changes in the human body — including changes in body temperature, blood pressure and skin perspiration — as responses to mental states including affective and cognitive states (e.g. stress, anxiety and attention). Nowadays, the existence of such link is undoubtedly accepted across researchers in any related field. These unconscious body changes are rooted in the autonomic nervous system (ANS) which is the subset of neurons in the peripheral nervous system. In turn, the ANS has been traditionally divided in two systems: the sympathetic and parasympathetic systems responsible for preparing the body for a physical activity in response to a threatening situation (fight-or-flee response) or for resting, respectively. An activation of the sympathetic nervous system (SNS) can increase heart rate, decrease blood flow to extremities and increase activity of sweat glands. Complementary, an activation of the parasympathetic nervous system (PSNS) produces decreased HR and has not known direct effects on the sweat glands or blood flow in extremities.

While several other effects are known and have been studied in relation to emotion (e.g. salivary secretions are affected by both systems and have been studied as indicators of stress and anxiety (Bohnen et al., 1991; Graham et al., 1988)), skin and cardiac activity changes are particularly interesting as they can be easily measured through sensors attached to the fingertips or the wrist, providing feedback in real time to an affect detector. In particular, the activity of the sweat glands can be collected attaching electrodes to two fingers of the same hand and measuring the *skin conductance* between them[1]. In turn, variation of blood flow can be measured from a fingertip using a *photoplethysmoraph*[2] that obtains the *blood volume* or *pulse volume* (more commonly referred to as *blood volume pulse*). Additionally, heart rate and other heart rate variability (HRV) signals such as *RR intervals* (i.e. time between consecutive heart beats) can be derived from BV and BVP using mathematical peak detector algorithms. Samples of these signals are shown in Chapter 4 as they are included in the two affect-related datasets used to test the proposed methodology (Figure 4.7 and Figure 4.8).

Physiology is a highly objective modality in the sense that a regular person cannot consciously affect her physiological state[3], thus she cannot hide an emotional response. A disadvantage is the intrusiveness of the sensors and their unavailability at large scale, although positive progress has been done on both fronts in recent years. The number of

[1] Two basic methods exist consisting of keeping the voltage constant and measuring the fluctuations of electric current or viceversa.

[2] This sensor transmits light to the skin and measures the reflected light variation.

[3] it has been suggested that through operand conditioning techniques it might be possible to alter one's own level of ANS activity (Andreassi, 2000)

commercial sensors to capture these responses is increasing (e.g. *IOM*[4], *ProComp Infinity*[5]) including solutions that are less intrusive (e.g. armbands as *Q-sensor*[6] and *Empatica*[7]). Furthermore, the game industry has shown interest on embedding physiological sensors in popular console gamepads[8][9] which could lead to the adoption of physiological monitoring during regular video-game play, and consequently facilitate the introduction of psycho-physiological models for procedural adaptation into commercial games. Models of affect that rely solely in physiological signals appear in Chapter 5 and are studied in more detail in Chapter 6.

Context and Interaction

It has been argued that for a complete understanding of experiences of emotion the psychological component must be addressed in addition to the neurobiological instantiation of affective states (Barrett et al., 2007). Part of this psychological component can be introduced into predictors of affect as information about the context and the interactions (actions, system responses and goals) during the affective experience; note that the context of our experiences affects the way that we feel, express and perceive affective states. For instance, a different social context may have a great impact on how we experience a game both psychologically and physiologically (Mandryk et al., 2006). Furthermore, the social context directly affects *display rules* which lead to different expressions of the same emotions (Zeman and Garber, 1996) as our audience changes. Finally, the affective states that we can perceive in e.g. facial expressions, can differ depending on the scene where those are staged (Aviezer et al., 2008) and the facial expression of the AI (or human) controlled agent we interact with (Gratch and Marsella, 2001). Therefore, in addition to the theoretical value of adding context information to an affect predictor, it also provides higher affect recognition rates as it directly influences our affective experiences.

The context modality is available in one of the two affect-related datasets used (Maze-Ball); this game dataset contains, among other information, the events that occur in the game session (e.g. the player is hit by an enemy), which will have a direct contribution to changes on the affective experience of the player. Models of affect that combine physiological signals and context are discussed in Chapter 7.

3.2 Feature Extraction

Feature extraction is applied to transform the raw input signals into a set of features that can feed a computational predictor. The goal of this process is to extract the distinctive characteristics of the signals creating an input space with a reduced number of meaningful dimensions that facilitate the prediction of the target function. In this thesis three approaches that capture dissimilar aspects of the signals are applied: an ad-hoc method containing features frequently used in AC studies (described in Chapter 4 for each different

[4]Accessed November 14, 2012 `http://www.wilddivine.com/`

[5]Accessed November 14, 2012 `http://www.thoughttechnology.com/`

[6]Accessed November 14, 2012 `http://www.affectiva.com/q-sensor/`

[7]Accessed November 14, 2012 `http://www.empatica.com/`

[8]Accessed November 14, 2012 `http://www.siliconera.com/2011/11/01/sony-patent-reveals-biometric-ps3-controller-and-handheld/`

[9]Accessed February 01, 2013 http://www.siliconera.com/2010/10/07/nintendo-patent-shows-wii-vitality-sensor-game-example/

dataset) and two unsupervised learning methods to create features from the data (Section 3.2.2 and Section 3.2.1).

3.2.1 Deep Learning

In this section we propose and describe a method for creating features based on *deep architectures* for sequential signals. The basic idea is to build a *deep* model that receives as inputs the raw signals and outputs a set of features that feeds the computational model of affect (see Figure 3.2). The automation of feature extraction via *deep learning* could potentially yield a more complete set of physiological features than ad-hoc feature extraction methods which, in turn, will deliver models of affect of higher accuracy. Deep learning has been used to improve the accuracy of large classifiers with many inputs (e.g. all the pixels in an image) (Bengio, 2009); in their process, DL methods tune the (initially random) parameters of a multi-layered (deep) model to facilitate supervised learning. While initially the layers of the model produce a random projection of the input data, after the tuning process they extract meaningful information. Hence, DL is an ideal solution for automatically creating features.

We use *convolutional auto-encoders* (CAEs) that utilize a convolutional neural network to create a hierarchy of features that reduces long and complex signals (see Section 3.2.1). Over other models, CNNs present the advantage of creating simple features at each layer, which allow us not only to reduce the signals, but also facilitates the analysis of the features, and therefore the affect models. As for training algorithm, CAE adjusts the parameters of the CNN using auto-encoders (Vincent et al., 2008) that train filters or feature extractors that capture a distributed representation of the leading factors of variation of their input signals (see Section 3.2.1). This is the same idea as the process followed by PCA but with the advantage of bypassing the linearity assumption (Wold et al., 1987). Alternative methods such as restricted Boltzmann machines present a more complex theoretical background but have not produced better results in practice. Thus, we prefer to utilize the simplest method.

Convolutional neural networks

Convolutional or time-delay neural networks (LeCun and Bengio, 1995) are hierarchical models designed to process input spaces where a spatial or temporal relation exists (e.g. images, speech or physiological signals). The hierarchy alternates a number of convolutional and pooling layers in order to process large input spaces (see Figure 3.2).

Convolutional layers contain a set of neurons that detect different patterns on a *patch* of the input (e.g. a time window in a time-series or part of an image). The inputs of each neuron (namely *receptive field*) determine the size of the patch. Each neuron contains a number of trainable weights equal to the number of its inputs and an additional *bias* parameter (also trainable); the output of each neuron is calculated by applying an activation function (e.g. logistic sigmoid) to the weighted sum of the inputs plus the bias (see Figure 3.3). Each neuron scans sequentially the input signal producing an output for each consecutive patch. The consecutive outputs generated assemble a new signal referred to as *feature map* (see Figure 3.2). The output of the convolutional layer is the set of feature maps resulting from convolving each of the neurons in the layer across the input. Note that a convolution layer does not typically reduce the size of the input as the convolution of each neuron produces a signal of the same length as the input signal minus the size of the patch (i.e. the size of the receptive field of the neuron), plus 1 (see Figure 3.2).

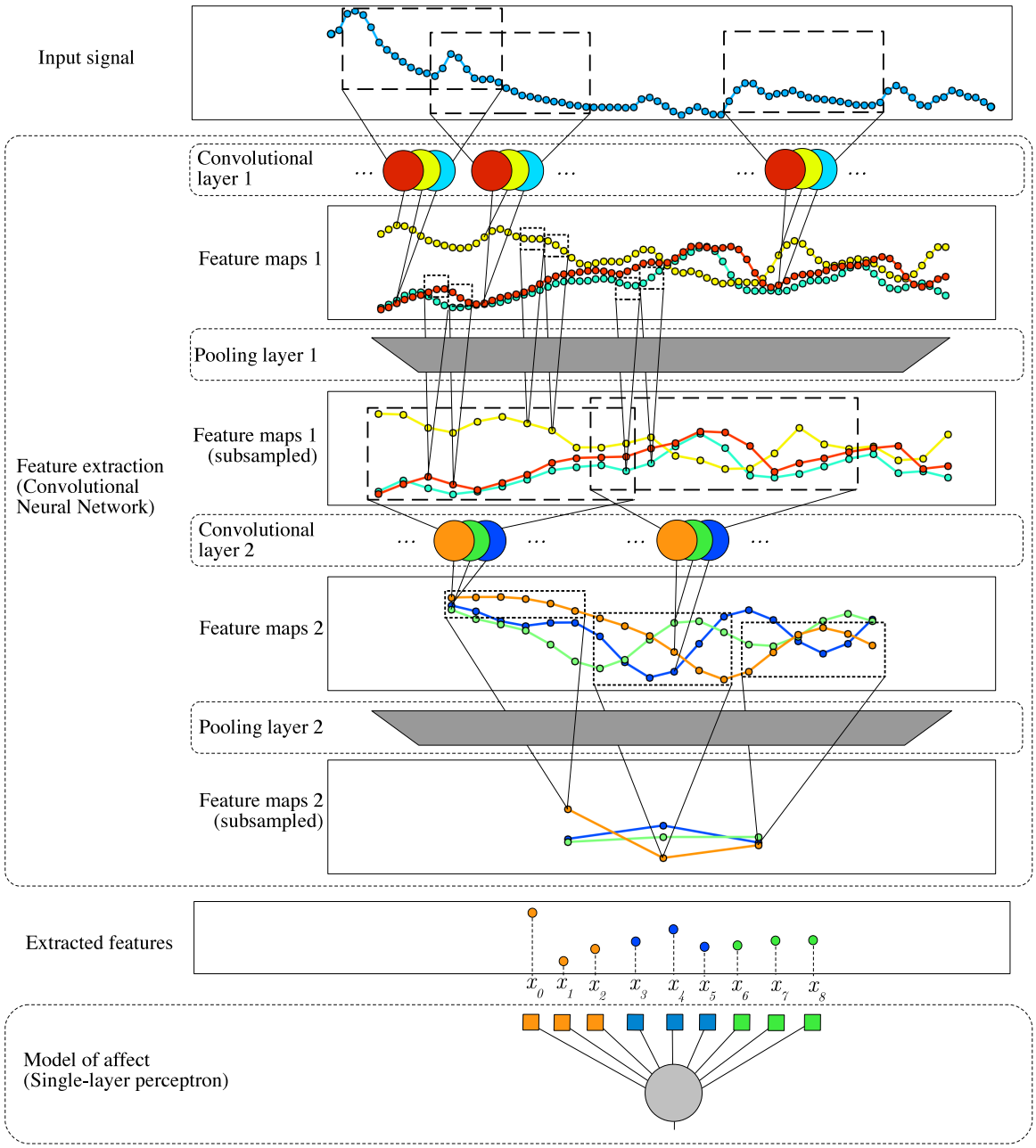


Figure 3.2: Example of the structure of a deep architecture. The architecture contains: (a) a convolutional neural network with two convolutional and two pooling layers, and (b) a single-layer perceptron predictor. In the illustrated example the first convolutional layer (3 neurons and path length of 20 samples) processes a skin conductance signal which is propagated forward through an average-pooling layer (window length of 3 samples). A second convolutional layer (3 neurons and patch length of 11 samples) processes the subsampled feature maps and the resulting feature maps feed the second average-pooling layer (window length of 6 samples). The final subsampled feature maps form the output of the CNN which provides a number of extracted (learned) features which feed the input of the SLP predictor.

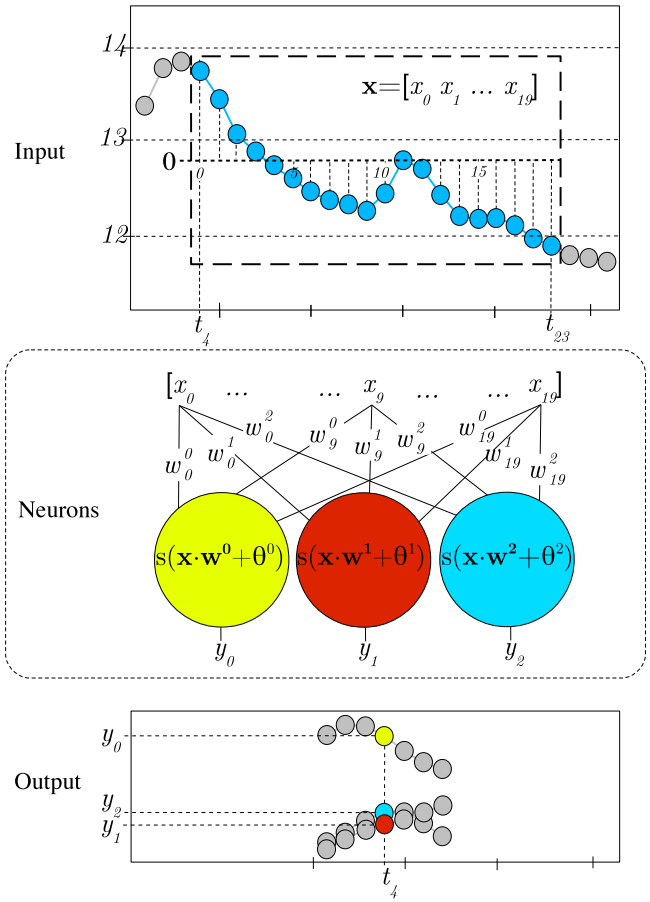


Figure 3.3: Convolutional layer. The neurons in a convolutional layer take as input a patch on the input signal $\mathbf{x}$. Each of the neurons calculates a weighted sum of the inputs ($\mathbf{x} \cdot \mathbf{w}$), adds a bias parameter θ and applies an activation function $s(x)$. The output of each neuron contributes to a different feature map. In order to find patterns that are insensitive to the baseline level of the input signal, $\mathbf{x}$ is normalized with mean equal to 0. In this example, the convolutional layer contains 3 neurons with 20 inputs each.

The reduction is produced in the *pooling* layers which reduce the resolution of each feature map through a pooling function. A window of given length slides over each feature map reducing consecutive and non-overlapping segments of the signal into one value (see Figure 3.4). The *maximum* or *average* values are the two most commonly used pooling functions providing *max-pooling* and *average-pooling* layers, respectively. The output of a pooling layer presents the same number of feature maps as its input, but the resolution of each of them has been reduced by a factor equal to the window length (see Figure 3.2). As consecutive values of a feature map are similar — due to be produced by the same neuron evaluated at contiguous locations of the input signal — small window lengths will produce small losses of information.

The pooling layers perform a very simple processing task, and thus the extracted features are mainly defined by the convolutional layers, and more concretely, their neurons. By analyzing the weights of each neuron, one can derive the characteristics of the input that every feature is capturing. When the input to the neuron is a 1-dimensional signal (e.g. heart rate), the weights of the neuron can be plotted in temporal order to reveal the input

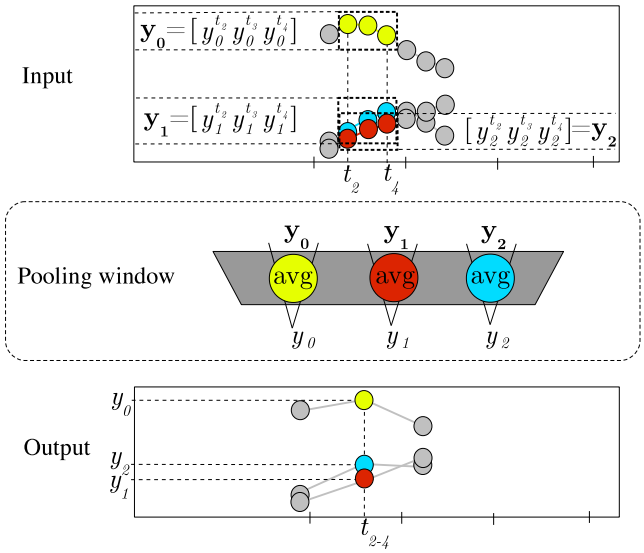


Figure 3.4: Pooling layer. The input feature maps are subsampled independently using a pooling function over non-overlapping windows, resulting in the same number of feature maps with a lower temporal resolution. In this example, an average-pooling layer with a window length of 3 subsamples 3 feature maps.

patterns that yield higher output values.

Auto-encoders

An auto-encoder (Hinton and Zemel, 1994; Hinton and Salakhutdinov, 2006; Bengio et al., 2007) is a model that transforms an input space into a new distributed representation by applying a deterministic parametrized function (e.g. a single layer of logistic neurons) called the encoder (see Figure 3.5). The AE also learns how to map back the output of the encoder into the input space, with a parametrized decoder, so as to have small reconstruction error on the training examples, i.e. one minimizes the discrepancy between the outputs of the decoder and the original inputs. However, constraints on the architecture or the form of the training criterion prevent the auto-encoder from simply learning the identity function everywhere. Instead, it will learn to have small reconstruction error on the training examples (and where it generalizes) and high reconstruction error elsewhere (Bengio et al., 2012). We train the convolutional layers of a CNN by casting their neurons as an encoder and training their weights to reconstruct the input signals or input feature maps. As decoder we used the same weights, approach known as *tied weights*.

The encoder is trained using backpropagation, a gradient descent method that iteratively adjust the weights to minimize the reconstruction error. We used a denoising auto-encoder (Vincent et al., 2008) in which the reconstruction error is defined as the sum of squared differences between the inputs and the reconstructed corrupted inputs. The latter corresponds to the output of the decoder, which takes as inputs the outputs of the encoder, which in turn takes the input signal with added noise (corrupted inputs). This added noise contributes to the auto-encoder's ability of reconstructing the input signal despite certain level of noise. In this thesis, each convolutional layer of the CNN is trained one by one, from bottom to top (Masci et al., 2011). The neurons of each convolutional layer are trained patch-wise, i.e. by considering the input at each position (one patch) in the sequence as one example.

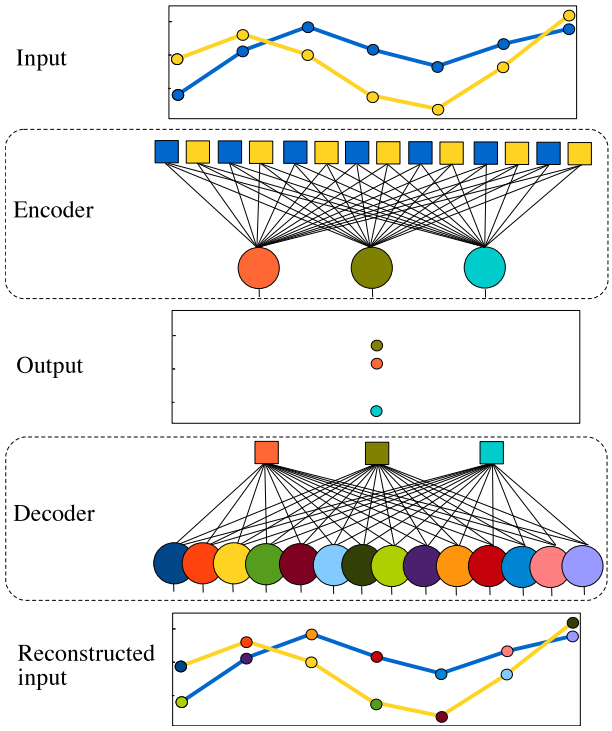


Figure 3.5: Structure of an auto-encoder. The encoder generates the learned representation (extracted features) from the input signals. During training the output representation is fed to a decoder that attempts to reconstruct the input.

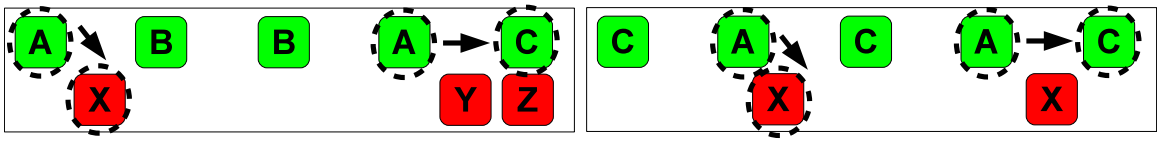
This allows faster training than training convolutionally, but may yield translated versions of the same filter.

3.2.2 Frequent Sequence Mining

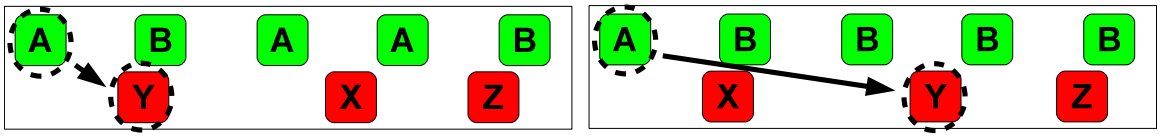
Frequent sequence mining methods are typically used to characterize sequential datasets by finding the regular (frequent) subsequences. When we apply feature extraction, we are trying to extract the defining characteristics of a dataset; thus, we can use the characteristics represented by the frequent sequences as features to reduce a dataset. We can define each data sample by the frequent sequences that it contains, thus reducing the dataset by removing information that is rare (or infrequent). While infrequent information can be relevant for affect prediction, patterns that are seen often will contribute to predictors with better generalization, as patterns present in the dataset are likely to be seen in new data samples. Moreover, sequence mining can be used to mine frequent sequences that span across several modalities, providing an unsupervised method to extract multi-modal features. In this section we formalize the problem of mining frequent sequences, describe the specific algorithm applied in this thesis (generalized sequence patterns) and outline the mechanisms used for defining features from frequent sequences.

Problem formulation

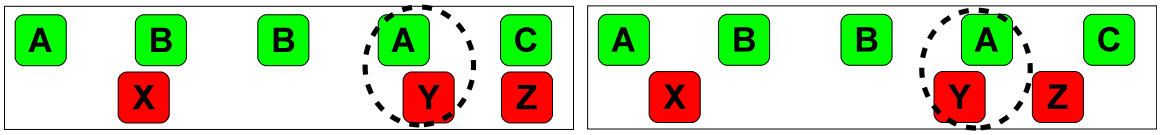
Let us define a *data-sequence* as sequence of *events*, which corresponds to an input sample in a dataset. Then, we define a *frequent sequence* as a subsequence of events that occurs



(a) Consecutive events: sequences $\langle AX \rangle$ and $\langle AC \rangle$ are both frequent despite events occurring in different modalities and other events occurring in between.



(b) Sliding window: the sequence $\langle (AY) \rangle$ (events A and Y considered simultaneous) is frequent if the distance between A and Y is below the time window W_{max} in both samples.



(c) Time constraint: the sequence $\langle AY \rangle$ is frequent if the time constraint G_{max} is equal or longer than the distance between A and Y in the sample on the right.

Figure 3.6: Frequent sequence constraints. The three main requirements or constraints for considering a sequence of events frequent are showcased with two data samples with two modalities (red and green) with events $\{A, B, C\}$ and $\{X, Y, Z\}$, respectively.

regularly in the dataset (i.e. in a large number of data-sequences). An event is associated with a time stamp (discrete moment on time when the event occurs) and an identification (type of event). An event could be for example an increase on heart rate, a key pressed or an action unit activation in a facial expression task (Pantic et al., 2005). Note that, signals composed by sequences of continuous values have to be transformed into sequences of discrete events. For example, a SC signal can be converted to a sequence of sudden increments and sudden decrements (detected as large changes in small time intervals of the signal). Certainly, this transformation removes some amount of information about the signal — in the SC example, the tonic component would be completely lost— but on the other hand may lead to the discovery of interesting patterns across signals that are not accessible otherwise.

More formally, a sequential pattern is defined as an ordered list of *elements* — denoted as $\langle e_0 e_1 \dots e_n \rangle$; e_i is the i^{th} element of the sequence — each containing a non-empty (unordered) set of m simultaneous *events* — denoted as $(x_0, x_1, \dots, x_m)$; x_i is an event. For example, an element could be two keys pressed simultaneously, several action units executed at the same time or an increase in heart rate (an element with only one event). A frequent sequence can be defined as a sequential pattern that is *supported* by, at least, a minimum amount of data-sequences as determined by the *minimum support* (S_{min}) value. A data-sequence supports a sequential pattern if and only if it contains all the events present in the pattern in the same order. Note that this definition does not restrict that events in consecutive positions within the pattern must be strictly consecutive in the data-sequence (see Figure 3.6a). For example, the data-sequence $\langle x_0 x_1 x_2 x_3 x_4 x_5 \rangle$ supports the pattern $\langle e_0 e_5 \rangle$ with $e_0 = (x_0)$ and $e_5 = (x_5)$ if further constrains are not specified. The amount of data-sequences that support a sequential pattern is referred as the *support count*.

Algorithm 1 Generalized Sequential Patterns.

Input: a dataset of data-sequences, S_{min}, G_{max} and W_{max}
Output: the set of sequence patterns that are supported by more than S_{min} data-sequences.
01: **procedure** GSP(data, S_{min}, G_{max}, W_{max})
02: Count the number of data-sequences in which each different event is contained (support count).
03: Insert the events with a support count greater than S_{min} into the set of frequent 1-sequences (L_1).
04: k = 1
05: **while** L_k is not empty
06: Generate the set of candidate $(k+1)$-sequences (C_{k+1}). See Algorithm 2 for more details.
07: Determine the support count of the sequences contained in C_{k+1}.
08: Create the set of frequent $(k+1)$-sequences (L_{k+1}) with the sequences in C_{k+1} that present a support count greater than S_{min}.
09: k = k +1
10: **return** $L_1 \cup L_2 \cup \cdots \cup L_{k-1}$

In order to model the fusion of sequences from different modalities, the basic definition of frequent sequence is extended with two of the generalizations proposed in (Srikant and Agrawal, 1996) which are as follows:

- Sliding window: given an element e_i containing two or more simultaneous events $(x_0, x_1...x_m)$, a data-sequence contains the element e_i if and only if all its events occur in the data-sequence within a given time window W_{max} (see Figure 3.6b). In other words, two or more events in a data-sequence can be considered to occur simultaneously (i.e. belonging to the same element) if and only if they occur within a time interval shorter than W_{max}.

- Time constraint: given two consecutive elements in a pattern, $e_i e_{i+1}$, a data-sequence may support the pattern only if both elements occur in the specified order and the time difference between their occurrences is lower than a specified time threshold, *maximum gap* G_{max} (see Figure 3.6c).

Given this formulation, we need to specify only three parameters, the minimum support, the time window and the maximum gap, which constrain the frequent sequences that will be extracted. Then, an automated search would mine the dataset to find every matching subsequence.

GSP algorithm

The generalized sequential patterns algorithm (Srikant and Agrawal, 1996) is used for mining the frequent sequences in this thesis. GSP is a *candidate generation and test* algorithm which supports the constrains mentioned in the previous section. It first finds the frequent sequences with one single event, namely 1-sequences. That set of sequences is self-joined to generate all 2-sequence candidates for which their support count is calculated. Those sequences that are frequent (i.e. their support count is greater than a threshold value S_{min}) are self-joined to generate the set of 3-sequence candidates. The algorithm is iterates, increasing the length of the sequences in each algorithmic step, until the next set of candidates

Algorithm 2 Candidate generation in GSP.

Input: a set of $(k-1)$-sequences L_{k-1}.
Output: the set of candidate k-sequences C_k.
01: **procedure** generateCandidates(L_{k-1})
02: **for each** pair of sequences $s_x, s_y \in L_{k-1}$ with $s_x = \langle e_1^x e_2^x \dots e_n^x \rangle$ and $s_y = \langle e_1^y e_2^y \dots e_n^y \rangle$
03: **if** the two sequences obtained by dropping the first event of s_x and the last event of s_y are identical
04: **if** e_n^y has only one event $e_n^y = (y_1)$
05: Generate the candidate sequence s_{xy} by inserting y_1 as last event of e_n^x: $s_{xy} = \langle e_1^x e_2^x \dots e_{n-1}^x (e_n^x, y_1) \rangle$
06: **else**
07: Generate the candidate sequence s_{xy} by replacing e_n^x with e_n^y: $s_{xy} = \langle e_1^x e_2^x \dots e_{n-1}^x e_n^y \rangle$.
08: **if** all contiguous subsequences of s_{xy} are contained in L_{k-1}
09: Insert s_{xy} into C_k.
10: **return** C_k

is empty. The basic principle of the algorithm is that if a sequential pattern is frequent, then its *contiguous* subsequences are also frequent. Given two sequences s_x and s_y, s_y is a contiguous subsequence of s_x if either: 1) s_y is obtained by dropping the first or last event of s_x; or 2) s_y is obtained by dropping an event from an element of s_x with two or more events; or 3) there exists a sequence s_z such that s_z is a contiguous subsequence of s_x and s_y is a contiguous subsequence of s_z.

By self-joining a set of frequent sequences of length k, the algorithm obtains only the $(k+1)$-sequences whose contiguous subsequences are frequent, thereby, reducing the number of sequential patterns for which support counts have to be determined. The reader is referred to Algorithm 1 and Algorithm 2 for a more detailed presentation of the basic steps of the GSP algorithm.

Feature creation

After the frequent sequences have been found in the full dataset, each data-sequence is transformed into a vector of features. For each frequent sequence we create one feature; and for each data-sequence we calculate its value using one of the following strategies:

- Count: the number of occurrences of the frequent sequence within the data-sequence is used as feature.
- Boolean: the feature is equal to one if the frequent sequence occurs at least once within the data-sequence, and 0 if it never occurs.

3.3 Automatic Feature Selection

As feature extraction derives a number of characteristics from the input signal regardless of the target affective state, automatic feature selection is an essential process towards distinguishing which of those features can assist the creation of the model of affect. FS consists of a search scheme to test alternative combinations of input features and a heuristic to determine their relevance. Opposed to other dimensionality reduction methods such as

principal component analysis (Wold et al., 1987) and *Fisher's projection* (Krzanowski, 1977) that project the feature space into a space of lower dimensionality, FS eliminates dimensions (features) from the original space maintaining the physical meaning of the inputs to the model. We consider that this is a key feature for affect modeling, as it is necessary to analyze the mappings captured by the models learned; analysis that can lead to a better understanding of human affect.

Two FS procedures are applied in this thesis: a local search and a global search, namely *sequential forward feature selection* (SFS) and *genetic search feature selection* (GFS). In both algorithms, the relevance of each set of selected features is measured through the performance of a model built on those features. Note that neither method presented is guaranteed to find the optimal feature set since all possible combinations are not evaluated (SFS is a hill-climber and GFS is based on genetic search).

3.3.1 Sequential Forward Feature Selection

Sequential forward feature selection is a bottom-up search procedure where one feature is added at a time to the current feature set. The feature to be added is selected from the subset of the remaining features such that the new feature set generates the maximum value of the performance function over all candidate features for addition. The search stops when adding a new feature does not yield an increase in performance. We apply this method because while being simple and fast, it has been successfully applied in dissimilar studies (Yannakakis and Hallam, 2007; Pedersen et al., 2010; Yannakakis et al., 2010 among many) for affective preference prediction.

3.3.2 Genetic Feature Selection

GFS implements a generational genetic algorithm (GA) to search for the set of features that yields the most accurate preference predictor for the investigated affective state. According to the GFS mechanism, the whole set of input features is encoded as a bit string chromosome, c:

$$c = (g_1, g_2, ..., g_{N_F}) \tag{3.1}$$

where

$$g_i = \begin{cases} 1, & \text{if feature } i \text{ is included} \\ 0, & \text{if feature } i \text{ is not included} \end{cases} \tag{3.2}$$

and N_F is the total number of features existent in the input dataset.

A population of N_c chromosomes is initialized with all bits set to zero but one selected randomly; i.e. the first generation consists of sets of one randomly selected feature. The reason for initializing chromosomes with only one feature is to obtain minimal feature subsets which, nevertheless, yield high performing predictors of reported affect — serving as the input of the model. Then, at each generation:

1. All chromosomes of the population are evaluated. For that purpose a preference model is built using as inputs the feature set presented by the chromosome and its performance serves as *fitness function*.

2. An *elitism* selection method chooses the N_p individuals with highest fitness to be the parents of the next generation.

3. Pairs of parents are selected using a rank selection method that ranks the parents by their fitness and then selects two of them with a probability proportional to r; where r is the position in the ranking. A total of $N_c - N_p$ offspring are reproduced via uniform crossover — i.e. every gene has the same probability of being from one or the other parent — with probability p_c. If crossover is not applied, the most-fit parent of the two is cloned to generate an offspring.

4. For each offspring, mutation occurs at each gene with probability p_m. The mutation scheme used flips the value of the selected gene which, in turn, suggests that the corresponding feature is either added (1) or removed (0) from the feature set. Finally, all offspring are inserted to the population.

The algorithm terminates after G_{max} generations and the set of features corresponding to the highest performing preference predictor found across all generations is chosen. It is noteworthy that parent chromosomes are cloned to the new generation but their performance is re-evaluated, i.e. a new model is built on that feature set. Therefore, when the training algorithm used to build the model is non-deterministic (e.g. neuroevolution), the fitness function of some individuals may fluctuate significantly from one generation to the next. We apply this method because it enables a more exhaustive exploration of the space of input features than SFS while maintaining an acceptable computational cost.

3.4 Preference Learning

Preference learning (Fürnkranz and Hüllermeier, 2010a) is a subfield of machine learning that deals with the problem of learning orders. Object ranking is a particular category of problems within PL in which a set of rankings or pairwise comparisons (partial orders) are specified over a set of data samples (objects). The goal is to learn a function that given a set of objects can predict the order among them. Considering affective experiences as different objects, modeling affect from ordinal reports (e.g. ratings and pairwise preferences) can be cast as an object ranking task.

Before delving into the details of the dissimilar methods used in this thesis, the problem of object ranking is formalized here. Let $\mathbf{x_i}$ denote an object from the complete set of possible objects χ^* and be defined as a feature vector $\mathbf{x_i} = [x_{i0}, x_{i1}, ..., x_{in}]$. An order O_k induces a preference relation among a set of objects, $\chi^k \subseteq \chi^*$ such that

$$\forall \mathbf{x_i}, \mathbf{x_j} \in \chi_k \begin{cases} (\mathbf{x_i} \succ \mathbf{x_j}) \in O_k \text{ , or} \\ (\mathbf{x_i} \prec \mathbf{x_j}) \in O_k \text{ , or} \\ (\mathbf{x_i} \equiv \mathbf{x_j}) \in O_k \end{cases} \tag{3.3}$$

where $\mathbf{x_i} \succ \mathbf{x_j}$ denotes that $\mathbf{x_i}$ has a higher ranking than (or is preferred over) $\mathbf{x_j}$, and $\mathbf{x_i} \equiv \mathbf{x_j}$ denotes that $\mathbf{x_i}$ and $\mathbf{x_j}$ present the same ranking (or are preferred equally). For example, consider 10 games for which final score and average heart rate of the player are recorded. $\mathbf{x_0}$ through $\mathbf{x_9}$ would refer to each of the games with $x_{0,0}$ and $x_{0,1}$ the final score and heart rate during the first game, respectively. O_k may represent frustration self-reports for player k (e.g. game 1 was more frustrating than game 2 which in turn was as frustrating as game 3 which is denoted as $O_k = \{\mathbf{x_0} \succ \mathbf{x_1} \equiv \mathbf{x_2}\}$). Given a set of orders $S = \{O_1, O_2, ..., O_m\}$, the object ranking task consists of learning the underlying function that maps the features of the objects onto an ordered space from which the orders in S are hypothetically sampled.

According to Fürnkranz and Hüllermeier (2010a), there are two main approaches to model that function: as *binary relations* and as *utility functions*.

Methods on the first OR approach learn classifiers that given two objects predict which one is preferred. This is, the models receive as inputs two affective experiences and predict whether int the first or the second the target affective state is felt stronger (e.g. it is more frustrating). These models are therefore sensitive to the order of the experiences but on the other hand, two experiences are required to make a prediction which might not be suitable for some real-time applications. Cohen's method (Cohen et al., 1999) is arguably the best representative of this category and it will be described in detail in Section 3.4.3. The methods within the second OR approach build models that receive as input a single experience and output a continuous value that estimates the preference (e.g. level of frustration). Given a pair of objects, the model predicts that the object that yields higher output is preferred. Such models are more apt for real-time applications as they only need information about the current experience. On the other hand, a major restriction of this approach is that it cannot model intransitive preferences/cycles (e.g. $\{(\mathbf{x_0} \succ \mathbf{x_1}), (\mathbf{x_1} \succ \mathbf{x_2}), (\mathbf{x_2} \succ \mathbf{x_0})\}$) which is useful, for instance, to model order effects.

The problem described in the second OR approach shares many characteristics with standard classification and regression tasks in which the goal is to learn an unknown utility function that maps input objects into given target values (nominal or real values). A widely used method to learn the utility function consists of choosing a particular parametric model $U^{\mathbf{w}}(\mathbf{x})$ (e.g. a perceptron) and finding the parameter values $\mathbf{w}$ for which a given error function E (e.g. sum of squared errors) is minimized. That error function generally depends on the estimated ($U^{\mathbf{w}}(\mathbf{x_i})$) and target ($y_i$) values for the samples on the training dataset ($(\mathbf{x_i}, y_i) \in D$). By definition, in object ranking there are no target values y_i existent for each object $\mathbf{x_i}$ which prevents from using standard error functions developed for classification and metric regression unless certain assumptions are made. A simple assumption would be — given that ordinal labels (ranks) are assigned to each object — to ignore the ordinal nature of the data and model ranks as independent classes. In such a scenario, multi-class classification algorithms can be applied; however, this approach is expected to yield less accurate models than other methods that take the order into account (Crammer and Singer, 2001). A different approach consists of assuming consecutive ranks as equidistant (e.g. Kamishima et al., 2005; Kramer et al., 2001) which, in turn, transforms the OR problem into a metric regression task. However, as discussed in Section 3.1.1, this assumption does not generally hold for data involving human subjective reports.

For the benefit of OR, several error functions have been developed to measure the degree of agreement between an OR model and a set of target orders. These functions can be used to extend standard ML algorithms without additional assumptions. This includes methods on support vector machines (Herbrich et al., 1999; Joachims, 2002), artificial neural networks (Burges et al., 2005; Yannakakis et al., 2009; Shivaswamy and Joachims, 2011; Delalleau et al., 2011), Gaussian processes (Chu and Ghahramani, 2005; Abbasnejad et al., 2011), linear discriminant analysis (Tognetti et al., 2010b,a) and boosting algorithms (Freund et al., 2003).

In this thesis we examine ANNs and SVMs. The main focus of this thesis is on ANNs because they can approximate models of any complexity (not only linear as for example LDA), and to some extent, they can be analyzed by an expert. SVMs are chosen as a comparative method because they have yielded excellent results in different modeling tasks (including affect) and can model a large number of function complexities. Finally, Cohen's method is also included to provide a comparison against a method from the alternative

OR approach. In the reminder of this section these three algorithms for object ranking are described in detail.

3.4.1 Artificial Neural Networks

Artificial neural networks (Bishop, 1995) are biologically-inspired computational models used as function approximators for pattern recognition in many domains such as computer vision. An ANN is defined as a network of processing units called *neurons*. Each neuron receives a number of real-valued inputs and calculates its output as follows:

$$o_j = f(\sum_{i=0}^{n-1} x_{ij} w_{ij} + \theta_j) \tag{3.4}$$

where $\mathbf{w_j} = [w_{j0}, w_{j1}, ..., w_{jn-1}]$ are the *connection weights*, $\mathbf{x_j} = [x_{j0}, x_{j1}, ..., x_{jn-1}]$ are the inputs, o_j is the output, θ_j is the *threshold* and $f(x)$ is the *activation function* of neuron j. The weights and threshold are real-valued parameters that are adjusted to yield different functions. The activation function is typically a logistic sigmoid or hyperbolic tangent in prediction tasks because they are monotonic, their output ranges are bounded and present a computationally cheap derivative (characteristics that facilitate gradient-descent training).

A single-layer perceptron is the simplest ANN topology in which all the inputs of the network are connected to the inputs of the neurons, and the output of each neuron forms one of the outputs of the network. While the expressivity of an SLP is limited, arbitrarily complex functions can be represented by more complex topologies where the output of some neurons is connected to the input of others. Theoretical results elaborated by Kolmogorov (1963) support that a neural network with a layer of neurons with logistic sigmoid activation functions connected to an output neuron with a linear activation function can approximate any continuous function for a given precision (i.e. universal approximation). This result suggests that, in theory, neural networks define the ideal tool for approximating continuous functions; however, in practise finding the appropriate topology and precise connection weights within a reasonable amount of time is not always guaranteed.

A multi-layer perceptron is a neural network topology where neurons are arranged in stacked layers; the outputs of every neuron in one layer are connected only to every neuron in the next layer (see Figure 3.7). The inputs of the network are connected to every neuron in the first layer and the outputs of the neurons in the last layer (output layer) form the outputs of the network. This has been the most popular type of ANN for decades in pattern recognition in part due to its theoretical representation power (MLPs with one logistic hidden layer can approximate any continuous function) and relatively simple and efficient *training* strategies. In ANN terminology, training refers to the automatic process by which the appropriate weights of a network, activation functions and network topology are selected to approximate an unknown function. This process requires two main elements: a function to determine the *"goodness"* of any configuration (*error function*) and a search algorithm to efficiently traverse the space of possible configurations (*training algorithm*). These elements are described in the following sections.

Training Algorithms

The space of possible configurations of a neural network, even if the topology of the network and the activation functions are fixed, is typically too big for exhaustive exploration. Back-propagation and neuroevolution are popular ANN training algorithms based on gradient-

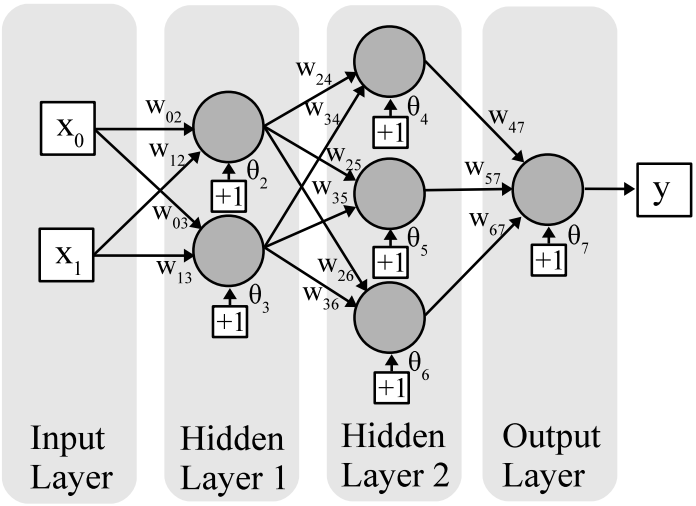


Figure 3.7: Example of a multi-layer perceptron with two inputs (x_0 and x_1) and one output (y). Two hidden layers with 2 and 3 neurons, respectively, transform the input and feed the single neuron in the output layer. Connection weights and thresholds are displayed over the corresponding connection arrows.

descent (local search) and evolutionary algorithms (global search), respectively. These algorithms have been largely applied to neural network training on regression and classification tasks and they can be applied unchanged to preference learning.

Backpropagation (Rumelhart, 1995) on its basic form optimizes an error function iteratively across a number of epochs by adjusting the value of each weight and threshold proportionally to the derivative (gradient) of the error with respect to the weight. This method requires that the topology and activation functions are fixed before hand. In this thesis, the following weight-update rule is used:

$$w_i^{t+1} = w_i^t - \lambda_1 \frac{1}{|S|} \sum_{(\mathbf{x_P},\mathbf{x_N})\in S} \frac{\partial E(U^{\mathbf{w^t}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N})}{\partial w_i^t} + \lambda_2 \frac{\partial (\mathbf{w^t})^2}{\partial w_i^t} \tag{3.5}$$

where E is the error function which depends on a pair of objects $(\mathbf{x_P}, \mathbf{x_N})$ and the current configuration of the network $U^{\mathbf{w^t}}(\mathbf{x})$ (see Section 3.4.1), S is the set of pairs in the training dataset, $|S|$ is the number of pairs in S, w_i^t is the value of weight i at epoch t, λ_1 is the learning rate and λ_2 is a weighting parameter for the *regularizer* term. The regularizer is used to maintain the weights of the network low and by doing so, avoiding (reducing) *overfitting* (Bishop, 1995). Overfitting is an issue inherent to ML algorithms that consists of a model *memorizing* training examples rather than learning the underlying function. If overfitted an ANN model makes good predictions in the training dataset, but poor approximations on similar (unseen) data. As the weight update relies on the calculation of the derivative of the error function with respect to the weights of the network, the error function needs to be differentiable and depend on the values of the weights at a sufficient number of points because otherwise, the weight update cannot be computed or is equal to zero.

Neuroevolution (Moriarty and Miikkulainen, 1997) is the application of evolutionary algorithms to ANN training. A genetic algorithm (Goldberg, 1989) is a global search strat-

egy that features methods inspired by natural evolution. In brief, a GA on its general form maintains a population of individuals or *phenotypes* each one encoded by a *genotype* or set of *genes* (*chromosome*). In NE, the phenotypes correspond to different neural networks while the genotype can be for example, the set of connection weights. Iteratively across a number of generations, the GA modifies the population by replacing existing individuals by new phenotypes created throughout a number of genetic operators, such as *cross-over* and *mutation*, applied to selected genotypes. Cross-over creates a new genotype by recombining genes from already existing individuals. In turn, mutation creates a new genotype by modifying existing individuals. The selection of individuals to be replaced, recombined or modified is influenced by their *fitness* (F) to induce an improvement of the fittest individuals across generations.

For the simplest form of NE, every individual represents a neural network with a fixed topology whose genotype consists of the set of weights; the fitness function is the inverse of an error function, in this way the GA evolves an initial population of neural networks with random weights towards different weight configurations that yield lower errors. More specifically, in this thesis NE is applied with an elitism replacement strategy in which a fraction of fittest individuals are retained into the population of the next generation and all the others are replaced keeping the size of the population constant across generations. The new individuals are generated by applying *uniform crossover* (Harik et al., 1999) with probability p_c to pairs of individuals selected from the whole population using a *rank selection strategy*. Rank selection picks an individual i with probability proportional to $\frac{1}{r_i}$ where r_i is the rank of that individual in the population according to its fitness. *Gaussian mutation* is applied to each gene of the new individuals with probability p_m which adds a random number sampled from a Gaussian distribution ($\mu = 0, \sigma = 0.1$) to the weight (gene) selected to be mutated. The fitness function for each individual is calculated as follows:

$$F(i) = \frac{1}{|S|} \sum_{(\mathbf{x_P}, \mathbf{x_N}) \in S} -E(U^{\mathbf{w}^i}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N}) - \frac{\lambda}{|\mathbf{w}^i|} \sum_{w_j^i \in \mathbf{w^i}} (w_j^i)^2 \tag{3.6}$$

where E is the error function which depends on a pair of objects $(\mathbf{x_P}, \mathbf{x_N})$ and the configuration of the network $U^{\mathbf{w^i}}(\mathbf{x})$ for individual i (see Section 3.4.1), S is the set of pairs in the training dataset, $|S|$ is the number of pairs in S, $\mathbf{w^i}$ is its genotype (vector of weights), $|\mathbf{w^i}|$ its length and λ is a positive weighting parameter for the regularizer term. Similarly to BP, a regularizer term is included in the calculation of the fitness evaluation to penalize the fitness of ANNs with high weight values that are more prone to overfit the training data. The value in this algorithm is divided by the number of weights to reduce the growth of the term as a consequence of increasing the size of the topology.

The simplest form of NE requires that the network topology and activation functions are fixed *a priori* and only the set of weights are evolved. However, more advanced methods for evolving the topology of the network (Stanley and Miikkulainen, 2002) could be used but are not explored in this thesis.

Error functions for object ranking

An error function is required to automatically evaluate the quality of any given parameter configuration that defines a computational model and, thus, enable automatic parameter tuning or training. The functions described in this section define a measure of agreement between two orders, and are used to evaluate the agreement between the order estimated

by a model and the order given by a dataset. All functions examined here are calculated for pairwise preferences (i.e. orders of length two) with the exception of the Spearman function that is calculated for orders of any length (see Equation 3.4.1). Nevertheless, this is not a limitation as any given order $O_k = \{\mathbf{x_0} \succeq \mathbf{x_1} \succeq ... \succeq \mathbf{x_{n-1}}\}$ can be described without loss of information as $\binom{n}{2}$ pairwise comparisons. Additionally, the functions in this thesis are used to model only *clear preferences*; that is, given two objects $\{\mathbf{x_i}, \mathbf{x_j}\}$ within an order O_k, either $(\mathbf{x_i} \succ \mathbf{x_j}) \in O_k$ or $(\mathbf{x_i} \prec \mathbf{x_j}) \in O_k$ is expected, but never $(\mathbf{x_i} \equiv \mathbf{x_j}) \in O_k$. With this in mind and to simplify the notation, for any pair of objects presented to an error function the preferred object is denoted as $\mathbf{x_P}$ and the non-preferred as $\mathbf{x_N}$ so that $\mathbf{x_P} \succ \mathbf{x_N}$ always holds true. Similarly, for brevity the difference between the outputs of a model $U^{\mathbf{w}}(\mathbf{x})$ for the preferred and non-preferred objects of a pair, i.e. $U^{\mathbf{w}}(\mathbf{x_P}) - U^{\mathbf{w}}(\mathbf{x_N})$, is denoted as $U^{\mathbf{w}}_{PN}$. It is worth highlighting that a pair $\{\mathbf{x_P}, \mathbf{x_N}\}$ is only classified correctly by a model $U^{\mathbf{w}}(\mathbf{x})$ when $U^{\mathbf{w}}_{PN} > 0$.

Note that if we use $U^{\mathbf{w}}_{ij}$ to calculate the probability of $\mathbf{x_i} \succ \mathbf{x_j}$, a small value of $U^{\mathbf{w}}_{ij}$ yields low certainty $p_{\mathbf{x_i} \succ \mathbf{x_j}} \simeq 0.5$). However, in the scenario of a strict classification (i.e. $U^{\mathbf{w}}(\mathbf{x_i}) > U^{\mathbf{w}}(\mathbf{x_j})$ implies $\mathbf{x_i} \succ \mathbf{x_j}$), a small value of separation may damage generalization to predictions on unseen data as small disturbances in object feature vectors (e.g. noise) may turn a correctly classified pair into a misclassified one. Particular error functions overcome this problem by rewarding differences that are larger than zero up to some threshold. In order to assess the potential impact of this threshold — referred to as *margin*, m, throughout this dissertation — on the generalizability of object ranking models, we introduce it in the definition of every error function. Figure 3.8 shows all functions with the margin values of 0.0, 0.5 and 1.0 examined in this thesis. In the reminder of this section, we describe all the error functions explored in this thesis and their gradients, as they are key for training neural networks via backpropagation.

Cross-entropy: given a pair of objects $\{\mathbf{x_P}, \mathbf{x_N}\}$, the probability of $\mathbf{x_P} \succ \mathbf{x_N}$ can be defined as the logistic sigmoid function of $U^{\mathbf{w}}_{PN}$; the cross-entropy error function, E_C, is calculated as the cross-entropy cost of this probability, resulting in the following equation:

$$E_C(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N}) = -p_{(\mathbf{x_P} \succ \mathbf{x_N})} \log\big(g(U^{\mathbf{w}}_{PN} - m)\big) - (1 - p_{(\mathbf{x_P} \succ \mathbf{x_N})}) \log\big(1 - g(U^{\mathbf{w}}_{PN} - m)\big) \tag{3.7}$$

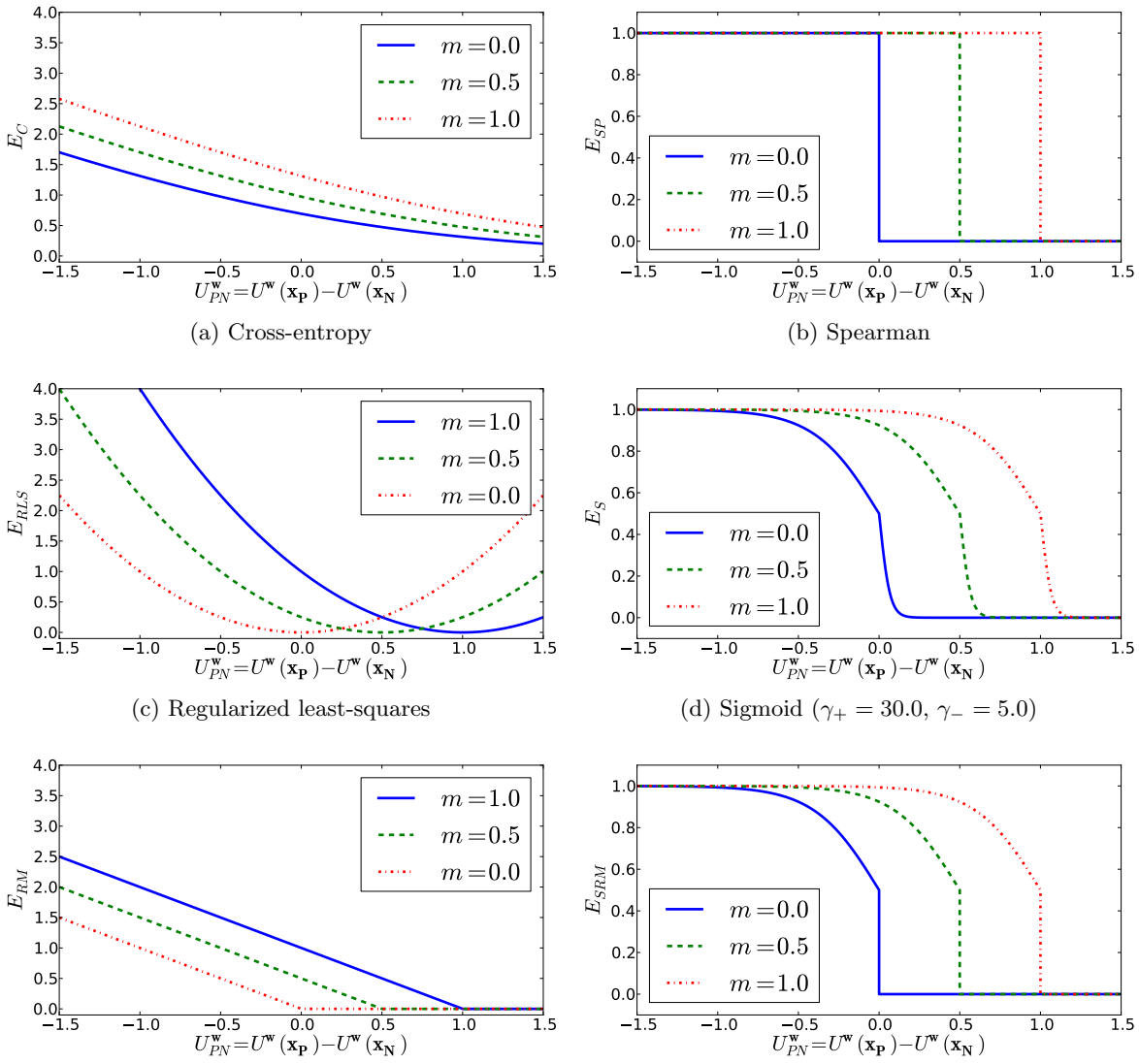
where $p_{(\mathbf{x_P} \succ \mathbf{x_N})}$ is the expected probability of $\mathbf{x_P} \succ \mathbf{x_N}$, m is the margin parameter (typically $m = 0.0$) and $g(x)$ is the logistic sigmoid function.

The resulting function is depicted in Figure 3.8a. In absence of more information it is assumed that $p_{(\mathbf{x_P} \succ \mathbf{x_N})} = 1$, i.e. there is complete certainty that $\mathbf{x_P} \succ \mathbf{x_N}$. For high negative $U^{\mathbf{w}}_{PN}$ values (i.e. clear erroneous predictions) the error function presents high values. Adjustments on the model parameters that correct the prediction produce rapid decrements on the error function which becomes steady after $U^{\mathbf{w}}_{PN}$ is large enough.

The gradient of the function with respect to the model parameters is calculated as

$$\frac{\partial E_C(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N})}{\partial \mathbf{w}} = (g(U^{\mathbf{w}}_{PN} - m) - p_{(\mathbf{x_P} \succ \mathbf{x_N})}) \frac{\partial U^{\mathbf{w}}_{PN}}{\partial \mathbf{w}} \tag{3.8}$$

where $p_{(\mathbf{x_P} \succ \mathbf{x_N})}$ is the expected probability of $\mathbf{x_P} \succ \mathbf{x_N}$, m is the margin parameter (typically $m = 0.0$) and $g(x)$ is the logistic sigmoid function.



(a) Cross-entropy

(b) Spearman

(c) Regularized least-squares

(d) Sigmoid ($\gamma_+ = 30.0$, $\gamma_- = 5.0$)

(e) Rank-margin

(f) Sigmoidal rank-margin ($\gamma = 5.0$)

Figure 3.8: Error functions for object ranking with respect to the difference between the model's output for the preferred and non-preferred objects of a pair ($U^{\mathbf{w}}_{PN}$) and three different margin values, m. The blue line corresponds to the typically used margin for each error function.

Regularized least-squares: given a pair of objects, $\{\mathbf{x_P}, \mathbf{x_N}\}$, the regularized least-squares error function, E_{RLS}, is calculated as follows:

$$E_{RLS}(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N}) = (m - U^{\mathbf{w}}_{PN})^2 \tag{3.9}$$

where m is the margin parameter (typically $m = 1.0$). This function decreases quadratically as the difference between the model's output for the preferred and the non-preferred objects increases. The function becomes zero when this distance is equal to m and increases quadratically after the distance grows past this threshold (see Figure 3.8c). By minimizing this function, the model adjusts its parameters to separate the output of the preferred object from the output of the non-preferred object exactly by the value of the margin . The gradient of this function with respect to the model parameters is calculated as follows

$$\frac{\partial E_{RLS}(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N})}{\partial \mathbf{w}} = 2(U^{\mathbf{w}}_{PN} - m)\frac{\partial U^{\mathbf{w}}_{PN}}{\partial \mathbf{w}} \tag{3.10}$$

where m is the margin parameter (typically $m = 1.0$).

Rank-margin: given a pair of objects, $\{\mathbf{x_P}, \mathbf{x_N}\}$, the rank-margin error function, E_{RM}, is calculated as follows:

$$E_{RM}(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N}) = max\{0, m - U^{\mathbf{w}}_{PN}\} \tag{3.11}$$

where m is a margin parameter (typically $m = 1.0$). This function decreases linearly as the difference between the model's output for preferred and non-preferred objects increases. The function becomes zero after this difference is greater than a margin value (see Figure 3.8e). By minimizing this function, the model adjusts its parameters to separate the model's output for the preferred object from the output for the non-preferred object as much as possible — and below a margin threshold. The gradient of this function with respect to the model parameters is calculated as follows

$$\frac{\partial E_{RM}(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N})}{\partial \mathbf{w}} = \begin{cases} 0 & \text{if } U_{PN} - m > 0 \\ -\frac{\partial U^{\mathbf{w}}_{PN}}{\partial \mathbf{w}} & \text{otherwise} \end{cases} \tag{3.12}$$

where m is a margin parameter (typically $m = 1.0$).

Sigmoid: given a pair of objects $\{\mathbf{x_P}, \mathbf{x_N}\}$, the sigmoid error function, E_S, is calculated as follows:

$$E_S(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N}) = 1.0 - \frac{1}{1 + e^{-\gamma(U^{\mathbf{w}}_{PN} - m)}},\ \gamma = \begin{cases} \gamma_+ & \text{if } U^{\mathbf{w}}_{PN} > m \\ \gamma_- & \text{otherwise} \end{cases} \tag{3.13}$$

where m is the margin parameter (typically $m = 0.0$), and γ_+ and γ_- are two weight parameters that influence the slope of the sigmoid function on the two ends of the decision boundary between correctly and incorrectly classified pair ($U^{\mathbf{w}}_{PN} = 0$). The value of E_S decreases monotonically with respect to difference between the model's output for the preferred and the non-preferred objects. Due to the sigmoidal shape of the function rapid E_S value changes occur around the decision boundary ($U^{\mathbf{w}}_{PN} = 0$) while relatively small changes occur far from it (see Figure 3.8d). The gradient of this function with respect to the model parameters is defined as follows:

$$\frac{\partial E_S(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N})}{\partial \mathbf{w}} = -\gamma\Big(1.0 - g\big(\gamma(U^{\mathbf{w}}_{PN} - m)\big)\Big) g\big(\gamma(U^{\mathbf{w}}_{PN} - m)\big) \frac{\partial U^{\mathbf{w}}_{PN}}{\partial \mathbf{w}},$$
$$\gamma = \begin{cases} \gamma_+ & \text{if } U^{\mathbf{w}}_{PN} > m \\ \gamma_- & \text{otherwise} \end{cases} \tag{3.14}$$

where m is the margin parameter (typically $m = 0.0$), and γ_+ and γ_- are two weight parameters and $g(x)$ is the logistic sigmoid function.

Spearman: the Spearman coefficient (Zar, 1998) is an index that measures the similarity between two orders over the same set of objects. This coefficient is calculated as the Pearson's correlation between the ranks induced by the two compared orders. In particular, given an order, O_k, over a set of objects, χ^k, a rank, $r(\mathbf{x_i}, O_k)$, is assigned to each object, $\mathbf{x_i} \in \chi^k$, as a cardinal number according to its position in O_k. Objects at the same position are assigned consecutive cardinal numbers arbitrarily and receive the same rank calculated as the average of those consecutive numbers. For example, given four objects $\{\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}\}$ and the order $O = \{\mathbf{x_0} \succ \mathbf{x_1} \equiv \mathbf{x_2} \succ \mathbf{x_3}\}$, their ranks are $r(\mathbf{x_0}, O) = 0$, $r(\mathbf{x_1}, O) = r(\mathbf{x_2}, O) = \frac{1+2}{2}$ and $r(\mathbf{x_3}, O) = 3$ following the above procedure. Given a target order O_k over a set of objects, χ^k, and an order, $O_{\mathbf{w}}$, induced by a model $U^{\mathbf{w}}(\mathbf{x})$ over the same set of objects, the Spearman error function E_{SP} can be defined subtracting the Spearman coefficient (Zar, 1998) normalized into $[0, 1]$ from 1.0 as follows:

$$E_{SP}(U^{\mathbf{w}}(\mathbf{x}), O_k) = 3 \frac{\sum_{\mathbf{x_i} \in \chi^k} (r(\mathbf{x_i}, O_k) - r(\mathbf{x_i}, O_{\mathbf{w}}))^2}{|\chi^k|^3 - |\chi^k|} \tag{3.15}$$

where $|\chi^k|$ is the number of objects in the target order O_k.

Clearly, E_{SP} is 0 when $U^{\mathbf{w}}(\mathbf{x})$ induces the same order as O_k over χ^k and 1 when $U^{\mathbf{w}}(\mathbf{x})$ induces the exact opposite order. This error function can be applied to orders of any length without converting them into pairwise comparisons. When the target order is a pair, E_{SP} is equivalent to an inverted step function (see Figure 3.8b) whose value is 0 if the pair is correctly classified ($U^{\mathbf{w}}_{PN} > m$ with the margin $m = 0.0$) and 1 if the pair is incorrectly classified ($U^{\mathbf{w}}_{PN} < m$ with the margin $m = 0.0$).

The gradient of this error function with respect to model parameters equals zero on every differentiable point which makes it unsuitable for gradient-based optimization. However, it conforms the simplest possible error function for object ranking using a global search technique such as a genetic algorithm.

Sigmoidal rank-margin: A clear distinction can be made among the functions defined above. E_C depends strongly on $U^{\mathbf{w}}_{PN}$ changes and very little on whether $\{\mathbf{x_P}, \mathbf{x_N}\}$ is correctly classified or not. E_{RM} shows the similar dependency that is neutralized after $U^{\mathbf{w}}_{PN}$ is over a margin threshold (usually 1.0). E_S shows more sensitivity to changes from correctly to incorrectly classified $(\mathbf{x_P}, \mathbf{x_N})$ (or vice-versa) as changes of $U^{\mathbf{w}}_{PN}$ around the decision boundary ($U^{\mathbf{w}}_{PN} = m$) produce larger changes to E_S than changes far from that boundary. Finally, E_{SP} strengthens this dependency showing only changes when the $U^{\mathbf{w}}_{PN} - m$ sign changes.

A hybrid function is proposed combining characteristics of E_S and E_{SP} featuring a continuous improvement dependent on the distance between the model's output for preferred and non-preferred objects until the target pair is correctly classified, point at which the function becomes a step zero (see Figure 3.8f). The sigmoidal rank-margin error function E_{SRM} is defined as follows:

$$E_{SRM}(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N}) = \begin{cases} 0 & \text{if } U_{PN} > m \\ 1.0 - \frac{1}{1+e^{-\gamma(U^{\mathbf{w}}_{PN} - m)}} & \text{otherwise} \end{cases} \tag{3.16}$$

where m is the margin parameter (typically $m = 0.0$), γ is a positive weighting parameter that defines the slope of the logistic sigmoid function and $g(x)$ is the logistic sigmoid function. The gradient with respect to the model parameters is calculated as

$$\frac{\partial E_{SRM}(U^{\mathbf{w}}(\mathbf{x}), \mathbf{x_P}, \mathbf{x_N})}{\partial \mathbf{w}} = \begin{cases} 0 & \text{if } U^{\mathbf{w}}_{PN} > m \\ -\gamma\Big(1.0 - g\big(\gamma(U^{\mathbf{w}}_{PN} - m)\big)\Big) g\big(\gamma(U^{\mathbf{w}}_{PN} - m)\big) \frac{\partial U^{\mathbf{w}}_{PN}}{\partial \mathbf{w}} & \text{otherwise} \end{cases} \tag{3.17}$$

where m is the margin parameter (typically $m = 0.0$), and γ is a positive weighting parameter that defines the slope of the logistic sigmoid function. This error function is similar to E_{RM} as both are defined as piece-wise functions that are constant if the difference between the model's output for the target pair is above a given threshold. If the threshold is set to the same value, the only difference expected when using ANN backpropagation will be caused by the difference between linear and sigmoid derivatives. On the other hand, from the point of view of global optimization (e.g. genetic search) E_{SRM}, compared to E_{RM}, puts more reward to parameter configurations that exceed the threshold.

3.4.2 Support Vector Machines

A support vector machine (Cortes and Vapnik, 1995) is a binary classifier that linearly separates in a projected space $\phi(X)$ the data samples $\mathbf{x_i}$. The decision boundary of the classifier is given by a weight vector $\mathbf{w}$ which is found by solving:

$$\begin{aligned} \text{minimize:} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum \xi_i \\ \text{subject to:} \quad & \forall \mathbf{x_i} \in D, \mathbf{w} \cdot \phi(\mathbf{x_i}) z_i \geq 1 - \xi_i \\ & \forall i \ \xi_i \geq 0 \end{aligned} \tag{3.18}$$

where $z_i \in \{+1, -1\}$ is the class of sample $\mathbf{x_i}$, D is the complete set of training samples, ξ_i is a set of non-negative variables, C a weighting parameter, $\mathbf{w}$ the trained decision boundary and $||\mathbf{w}||$ its module. The above optimization conditions can be easily modified to accommodate pairwise comparisons as follows:

$$\begin{aligned} \text{minimize:} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\sum \xi_i \\ \text{subject to:} \quad & \forall (\mathbf{x^i_P}, \mathbf{x^i_N}) \in D, \mathbf{w} \cdot (\phi(\mathbf{x^i_P}) - \phi(\mathbf{x^i_N})) \geq 1 - \xi_i \\ & \forall i \ \xi_i \geq 0 \end{aligned} \tag{3.19}$$

where $\mathbf{x^i_P}$ and $\mathbf{x^i_N}$ represent the preferred and non-preferred objects in the pair i. This optimization can be solved by introducing Lagrangian multipliers and solving the transformed optimization problem using techniques from mathematical programming (Herbrich et al.,

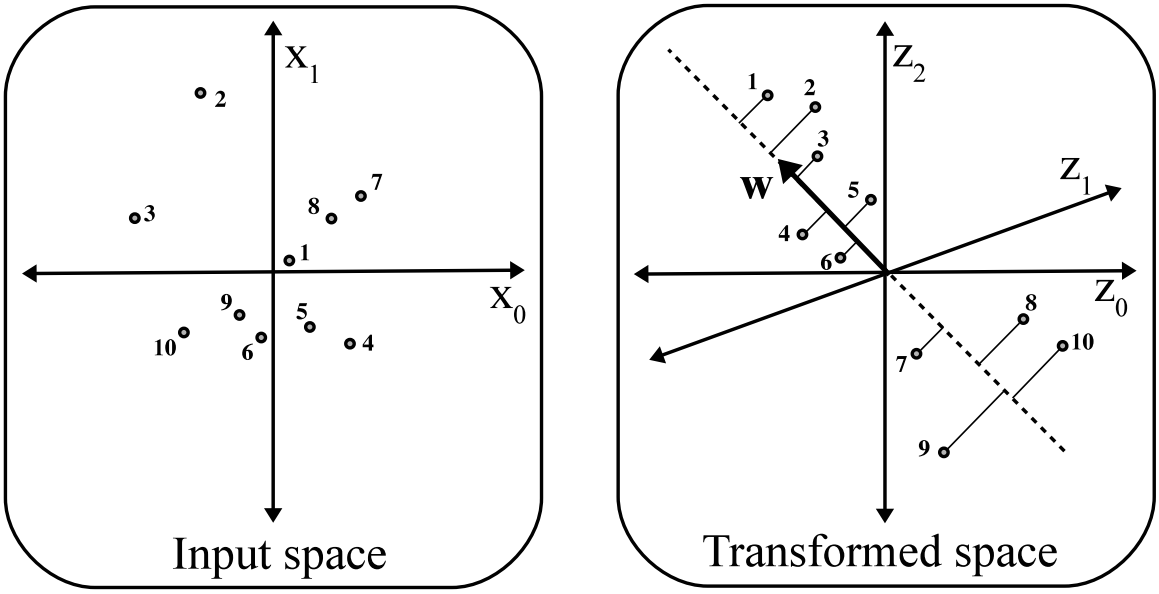


Figure 3.9: Support vector machine classification mechanism. Ten data samples in the input space (x_0, x_1) are transposed into a projected space $\phi(x_0, x_1) = (z_0, z_1, z_2)$. The projection of the data samples over the line defined by $\mathbf{w}$ defines the global order modeled by the SVM.

2000). The solution is expressed as a linear combination of some training samples which receive the name of *support vectors*:

$$\begin{aligned} \mathbf{w} &= \sum_{(\mathbf{x_P^i}, \mathbf{x_N^i}) \in S} \alpha_i (\phi(\mathbf{x_P^i}) - \phi(\mathbf{x_N^i})) \\ &\forall \alpha_i \; 0 \leq \alpha_i \leq C \end{aligned} \tag{3.20}$$

Once $\mathbf{w}$ is trained, given a pair of objects $\{\mathbf{x_i}, \mathbf{x_j}\}$ the SVM predicts that $\mathbf{x_i} \succ \mathbf{x_j}$ if $\mathbf{w} \cdot \phi(\mathbf{x_i}) > \mathbf{w} \cdot \phi(\mathbf{x_j})$. Graphically, this means that $\mathbf{w}$ specifies the direction along which the projected space is ordered (see Figure 3.9). Although the SVM creates a linear separation, this is defined on the transformed space defined by ϕ which yields more complex boundaries in the input space. Finally, replacing each calculation of the dot products by a kernel function $\kappa(\mathbf{x_i}, \mathbf{x_j}) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$, it is no longer required to explicitly project the input samples into the transformed space, thus saving on computational effort during training but more importantly enabling a large number of transformed non-linear spaces, including Hilbert spaces of infinite dimensions (Herbrich et al., 1999). In those cases, the vector $\mathbf{w}$ cannot be obtained and the output of the model can only be calculated in terms of the support vectors:

$$U^{\alpha,\kappa}(\mathbf{x}) = \mathbf{w} \cdot \phi(\mathbf{x}) = \sum_{(\mathbf{x_P^i}, \mathbf{x_N^i}) \in S} \alpha_i (\kappa(\mathbf{x}, \mathbf{x_P^i}) - \kappa(\mathbf{x}, \mathbf{x_N^i})) \tag{3.21}$$

While the training algorithm is completely different, a clear parallelism between SVMs and ANNs can be seen within this formulation: each of the support vectors can map to the connection weights of one of the hidden neurons in a 2-layer MLP while the kernels would represent the activation functions and, finally, the α_i parameters would map to

the connection weights of a single output neuron with a linear activation function. With regards to expressivity, ANNs show a greater potential because its hidden layers can be trained (instead of relying on training examples) and because several hidden layers can be stacked yielding to hierarchical transformations with an overall lower number of nodes (Bengio, 2009).

The five following kernels have been used in this thesis:

- Linear: $\kappa^1(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{x_i} \cdot \mathbf{x_j}$
- Polynomial 2^{nd} degree: $\kappa^2(\mathbf{x_i}, \mathbf{x_j}) = (\gamma \mathbf{x_i} \cdot \mathbf{x_j} + \beta)^2$
- Polynomial 3^{rd} degree: $\kappa^3(\mathbf{x_i}, \mathbf{x_j}) = (\gamma \mathbf{x_i} \cdot \mathbf{x_j} + \beta)^3$
- Gaussian radial basis function): $\kappa^G(\mathbf{x_i}, \mathbf{x_j}) = e^{-\gamma ||\mathbf{x_i} - \mathbf{x_j}||^2}$
- Sigmoid: $\kappa^S(\mathbf{x_i}, \mathbf{x_j}) = tanh(\gamma \mathbf{x_i} \cdot \mathbf{x_j} + \beta)$

For a more detailed description of the ranking SVM algorithm the reader is referred to (Herbrich et al., 1999; Joachims, 2002).

3.4.3 Cohen's Method

By contrast to the methods described above, Cohen's method (Cohen et al., 1999) does not solve the problem of OR by finding a utility function that induces a global order on the object space. Instead, this method creates a function that defines the probability of any object being preferred over any other object. Formally, given two objects $\mathbf{x_i} = [x_i^0, x_i^1, ..., x_i^{n-1}]$ and $\mathbf{x_j} = [x_j^0, x_j^1, ..., x_j^{n-1}]$, with $\mathbf{x_i}, \mathbf{x_j} \in [0,1]^n$ a Cohen's model is defined as follows:

$$C^{\mathbf{w}}(\mathbf{x_i}, \mathbf{x_j}) = \sum_{k=0}^{n-1} w_k R_k(\mathbf{x_i}, \mathbf{x_j}) + \sum_{k=0}^{n-1} w_{2*k+1} R_k(1 - \mathbf{x_i}, 1 - \mathbf{x_j}) \tag{3.22}$$

$$R_k(\mathbf{x_i}, \mathbf{x_j}) = \begin{cases} 1 & \text{if } x_i^k > x_j^k \\ 0 & \text{if } x_i^k < x_j^k \\ 0.5 & \text{otherwise} \end{cases} \tag{3.23}$$

The probability of an object $\mathbf{x_i}$ being preferred over an object $\mathbf{x_j}$ is determined by the comparison of each and every input feature (and their inverse values), independently. Each feature contributes with 1, 0 or 0.5 depending on whether its value is higher on the object $\mathbf{x_i}$, higher on the object $\mathbf{x_j}$ or equal in both. The contributions of all features are then aggregated through a weighted sum. Originally, Cohen et al. (1999) proposed an algorithm to train these weights $\mathbf{w}$ iteratively using the following equations:

$$w_k^{t+1} = \frac{w_k^t \beta^{L(R_k, O_t)}}{Z_t} \tag{3.24}$$

$$L(f(\mathbf{x}, \mathbf{y}), O_t) = 1 - \frac{1}{|O_t|} \sum_{(\mathbf{x_i}, \mathbf{x_j}) \in O_t} f(\mathbf{x_i}, \mathbf{x_j}) \tag{3.25}$$

where w_k^t is the value of the weight associated with feature x^k, R_k is the function defined in Equation 3.4.3, Z_t is a normalization factor such that $\sum_k w_k^{t+1} = 1$, O_t is the set of pairwise

preferences presented at iteration t, $|O_t|$ is the number of pairs in O_t and β is the learning rate. As proposed in (Kamishima et al.), the pairs in the training dataset S are presented successively until $L(C^{\mathbf{w}}, S)$ converges.

Cohen's method presents two advantages over the other methods examined in this thesis. First, the created model can model intransitive preferences/cycles (e.g. $\{\mathbf{x_A} \succ \mathbf{x_B}, \mathbf{x_B} \succ \mathbf{x_C}, \mathbf{x_C} \succ \mathbf{x_A}\}$) because it takes two objects as input. Second, the model is not affected by baseline problems (i.e. each participant with features in a separate range of values) as the actual values of the features are not directly connected to the output of the model (their effect is mediated through $R_k(\mathbf{x_i}, \mathbf{x_j})$. On the other hand, this method can create models that approximate functions of reduced complexity, as the output is always a linear combination of fixed single-feature comparisons.

3.5 Summary

This chapter described in detail the methods which are proposed in this thesis for automatic and reliable affect modeling. These methods are grouped into the four phases of a standard machine learning methodology, namely data collection, feature extraction, feature selection and model training. The next chapter corresponds to the first phase of the methodology, data collection, and introduces the datasets used to validate the rest of the methods in the remaining of this dissertation.

Chapter 4

Data Collection and Generation

The main contributions of this thesis arise from empirical evaluations of a number of modeling and feature extraction methods. We present the datasets used for these evaluations in this chapter, starting with the generation of several synthetic datasets and following with the description of two affect-related (i.e. *real*) datasets. The synthetic datasets were created for evaluating the modeling methods. Although synthetic data cannot capture all particularities of real affect datasets, it allows us to evaluate the behavior of the examined methods under particular known conditions which are unknown or uncontrollable in real data. In particular, the synthetic datasets generated here served to analyze the preference learning methods under different input data distributions (that resemble particular characteristics of affect datasets) and different target function complexities (as the methods are designed to find affect models of unknown complexity, we tested one linear, one quadratic and another non-linear target functions).

The real datasets were used to validate both the modeling and the feature extraction methods. We describe both datasets, Maze-Ball and DEAP, with sufficient detail to follow the results presented in the following chapters. However, as the data collection is not part of this thesis, we refer the reader to the original sources (Yannakakis et al., 2010; Martínez et al., 2010; Koelstra et al., 2012) for an in-depth analysis of the data.

4.1 Synthetic Data

In this dissertation we examined a number of preference learning methods that learn a computational model from a dataset. The fundamental premise for these methods (and for most other parametric learning techniques) is that the dataset is generated by sampling a *utility function* $U^*(\mathbf{x})$ that relates a set of observable features to the learning target (e.g. average heart rate and final score in a game to level of frustration of the player). By fitting a computational model to the dataset, these methods are approximating this utility function. In a real dataset, the utility function is unknown (e.g. frustration could be a linear function of heart rate or a quadratic function of final score) and the sampling cannot be easily controlled (e.g. the average heart rate of different participants in the dataset could be very similar or very different). Hence, we use synthetic datasets to evaluate the behavior of all methods under different utility functions and sampling distributions.

Note that the datasets required for the methods explored in this thesis — i.e. object ranking, a group of methods within preference learning — consist of a set of objects $S = \{\mathbf{x_0}, \mathbf{x_1}, ..., \mathbf{x_m}\}$ defined by n real-valued features, and a set of pairwise preferences (orders)

determined by the utility function. For any pair of objects, the utility function assigns a higher value to the object that is preferred within the pair. Formally, the utility function is related to the objects and pairwise preferences as follows:

$$U^*(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}; \forall(\mathbf{x_A}, \mathbf{x_B}) \in S, U^*(\mathbf{x_A}) > U^*(\mathbf{x_B}) \rightarrow \mathbf{x_A} \succ \mathbf{x_B} \tag{4.1}$$

where $U^*(\mathbf{x})$ is the utility function, $(\mathbf{x_A}, \mathbf{x_B})$ is any pair of objects in S and $\mathbf{x_A} \succ \mathbf{x_B}$ denotes that $\mathbf{x_A}$ precedes or is preferred over $\mathbf{x_B}$.

We generated nine synthetic datasets by combining three known utility functions with three methods for sampling the feature space. The utility functions are a linear combination of features, $U^1(\mathbf{x})$, a quadratic weighted sum, $U^2(\mathbf{x})$, and a feed-forward neural network with two logistic hidden layers (with 5 and 10 neurons, respectively), $U^{ANN}(\mathbf{x})$. Formally, the functions are defined as follows:

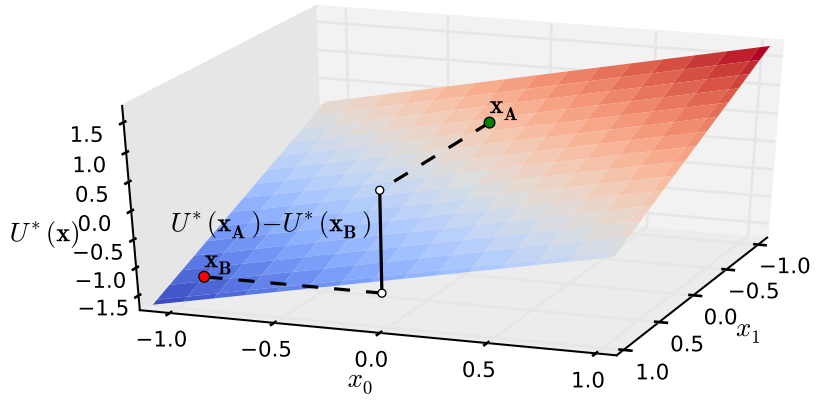
$$U^1(\mathbf{x}) = \sum_{i=0}^{n-1} x_i w_i \tag{4.2}$$

$$U^2(\mathbf{x}) = (\sum_{i=0}^{n-1} x_i w_i)^2 \tag{4.3}$$

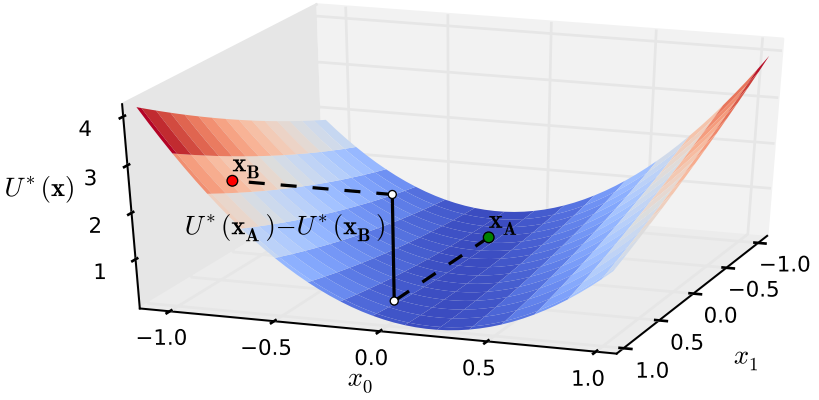
$$U^{ANN}(\mathbf{x}) = \sum_{k=0}^{9} w^k \left(s(\sum_{j=0}^{4} w_k^j \left(s(\sum_{i=0}^{n-1} x_i w_j^i + \theta_j) \right) + \theta_k) \right); s(x) = \frac{1}{1+e^{-x}} \tag{4.4}$$

where the parameters w and θ are sampled randomly from a Gaussian distribution ($\mu = 0$, $\sigma = 1$) for each utility and $\mathbf{x} = [x_0, x_1 ... x_{n-1}]$ is an object defined by n features. Figure 4.1 depicts an example of these synthetic utility functions in a 2-feature object space. While many other functions could have been tested, the selected set cover distinct levels of complexity that could be found in real datasets.

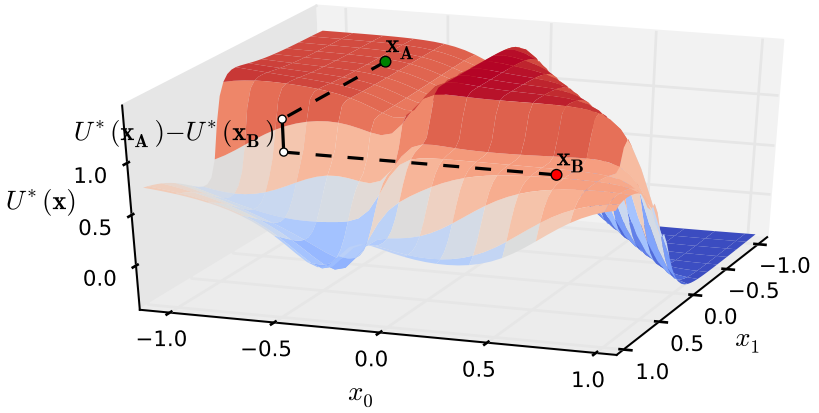
For each of the three functions, 3 sets of objects are sampled. The number of features used is all synthetic sets is 10 which represents a reasonable number of features to train a model of affect using a real dataset taking in consideration the (typically) small size of affect-related datasets. The first object set $S_{\mathcal{U}_{-1}^{+1}}$ contains 10000 pairs of objects sampled from a uniform distribution ($min = -1, max = +1$) while the second $S_{\mathcal{N}_{\mu}^{0.1}}$ contains 10000 pairs of objects sampled from 20 Gaussian distributions with different means and standard deviations equal to 0.1. The second set in contraposition to the first showcases a challenge that affect datasets (and physiological in particular) present: data from separate participants or sessions have different baselines. Figure 4.3 shows an example of the distribution of object pairs in a 2-dimensional feature space for each of the three sampling method. The third set $S^d_{\mathcal{U}_{-1}^{+1}}$ is created by initially sampling a larger number of pairs from a uniform distribution ($min = -1, max = +1$) and then selecting two subsets of 5000 pairs each with the shortest and largest *utility differences*, respectively. The utility difference between two objects $U^*(\mathbf{x_A}) - U^*(\mathbf{x_B})$ defines the strength of the pairwise preference between them. This is exemplified in Figure 4.1; note that pairs of objects that are close in the feature space typically present similar utility functions (as utility functions are expected to be smooth) but the contrary statement does not necessary hold true for pairs of objects that are separated in the feature space as seen in the highlated pair of objects in Figure 4.1c. In a real



(a) $U^1(\mathbf{x})$

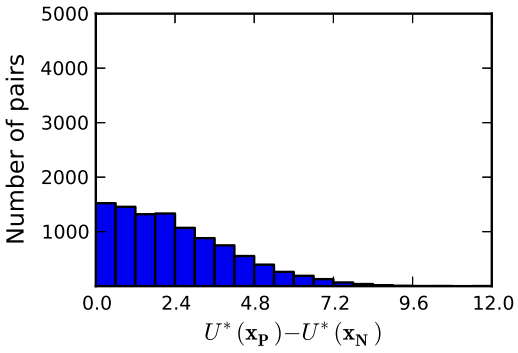


(b) $U^2(\mathbf{x})$

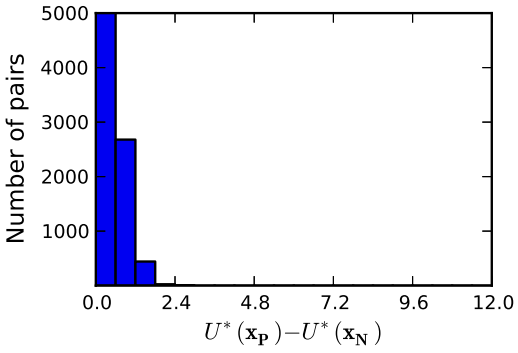


(c) $U^{ANN}(\mathbf{x})$

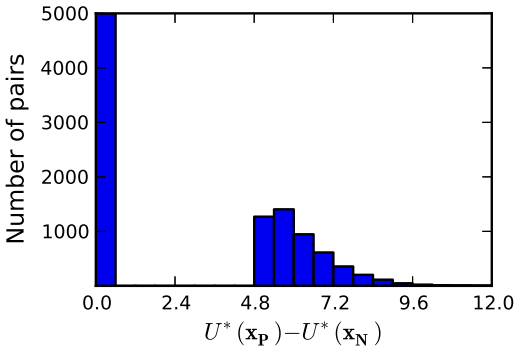
Figure 4.1: Synthetic utility functions. Utility functions for two-feature objects are displayed. Two objects ($\mathbf{x_A}$ and $\mathbf{x_B}$) and the utility difference between them $U^*(\mathbf{x_A}) - U^*(\mathbf{x_B})$ are depicted.



(a) $S_{\mathcal{U}_{-1}^{+1}}$

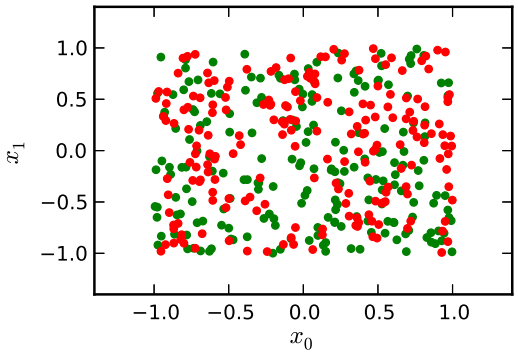


(b) $S_{\mathcal{N}_{\mu}^{0.1}}$

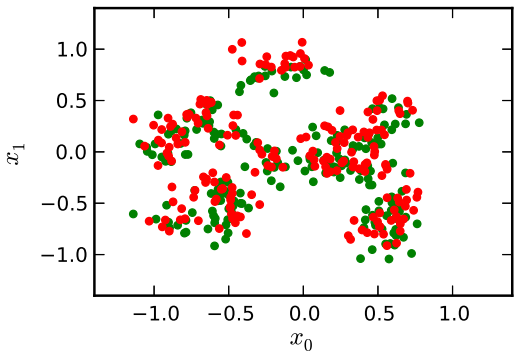


(c) $S_{\mathcal{U}_{-1}^{+1}}$

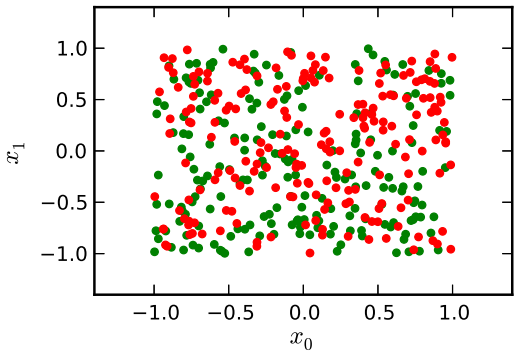
Figure 4.2: Example of the frequency distribution of the utility difference between preferred and non-preferred objects in pairs of the synthetic datasets with sets of objects sampled from a uniform distribution ($S_{\mathcal{U}_{-1}^{+1}}$), several Gaussian distributions with different means ($S_{\mathcal{N}_{\mu}^{0.1}}$) and a uniform distribution forcing two differentiated levels of utility difference ($S_{\mathcal{U}_{-1}^{+1}}$).



(a) $S_{\mathcal{U}_{-1}^{+1}}$



(b) $S_{\mathcal{N}_{\mu}^{0.1}}$



(c) $S^d_{\mathcal{U}_{-1}^{+1}}$

Figure 4.3: Distribution of object features in the synthetic datasets. These graphs depict 400 objects with two features (x_0, x_1) sampled in pairs from a uniform distribution ($S_{\mathcal{U}_{-1}^{+1}}$), 20 Gaussian distributions with different means ($S_{\mathcal{N}_{\mu}^{0.1}}$) and a uniform distribution forcing two differentiated levels of utility difference ($S^d_{\mathcal{U}_{-1}^{+1}}$). Green and red dots represent, respectively, the preferred and non preferred objects of each pair.

dataset, a pairwise preference with a low utility difference may correspond to an unclear preference, i.e. the two options compared have a very similar associated utility (e.g. level of frustration) and the user cannot clearly discern which one is higher (which one is more frustrating). When dealing with human subjective self-reports in experimental settings, and especially with affective states, a number of unclear preferences are always expected. Hence, this third sampling method aid us in evaluating the tolerance of the PL methods examined to this phenomenon. Figure 4.2 depicts an example of the frequency distribution of the utility difference for the three sets of objects used.

Finally, after the pairs of objects are sampled, the pairwise preferences are defined using Equation 4.1 after Gaussian noise δ is added to the input features. The amount of noise is adjusted to create around a 5% of misclassified pairs, that is for 5% of the pairs the utility function assigns a higher value to the object that is not preferred. Formally, for each of these 5% object pairs $(\mathbf{x_A}, \mathbf{x_B})$, the dataset contains the preference $\mathbf{x_A} \prec \mathbf{x_B}$ with:

$$U^*(\mathbf{x_A}) > U^*(\mathbf{x_B}); U^*(\mathbf{x_A} + \delta_A) < U^*(\mathbf{x_B} + \delta_B) \tag{4.5}$$

4.2 Maze-Ball

The Maze-Ball dataset contains data gathered through a game-based experimental survey. This dataset has been used in a number of articles which amount to a significant part of the contributions of this thesis (Martínez et al., 2010; Schwartz et al., 2009; Martínez et al., 2009; Martínez and Yannakakis, 2010; Yannakakis et al., 2010; Martínez et al., 2011; Martínez and Yannakakis, 2011a,b).

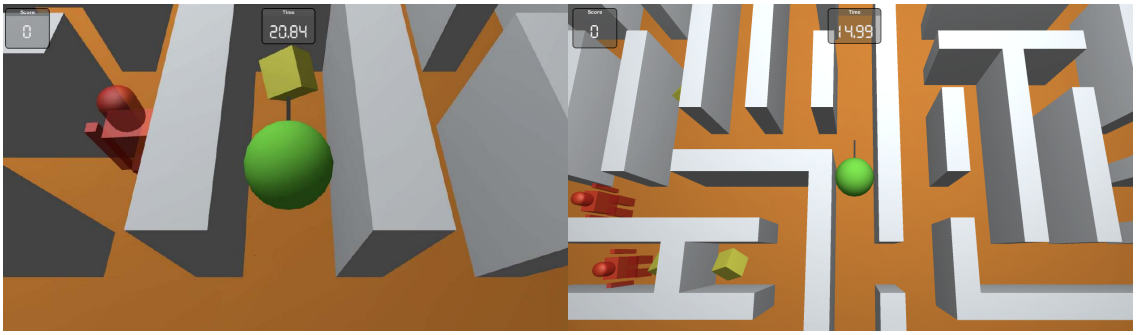
4.2.1 Materials and Set-up

Maze-Ball is a three-dimensional prey/predator *PacMan*-like game (see Figure 4.4). The player (prey) controls a ball which moves inside a maze where 10 red-colored opponents (predators) move around. The goal of the player is to maximize her score by gathering as many gold tokens, scattered in the maze, as possible while avoiding the enemies in a predefined time window of 90 seconds. The 90 second play-time window is designer-driven and attempts to maintain a good balance between sufficient gameplay interaction and the player's cognitive load. On one hand, a short game is required to minimize memory-dependent effects of post-experience on questionnaire items and the total time required for the experience to run; on the other hand the game should provide sufficient interaction for the requested affective states to be elicited.

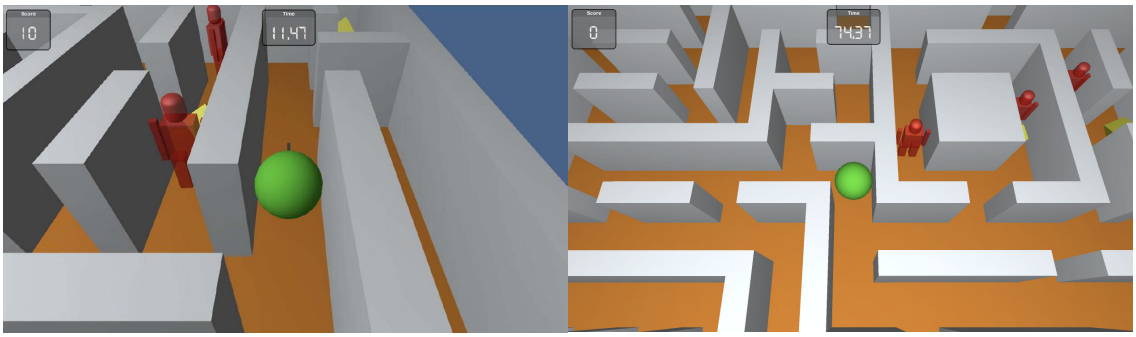
The game implements a dynamic camera controller that adapts the view of the graphical world given a desired 3-parameter profile: *distance* to the player, *height* above game the level and *frame coherence* (i.e. smoothness of the camera movements). Eight variations of the game are deployed as a result of changing the camera profile while keeping the game design, level design, and game mechanics unaltered. For each of the three camera control variables, two states ('*High*' and '*Low*') are selected[1]. All game variants are illustrated in Figure 4.4.

The purpose of using Maze-Ball for collecting affective information is two-fold: first, it consists of a minimal interface for an enjoyable game (arrow keys for controlling the character) and a simple visual environment. Single-hand game control via the keyboard allows

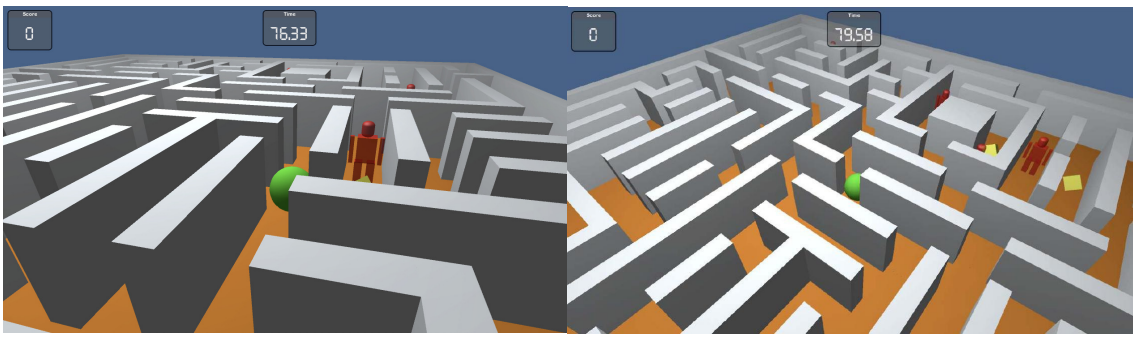
[1]The *Low* and *High* values selected for distance, height and frame coherence are respectively 2.5 and 6; 6 and 15; and 0.01 and 0.35



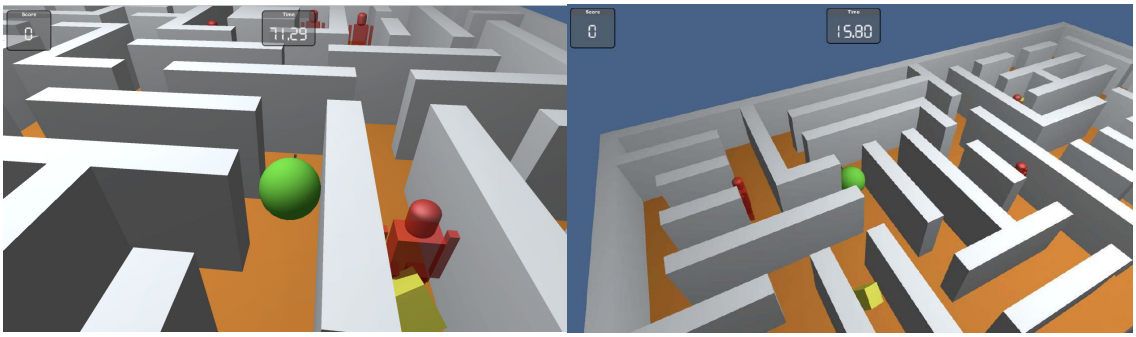
(a) Game variant 1: (*Low*, *Low*, *Low*) (b) Game variant 2: (*High*, *Low*, *Low*)



(c) Game variant 3: (*Low*, *High*, *Low*) (d) Game variant 4: (*High*, *High*, *Low*)



(e) Game variant 5: (*Low*, *Low*, *High*) (f) Game variant 6: (*High*, *Low*, *High*)



(g) Game variant 7: (*Low*, *High*, *High*) (h) Game variant 8: (*High*, *High*, *High*)

Figure 4.4: Screen-shots of Maze-Ball. All eight game variants of Maze-Ball generated with a different camera profile given by the tuple (height, distance, frame coherence).

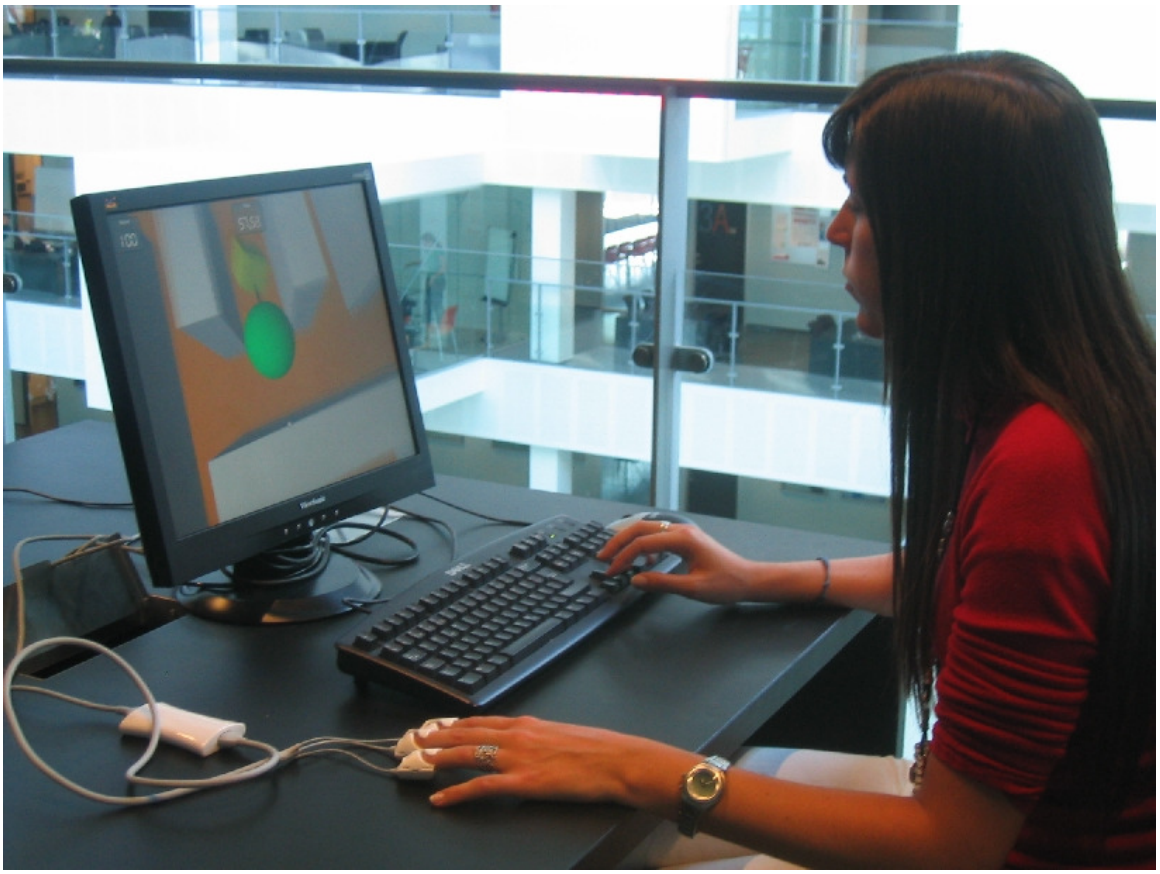


Figure 4.5: The set-up of the experiment. The Maze-Ball game is viewed on screen; the IOM bio-sensing hardware is placed on the subject's left hand in this picture.

for the unobtrusive placement of biofeedback sensors to the free hand which is essential for physiological recordings. Second, there is a direct effect of the amount of information available to the player about the world — via camera viewpoint — on her movement strategy and consequently her experience. For instance, in the top-down view of the full maze the player has complete global information about the world for planning out the path along the maze. This viewpoint, however, may not be optimal for controlling the character's local movement as the character takes up only a small fraction of the entire screen. A close view, such as the first person view, makes moving the character and avoiding enemies easier but strategically moving along the maze harder. Thus, an experimental survey can provide rich data for affect modeling and insights on the effect of virtual camera settings to player experience.

The participants play the game with the IOM biofeedback device placed on the fingertips (two electrodes for recording skin conductance and one photo sensor for recording blood volume pulse) of the non-dominant hand as seen in Figure 4.5. By using small and accurate commercial apparatus like the IOM biofeedback device in the least intrusive way, experiment psychological effects caused by the presence of recording devices are minimized. Furthermore, a number of gameplay metrics are logged including the position of the player and enemies, the position of the camera, the keys pressed and relevant game events such as collecting pellets.

4.2.2 Experimental Protocol

Physiological signals and subject's emotional preferences were acquired for Maze-Ball through the following game survey experiment. Thirty six subjects (males: 80%) aged from 21 to 47 years (mean and standard deviation of age equal 27.2 and 5.84 respectively) participated in the experiment. Participants were recruited at different universities in Copenhagen area and received no payment. Each participant was led into the experiment room, briefed about the experiment and the sensors were placed on her non-dominant hand. After the experimenter left the room, the computer displayed the instructions, and a consent and demographic forms. After the forms were filled in, the participant played a tutorial game. Then, each participant played a predefined set of eight games for 90 seconds each — the games differ in the levels of *distance*, *height* and *frame coherence* — and reported their experience after each completed pair of games. Between the games of the same pair, the participants rest for at least 15 seconds, time after which they can start the next game.

The number of experiment participants is determined by $C_2^9 = 36$, this being the required number of all combinations of 2 out of 9 game variants. Each participant played one pair of variants in both orders and other two pairs with different game variants. The games played by each participant are assigned in such a way that, in total, 4 preference instances should be obtained for each pair of the 9 game variants in both orders (2 preference instances per playing order). Given that, a number of 4 pairs of games is required to be played by each participant resulting to $36 \cdot 4 = 144$ game pair preferences. In addition to the 8 game variants generated by solely varying the camera profile, a ninth variant without visible walls is included to test the impact of walls in camera-profile preferences. Pairs containing this variant are not included in this thesis yielding a total of 112 valid pairs for modeling affective states.

4.2.3 Participants Self-assessment

After each completed pair of games, A and B, participants report their emotional preference using a 4-alternative forced choice protocol:

- game A [B] was/felt more E than game B [A] game (*cf.* 2-alternative forced choice);
- both games were/felt equally E or
- neither of the two games was/felt E.

Where E is the user (affective and cognitive) state under investigation and contains *fun*, *challenging*, *boring*, *frustrating*, *exciting*, *anxious* and *relaxing* (see Figure 4.6). The selection of these seven states is based on their relevance to computer game playing with parameterized camera positioning. The first five have been previously used in game-related user studies (Mandryk and Atkins, 2007) while the last two are included for maintaining a uniform covering of the arousal-valence appraisal space (Russell, 1980).

Note that participants are not interviewed but are asked to fill in a comparison questionnaire, minimizing interviewing effects. The 4-alternative forced choice protocol is used since it offers several advantages for subjective emotion capture: explicit comparisons can potentially minimize participants' subjective notions of scaling and allow a fair comparison between the answers of different participants while also making explicit the "no preference" cases concealed by 2-AFC. The 4-AFC and 2-AFC protocols have been successfully utilized

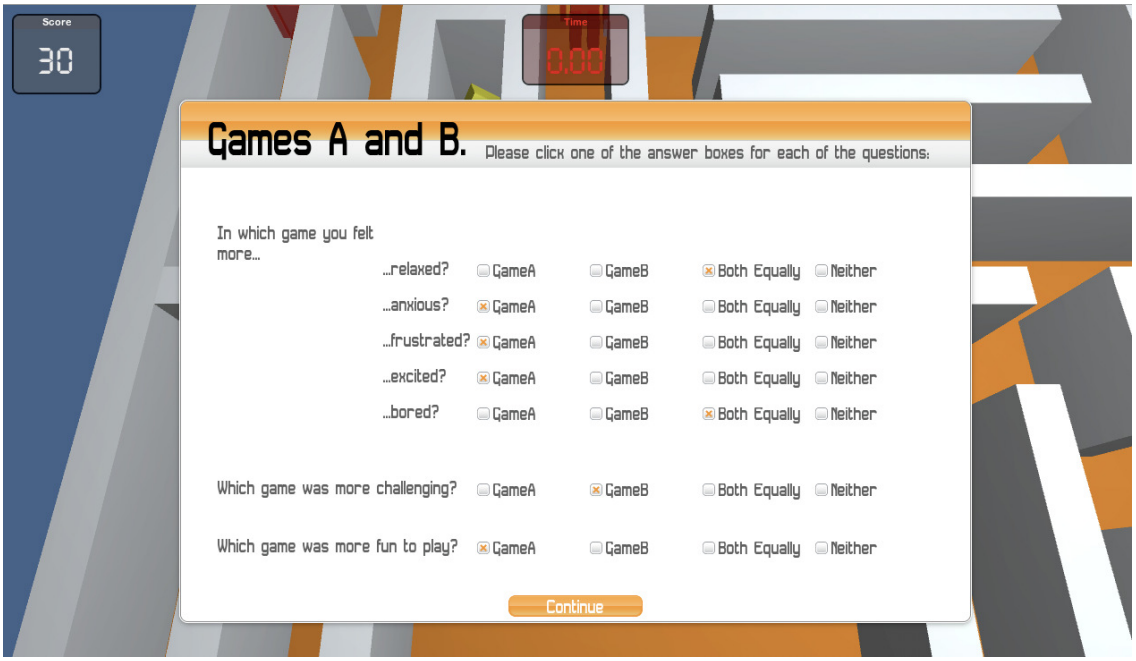


Figure 4.6: Preference questionnaire used in the Maze-Ball game survey.

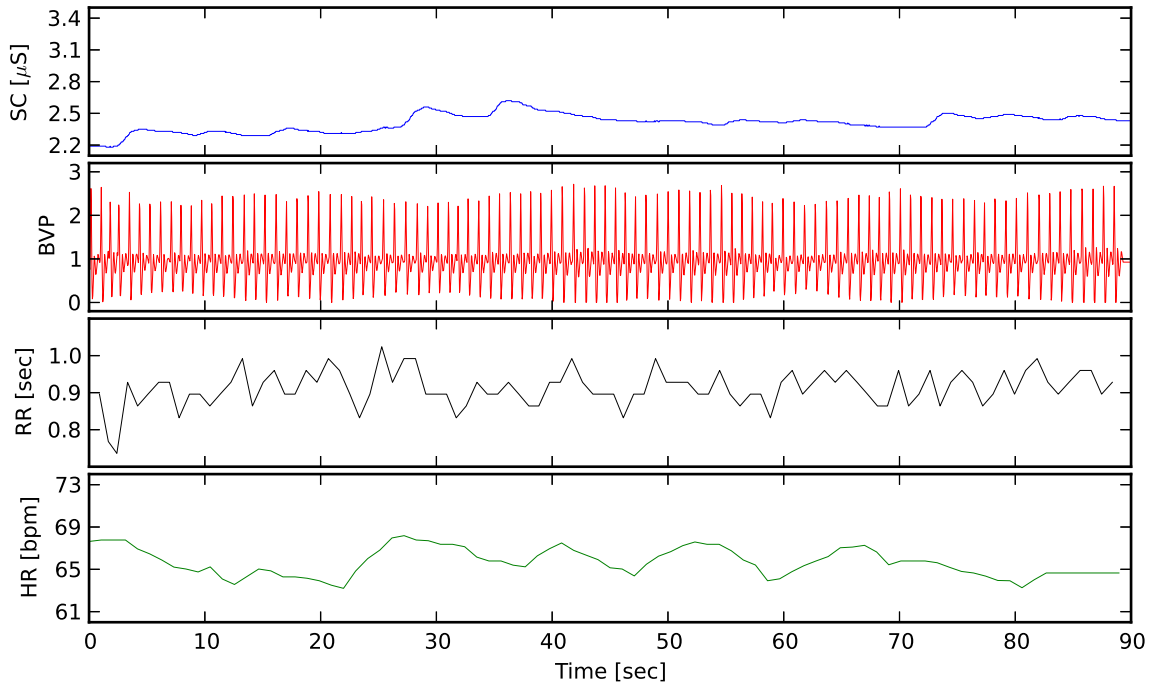
to provide data for building accurate computational models of reported emotional preferences (Yannakakis et al., 2008; Yannakakis and Hallam, 2008, 2011). The "no preference" cases are not used for modeling which leaves 92, 90, 90, 86, 83 and 54 pairs, respectively, for reported challenge, fun, frustration, relaxation, anxiety, excitement and boredom. The number of boredom pairs is rather too small for modeling but we will include it for completeness.

4.2.4 Signals and Features

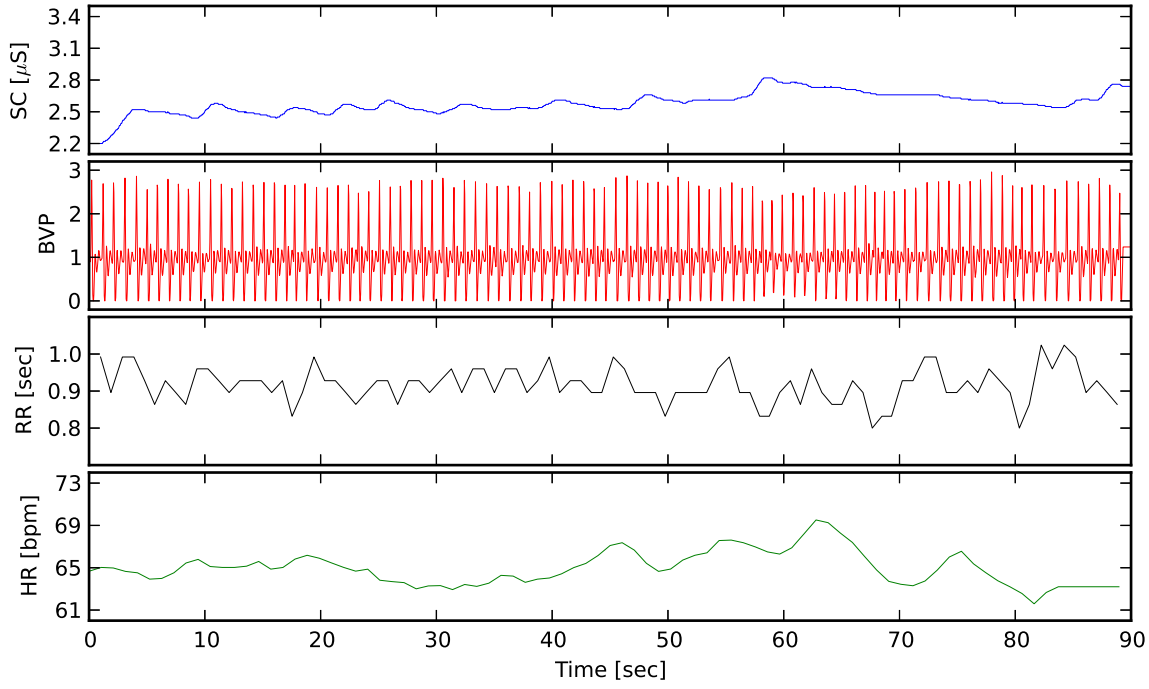
In this section we describe the data collected on the Maze-Ball game survey and used in the following chapters: the raw physiological signals used in Chapter 6, the sequences of gameplay and physiological events used in Chapter 7 and the ad-hoc statistical features used throughout all three results chapters.

Physiological signals

For each game blood volume pulse, and skin conductance, were collected in real-time at a sample rate of 31.25 Hz (32 ms sampling interval). Heart rate is computed using a 5-second sliding window by extrapolating the inter-beat time intervals detected in the BVP signal (see Figure 4.7). Measurement units for HR and SC are, respectively, heart beats per minute (bpm) and micro-Siemens (μS) whereas BVP is a relative measure of blood vessel pressure.



(a) Participant no. 5: Game variant 6



(b) Participant no. 5: Game variant 1

Figure 4.7: Example of the SC, BVP, RR and HR signals obtained in a pair of Maze-Ball games: a participant (no. 5) plays a game of *High* height, *Low* distance and *High* frame coherence (a) and then a game of *Low* height, *Low* distance and *Low* frame coherence (b). The participant expressed a fun, boredom and relaxation preference for the game variant 6 whereas expressed a challenge, excitement, frustration and anxiety preference for game variant 1.

Sequences of Events

A data-sequence is created from the logs of each game by concatenating all the events logged in temporal order. The following list describes the events included for this dissertation:

- Performance Events
  - Player collects a pellet ($): 10 identical pellets are placed in different areas of the maze enforcing a difference of at least few seconds between two pellets. This event is picked as it is expected to have an impact on reported challenge and fun (among other reported user states).
  - Enemy hits the player (E): 14 enemies follow predefined paths guarding a pellet causing this event to occur very close in time with the $ events frequently. Enemy hits are selected as events since enemies are critical to player experience in a prey/ predator game.
  - Countdown starts (t^{10}): when entering the last 10 seconds of the game the timer changes its color *rushing up* the player. This event occurs exactly once in each game, thus it does not provide sufficient information about the experience per se. However, sequences combining this event with physiological events or other gameplay events are expected to have a direct impact to reported anxiety and excitement.
- Navigation Events
  - Moving to a new area of the maze ($m^0, \ldots, m^7$): although there are not explicit boundaries between areas of the maze, 8 different sectors can be distinguished based on the different wall layout, placement of the pellets and movement of the enemies which, in turn, represent different degrees of difficulty. These events are expected to have a direct impact on the challenge reports.
  - Press an arrow key (▲, ▼, ◄, ►): pressing the right and left arrows make the ball turn if it is located in a corner; the down key forces the ball to turn 180^o and the up arrow has no effect. Each single one of these events most likely holds a tiny piece of information about user experience; however, sequences combining many of these events may point to more complex navigation patterns with a potential impact on experience.
  - Inactivity for more than 1 second (*Stop*): the player avatar is moving forward at any time unless it hits a wall. In that case, the ball will only continue moving if the player turns. When the ball is stopped for 1 second, the event is logged. It could indicate that the player is planning a strategy or waiting for an enemy to move away from a pellet. Thus, this event is relevant for the identification of a player's behavioral patterns and, indirectly, for affect detection.
- Physiological Events
  - Difference between two inter-beat intervals (*RR intervals*) is greater than 50 ms (r^{+50}, r^{-50}): the heart beats are detected from the BVP signal and when two consecutive inter-beat intervals differ for more than 50 ms, an event is logged. The threshold of 50 ms is commonly used in affective and medical studies Goldberger et al. (2001); Yannakakis et al. (2010) as an indicator of arousal which in turn is one of potential identifiers of the affective states examined.

– SC increase/decrease ($s^{\Uparrow}, s^{\Downarrow}$): sudden changes in the SC signal are detected and logged as events. They are normally detected in pairs: after $s^{\Uparrow}$ the SC will increase for a while and $s^{\Downarrow}$ will be logged when it starts decreasing. These SC signal events are picked because they suggest changes in sympathetic activity and, thereby, may relate to reported user experience.

Ad-Hoc Features

This section lists the common feature extractors used for skin conductance, blood volume pulse and heart rate (ad-hoc physiological features). It also introduces the game metrics (ad-hoc context features) proposed for the Maze-Ball game.

Physiology: same extractors are applicable to the SC and HR signals while BVP presents several signal-dependent as seen in the list below. The choice of those specific statistical feature extractors is made in order to cover a decent amount of the BVP, SC and HR signal dynamics proposed in the majority of previous studies in the field (Picard et al., 2001; Goldberger et al., 2001; Yannakakis and Hallam, 2008).

- **SC and HR** ($\alpha \in \{SC, HR\}$)**:** Average $E\{\alpha\}$, standard deviation $\sigma\{\alpha\}$, maximum $\max\{\alpha\}$, minimum $\min\{\alpha\}$, the difference between maximum and minimum signal recording D^{α} =$\max\{\alpha\}-\min\{\alpha\}$, time when maximum α occurred $t_{\max}\{\alpha\}$, time when minimum α occurred $t_{\min}\{\alpha\}$ and the difference $D_t^{\alpha} = t_{\max}\{\alpha\} - t_{\min}\{\alpha\}$; autocorrelation (lag equals 1) of the signal ρ_1^{α} and mean of the absolute values of the first and second differences of the signal (Picard et al., 2001) ($\delta_{|1|}^{\alpha}$ and $\delta_{|2|}^{\alpha}$ respectively); initial, α_{in}, and last, α_{last}, α recording, the difference between initial and final α recording D^{α} =$\max\{\alpha\}-\min\{\alpha\}$ and Pearson's correlation coefficient R_{α} between raw α recordings and the time t at which data were recorded.

- **BVP:** Average $E\{BVP\}$, standard deviation $\sigma\{BVP\}$, mean of the absolute values of the first and second differences of the signal ($\delta_{|1|}^{BVP}$ and $\delta_{|2|}^{BVP}$ respectively), average and standard deviation of the inter-beat amplitude $E\{IBAmp\}$ and $\sigma\{IBAmp\}$. Moreover, given the inter-beat time intervals (RR intervals) of the BVP signal the following heart rate variability extractors are proposed:

  – HRV- time domain: the average and standard deviation of RR intervals $E\{RR\}$ and $\sigma\{RR\}$, the fraction of RR intervals that differ by more than 50 msec from the previous RR interval $pRR50$ and the root-mean-square of successive differences of RR intervals RMS_{RR} (Goldberger et al., 2001).

  – HRV - frequency domain: the frequency band energy values derived from power spectra obtained using the Lomb periodogram (Moody, 1993); energy values are computed as the integral of the power of each of the following two frequency bands, relevant for short experiences (Force, 1996): High Frequency (HF) band: (0.15, 0.4] Hz and Low Frequency (LF) band: (0.04, 0.15] Hz. In addition, the ratio LF/HF and the normalized values $LF/(LF + HF)$ and $HF/(LF + HF)$ are also included as recommended in (Force, 1996).

Game context: several events and game state variables are logged for each game, including elements of the game state and the player's inputs (keystrokes). A list of ad-hoc features

that attempt to capture the relevant elements of the experience in this specific game are listed in detail in this section.

- **Performance**: the final score (S), the percentage of the grid explored (G), and the percentage of paths covered several times (P) (calculated by dividing the number of explored cells of the grid by the times the player leaves a cell).
- **Time**: average and standard deviation of time intervals the player stays in certain cell (t^c) and the number of these intervals that are greater than 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0 seconds ($t^c_{0.5}$, $t^c_{0.6}$, $t^c_{0.7}$, $t^c_{0.8}$, $t^c_{0.9}$, $t^c_{1.0}$ respectively).
- **Space**: average and standard deviation of the Euclidean distance between the ball and the closest token (D^ϵ_t) and between the ball and the closest enemy (D^ϵ_e), average and mean of the standard deviation of the Euclidean distance to all enemies ($D^\epsilon_{\forall e}$), average and standard deviation of the manhattan distance between the ball and the closest token (D^μ_t) and between the ball and the closest enemy (D^μ_e), average and the mean of the standard deviation of the manhattan distance to all enemies ($D^\mu_{\forall e}$).
- **Input**: number of right (90°), left (−90°) and 180° turns divided by the times the right, left and down key arrows were pressed respectively (ω_{90}, ω_{-90}, ω_{180}), number of times the up arrow (K_{up}) key was pressed, average and standard deviation of the time that either the right or the left arrow keys were held down (t^K).

4.3 DEAP

DEAP (a database for emotion analysis using physiological signals) is a dataset that gathers brain activity and physiological reactions of users to music videos. This publicly available dataset was collected by Koelstra et al. (2012) and a first exploration of the data and a thorough description was included in that work.

4.3.1 Materials and Set-up

Forty music video-clips were selected to elicit dissimilar levels of arousal and valence. Initially, a larger set of videos were chosen manually (60) and automatically (60) based on crowdsourced affect labels. After cropping the videos to 1-minute long segments, they were rated by a number of volunteers on 9-point scales for valence, arousal and dominance. The 40 videos with strongest ratings and minimal variation across volunteers were selected.

The videos were presented in randomized sequences to participants while several sensors connected to one hand, face and back, and a electro-encephalographic cap recorded a myriad of physiological signals including blood volume, skin conductance, skin temperature, electroencephalogram, respiration rate and electromyography of the face using the Biosemi ActiveTwo system. The experiments were performed in two laboratory environments with controlled illumination. The videos and questionnaires were automatically presented in a computer screen.

4.3.2 Experimental Protocol

Twenty two healthy participants (50% female), aged between 19 and 37 (mean and standard deviation of age 26.5 and 3.99), participated in the experiment. The original dataset con-

tains additional participants but have been left out due to differences in the physiological sensors used.

Prior to the experiment, each participant signed a consent form, filled out a demographic questionnaire and read a set of instructions informing them of the experiment protocol and the meaning of the different scales used for self-assessment. The participant was only led into the experiment room after the instructions were clear. After the sensors were placed and their signals checked, the participants performed a practice trial to familiarize themselves with the system. Next, the experimenter started the physiological signals recording and left the room, after which the participant started the experiment by pressing a key on the keyboard. The experiment started with a 2 minute baseline recording, during which a fixation cross was displayed to the participant. Then the 40 videos were presented in 40 trials in a different order for each participant. Each trial consists of the following steps:

1. A 2-second screen displaying the current trial number to inform the participants of their progress.
2. A 5-second baseline recording (fixation cross).
3. The 1-minute display of the music video.
4. Self-assessment questionnaire.

After 20 trials, the participants took a short break.

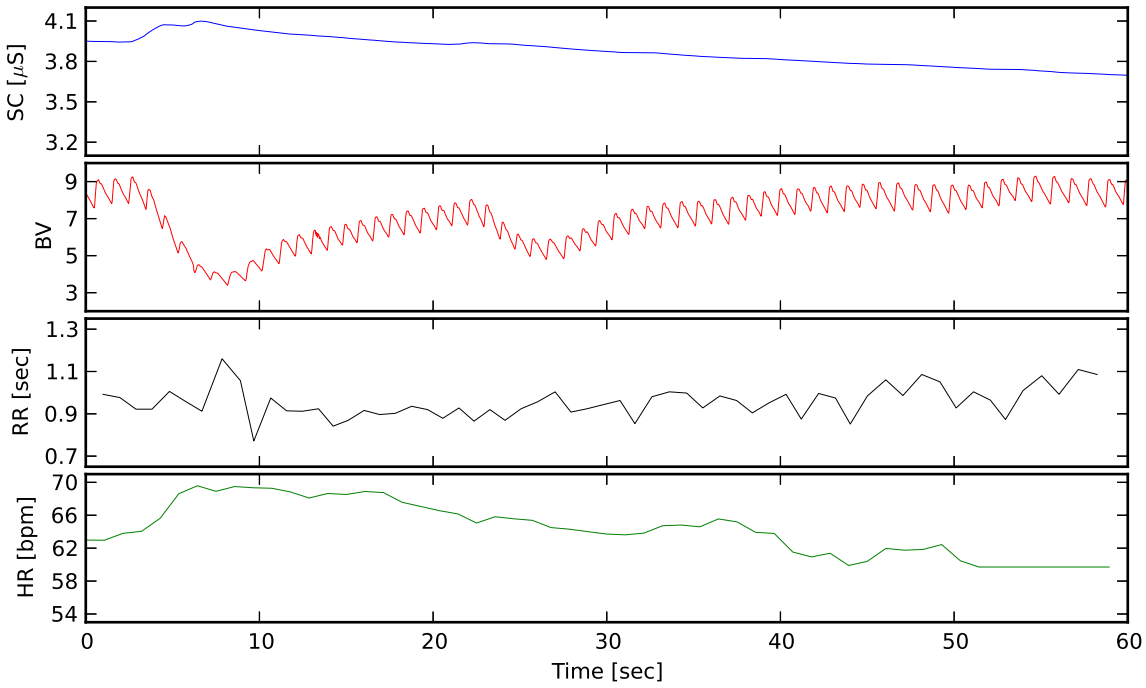
4.3.3 Participants Self-assessment

At the end of each trial, participants performed a self-assessment of their level of *arousal*, *valence* and *dominance* using a continuous 9-point scale visualized via self-assessment manikins (Morris, 1995). A similar scale but with a different visualization (thumbs down/thumbs up symbols) was provided to report *liking*. Finally, after the experiment, participants were asked to rate their familiarity with each of the songs on a 5-point scale. The questionnaires are displayed on the same screen and participants used the mouse to choose the answers.

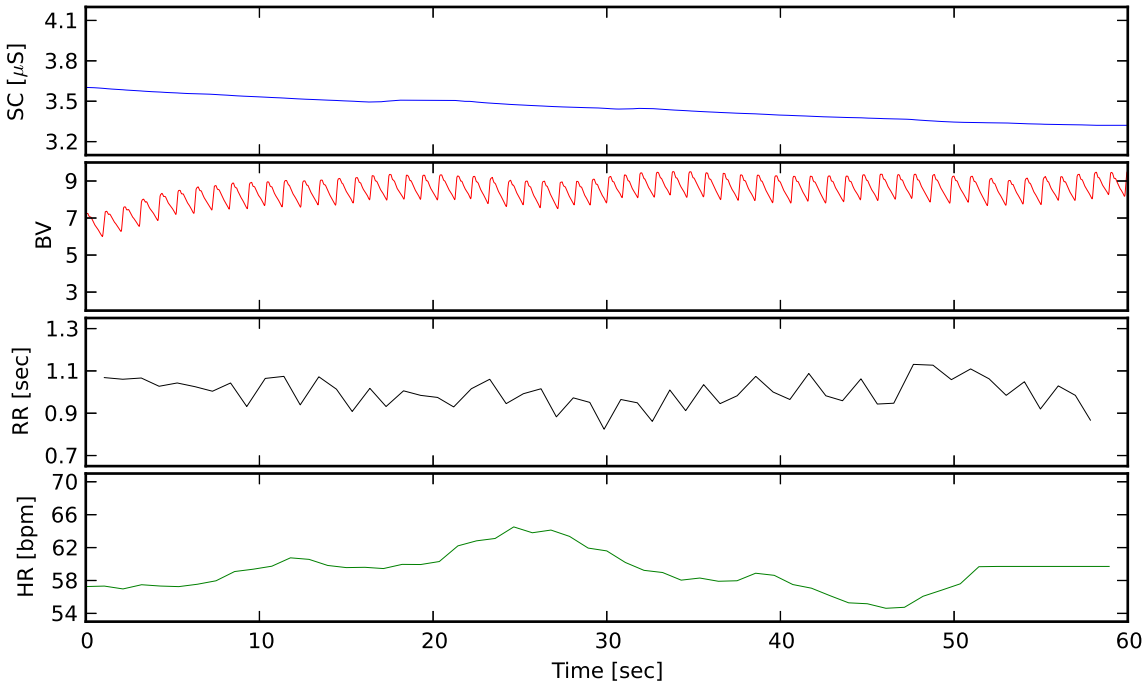
From the arousal, valence and liking ratings, three separate pairwise self-reports are created by including all consecutive pairs that are rated with a difference greater than 1. Note that $\sum_{i=1}^{39} i$ pairs could be extracted from each participant; however, only consecutive pairs are chosen to minimize the effects of rating scales as it could be expected that participants have taken into account the previous rating to rate the current video. Additionally, differences between two ratings lower than 1 are considered as *unclear preferences* and thus not included. The threshold of 1 is chosen as it is the smallest unit on the visual scale (SAM). In total, 590, 547 and 542 pairs are extracted for valence, liking and arousal respectively.

4.3.4 Signals and Features

In this section we describe the data collected on the DEAP survey and used in the following chapters: the raw physiological signals used in Chapter 6 and the ad-hoc statistical features used in Chapter 5 and Chapter 6.



(a) Participant no. 1: first trial



(b) Participant no. 1: second trial

Figure 4.8: Example of the SC, BV, RR and HR signals obtained in a pair of DEAP videos: a participant (no. 1) watches (a) a video with high arousal and valence (*Song 2* by *Blur*) and then (b) a video with medium arousal and high valence (*What a Wonderful World* by *Louis Armstrong*). The participant expressed a slightly higher valence, higher arousal and higher liking for the second video (differences of 0.27, 3.23 and 1.96, respectively).

Physiological signals

For each 1-minute video skin conductance and blood volume (BV) were recorded from the hand at 512 Hz. Heart beats are detected from the BV signal and RR intervals calculated. Heart rate is calculated from the RR signal over a 5-second sliding window (see Figure4.8). The additional signals recorded are not used in this thesis to keep a set of similar signals in both real datasets.

Ad-hoc features

To extract features from SC and HR, the same features extractors defined for the previous dataset are applied (see Section 4.2.4). For blood volume, in addition to all the features extractors applied to BVP in MB, the following feature extractors are defined: maximum $\max\{BV\}$, minimum $\min\{BV\}$, the difference between maximum and minimum signal recording $D^{BV} = \max\{BV\} - \min\{BV\}$, time when maximum BV occurred $t_{\max}\{BV\}$, time when minimum BV occurred $t_{\min}\{BV\}$ and the difference $D_t^{BV} = t_{\max}\{BV\} - t_{\min}\{BV\}$; autocorrelation (lag equals 1) of the signal ρ_1^{BV} initial, BV_{in}, and last, BV_{last}, BV recording, the difference between initial and final BV recording and Pearson's correlation coefficient R_{BV} between raw BV recordings and the time t at which data were recorded.

4.4 Summary

This chapter introduced the datasets used to test the affect modelling methodology presented in this dissertation. Specifically, the procedure for generating synthetic datasets for evaluating preference learning methods (Chapter 5) was detailed. Additionally, two user studies designed to collect the psycho-physiological changes of users watching videos and playing games, respectively, were outlined. Data collection conforms the first phase of the methodology for modeling affect presented in this thesis. The next chapter jumps into the last phase, namely preference modeling, as it will be used in later chapters to evaluate the remaining intermediate phases.

Chapter 5

Modeling Preferences

Modeling is the last phase of the methodology proposed in this thesis. It is used to find the mapping between a set of input features and an affective state. The mapping or model is learned from a set of data samples or objects (input feature values) recorded during affective experiences and annotations of the states felt during those experiences. This chapter focuses on methods that learn the mapping from affective preferences, i.e. annotations that sort the objects by the intensity of the affective state felt.

In particular, we introduce an empirical evaluation of several training algorithms for artificial neural networks, support vector machines and Cohen's method, in order to investigate which method is more suited for affect modeling. We first test these algorithms on synthetic datasets that let us recreate a variety of problems often seen in affect datasets. Specifically, we create 9 datasets by combining 3 mapping complexities with 3 input feature distributions. All datasets contain pairwise preferences which are extracted from a fixed mapping or utility function (i.e. the relation between input features and affect intensity); the three functions used are a linear function $U^1(\mathbf{x})$, a quadratic function $U^2(\mathbf{x})$ and a non-linear function based on a 2-layer ANN $U^{ANN}(\mathbf{x})$. Additionally, the sets of objects are sampled from distributions that simulate properties typically present in affect data. The first set of objects $S_{\mathcal{U}_{-1}^{+1}}$ is sampled from a uniform distribution which represents an ideal scenario serving as a baseline. The second set $S_{\mathcal{N}_{\mu}^{0.1}}$ is sampled from several normal distributions with different means; this results in objects grouped in clusters, and pairwise preferences defined only between objects of the same cluster. This set resembles datasets that aggregate groups of objects as, for instance, datasets with physiological data from different users. The third set $S^d_{\mathcal{U}_{-1}^{+1}}$ samples two groups of objects from a uniform distribution. The pairs of objects in one group feature large differences between the value of the utility function for the preferred and non-preferred object of the pair; this resembles comparisons between experiences that elicit very different affective states (or intensities of the same affective state). The second group of objects within $S^d_{\mathcal{U}_{-1}^{+1}}$ present low utility differences which, in turn, resembles *unclear preferences* in which the affective state felt in the compared experiences is very similar (these datasets are defined in more detail in Chapter 4). In addition to experiments in synthetic data, a first glimpse at affect datasets is also presented by testing the algorithms in a selection of the affect datasets presented in Chapter 4: Maze-Ball and DEAP.

The significance of the results reported in this chapter is evaluated using t-test, and only differences that yield p-values below 0.05 are considered significant. Each of the following sections presents the results on synthetic and real-datasets of one of the computational

methods examined.

5.1 Experiments with Artificial Neural Networks

The experiments analyzed in this section focus on the efficacy of a variety of error functions. ANNs present many other relevant hyper-parameters (e.g. topology), but those are common to other tasks (classification and regression). On the other hand, the error function is the key that enables training from preferences. The error functions explored induce different biases to the trained ANNs as they define differently how the continuous output of the ANN ($U^{\mathbf{w}}(\mathbf{x})$) relates to the ordinal target values (pairwise preferences). Specifically, three of the error functions examined (rank-margin E_{RM}, regularized least-squares E_{RLS} and cross-entropy E_C; see Chapter 3) show a strong dependency with respect to the difference between the model's output for the preferred and non-preferred objects of each pair (denoted as $U^{\mathbf{w}}_{PN}$) while the remaining three (Spearman E_{SP}, sigmoidal E_S and sigmoidal rank-margin E_{SRM}; see Chapter 3) are measures more related to the number of correctly classified pairs — which is connected to the sign of $U^{\mathbf{w}}_{PN}$ (a positive value corresponds to a correctly classified pair) but not its magnitude.

In addition to the error functions, we also examine two training algorithms, namely backpropagation and neuroevolution, in order to assess interactions between the error functions and different optimization strategies. Finally, we also include in the analysis the activation function for the output neurons. For every ANN topology tested, the activation function of the hidden neurons is a logistic sigmoid; however, the output neuron employs either a linear activation function or a logistic sigmoid function (referred to as linear and logistic topologies, respectively). While the choice between linear and logistic does not strongly affect the expressivity of a preference model, it is expected to have a large effect on the training process as the logistic function is bounded and the linear is not. Note that the expressivity is not affected because both activations are monotonic strictly increasing functions; this property implies that both induce the same order when applied to the same set of data samples (i.e. the same network with a linear or logistic output predicts the same pairwise preferences). On the other hand, the training process is affected as the error functions investigated depend on the magnitude of the output in a different manner (to be more precise, they depend on $U^{\mathbf{w}}_{PN}$) which is, in turn, affected by the output activation. In particular, the linear and logistic functions in the output neuron determine a different range of values for $U^{\mathbf{w}}_{PN}$: a linear activation function creates ANNs with unbounded outputs which leads to an unbounded difference between outputs ($U^{\mathbf{w}}_{PN} \in (-\infty, \infty)$) whereas the logistic sigmoid function constricts the output of the networks to the interval $[0, 1]$ which, in turn, yields bounded output differences ($U^{\mathbf{w}}_{PN} \in [-1, +1]$). As each error function integrates $U^{\mathbf{w}}_{PN}$ in a distinct manner, the selection of the activation function presents dissimilar effects across experiments.

In summary, we explore the dependencies expected among training algorithm, error function and output activation across several datasets of varying characteristics. Section 5.1.1 presents experiments on synthetic datasets aimed at analyzing these dependencies. Additionally, a parameter named *margin* was introduced into every error function to regulate the impact of $U^{\mathbf{w}}_{PN}$ on the overall error; Section 5.1.2 presents experiments assessing the effect of this parameter in the training process. Finally, an evaluation of the different methods and variants on affect datasets is introduced in Section 5.1.3.

In all experiments presented in this section, a number of hyper-parameters are adjusted

systematically for every topology, training algorithm, error function and margin value. In particular, the regularizer parameter (for BP and NE), learning rate (for BP only) and the number of parents and chromosomes copied to the next population (for NE only) are tuned. The rest of the parameters were fixed after preliminary experiments. For every experiment, we report the average and standard error of the accuracy on unseen data of 10 models; for synthetic datasets the accuracy is calculated as the percentage of correctly classified pairs in the testing set (20% of the pairs) and for the affect dataset as the average percentage of correctly classified pairs in 3-fold cross-validation.

5.1.1 Error Functions

In the following, an evaluation of the error functions with their original margin values is presented for the synthetic datasets generated with each of the three utility functions. In particular, the value of the margin is set to 1.0 for E_{RM} and E_{RLS} and to 0.0 for the remaining error functions.

Linear synthetic data

Single-layer perceptrons are trained to learn the three synthetic datasets generated via a linear utility function. As the features in these datasets are linearly related to the synthetic preferences, an SLP suffices to approximate the utility function. As it can be observed from Figure 5.1, both training algorithms learn successfully the target utilities despite the input noise and different data distributions. However, several error functions show significant accuracy decrements despite the simplicity of learning a linear function.

In the uniformly distributed set ($U^1S_{\mathcal{U}_{-1}^{+1}}$), both training algorithms and topologies across error functions yield models with accuracies above 94% (see Figure 5.1a and Figure 5.1b). These accuracies match the accuracy achieved by the target utility function $U^1(\mathbf{x})$ as noise was introduced in the creation of the dataset. This suggest that the training methods examined can learn linear functions with a high precision despite certain level of input noise.

Backpropagation presents the only noticeable decrement in accuracy when training linear SLPs with E_S. Note that some experiments (out of 10 trials) yield accuracies below 94% as reflected in the relatively large standard error. This suggests, that for this error function the initial weights of the SLP, even for a simple linear function, may have a significant impact on the performance of gradient descent.

When the objects in the dataset are sampled from clusters ($U^1S_{\mathcal{N}_{\mu}^{0.1}}$), NE trains linear SLPs that maintain accuracies around 94% independently of the error function employed (Figure 5.1c). On the other hand, solely E_{SP} and E_{SRM} yield logistic SLPs with similar accuracies (93.86% and 93.83%, respectively) as seen in Figure 5.1d. The significant decrement in accuracy for the other error functions can be explained by their stronger dependency on $U_{PN}^{\mathbf{w}}$. This strong dependency implies that the training algorithm reduces the error by increasing $U_{PN}^{\mathbf{w}}$ for most pairs in the dataset; however, when increasing $U_{PN}^{\mathbf{w}}$ with a logistic network for pairs within particular clusters, differences in other clusters shrink. This reduction in particular clusters appears as the outputs of the network in that region of the input space become very similar because the logistic output of the network saturates (outputs values nearly 1 or

nearly 0). Thus, the training algorithm with these error functions trades off correctly classified pairs (in clusters that saturate the ANN's output) for higher output pair differences (in clusters that lie on the linear regime of the logistic function). On the other hand, the objectives imposed by E_{SP} (maximizing the number of correctly classified pairs) and E_{SRM} (minimizing $U^{\mathbf{w}}_{PN}$ among incorrectly classified pairs) regard the number of incorrectly classified pairs uniformly across clusters, independently of which regime of the logistic function the lie on.

When backpropagation is applied to the linear clustered dataset, we observe a larger number of accuracy decrements with respect to results in the uniformly-distributed dataset. For linear topologies, only E_C and E_{RM} present small (yet significant) performance drops which appears to be caused by the training process stopping before all pairs are classified correctly; the reason for this early stop is that the contribution to the gradient from correctly classified pairs nullify the contributions of incorrectly classified pairs towards the end of the training process; E_{RLS}, E_{SRM} and E_S reduce this effect by generating a larger difference on the gradient between correctly and incorrectly classified pairs (giving more emphasis to the latter).

For logistic SLPs, backpropagation does not train models with accuracies above 92% with any of the error functions while E_{SRM} yields accuracies that are even lower. This is because the gradient of this error function (which tries to reduce $U^{\mathbf{w}}_{PN}$ among incorrectly classified pairs) combined with the regularizer used by backpropagation (which tries to reduce $\mathbf{w}$) drives training towards minimal weight configurations (i.e. $|\mathbf{w}| \approx 0$); that generate, through the logistic activation, outputs close to 0.5 for every object in the dataset (effectively reducing $U^{\mathbf{w}}_{PN}$ for every pair). As the gradient does not strongly promote positive values of $U^{\mathbf{w}}_{PN}$, trained SLPs present suboptimal training, validation and testing accuracies. In comparison, NE paired with E_{SRM} trains SLPs with larger weights and higher prediction accuracies, in part, due to the global-search nature of the training process and, in part, due to the apparent inefficacy of the regularizer term.

Altogether, these decreases in accuracy with respect to the uniform dataset highlight the complexity inherent to learning from pairs located in different and compact areas of the input space.

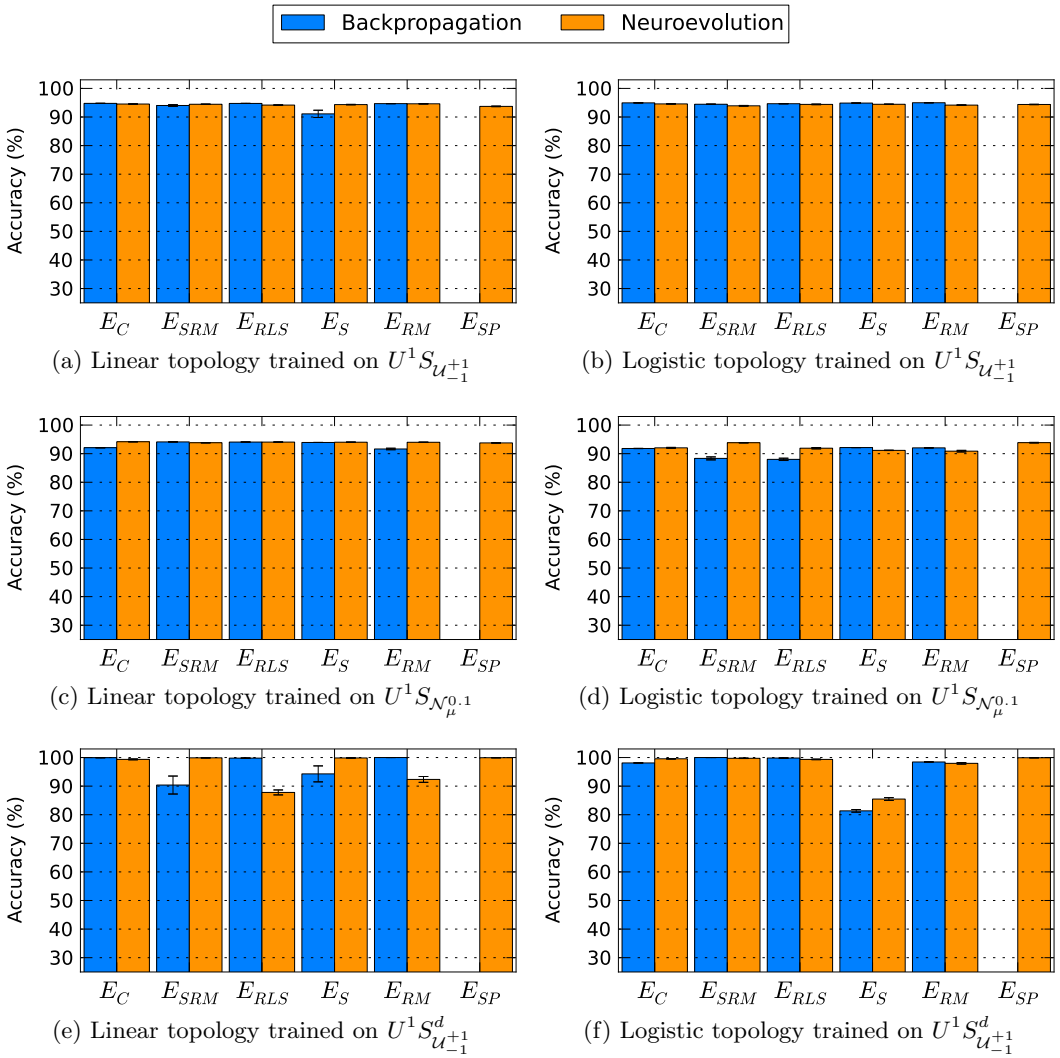
In the dataset with synthetic unclear preferences ($U^1S^d_{\mathcal{U}^{+1}_{-1}}$), several SLPs reach 100% accuracy (see Figure 5.1e and Figure 5.1f). This is possible as none of the 5% pairs with inverted preferences fall within the testing partition for this dataset. These results confirm that the methods investigated can train SLPs that are able to generalize well to unseen data despite the noise in the training set. The accuracy drops seen in some error functions are partly motivated by the difficulty of learning the preferences linked to small utility differences (i.e. the unclear preferences). For linear SLPs, NE presents lower accuracies with E_{RLS} and E_{RM} (87.8% and 92.35%, respectively). These accuracy decrements appear to come in hand with their strong dependence on $U^{\mathbf{w}}_{PN}$: as the SLPs in the NE population come closer to the target utility, clear preference pairs yield large $U^{\mathbf{w}}_{PN}$ values and unclear preference pairs yield small $U^{\mathbf{w}}_{PN}$ values. In consequence, error differences across these SLPs are dominated by pairs sampled with large utility differences because these pairs contribute to the error function with larger $U^{\mathbf{w}}_{PN}$ changes. Consequently, NE trains SLPs that classify correctly clear pairs but does not necessarily unclear pairs. Similarly, one could expect

the same behavior from E_C as it also depends strongly on $U^{\mathbf{w}}_{PN}$; however, it appears to perform well with NE. This could be motivated by the smaller difference between the contribution to the error from large and small $U^{\mathbf{w}}_{PN}$ values (i.e. from clear and unclear pairs), as determined by a less steep slope (see Figure 3.8).

BP does not show problems with those error functions but instead yields accuracy drops for E_{SRM} and E_S (90.38% and 94.29%, respectively). Both error functions show a large standard error as consequence of the training algorithm finding optimal solutions in several runs. This shows that both error functions can lead to optimal solutions but inappropriate initial weights can yield to a local optimum. The better results for E_{RLS} and E_{RM} appear to be due to the more effective use of the regularizer, that can neutralize the gradient before the weights grow too large (giving larger differences).

Logistic SLPs trained for $U^1 S^d_{\mathcal{U}^{+1}_{-1}}$ — independently of the error function or training algorithm — converge to networks with large weights that produce saturated outputs for most objects in the dataset. On one hand, $U^{\mathbf{w}}_{PN}$ is maximal for pairs sampled with large utility differences because the output equals 1 for preferred objects and equals 0 for non-preferred objects. On the other hand, $U^{\mathbf{w}}_{PN}$ is close to zero for most pairs sampled with small utility differences because the output for both objects in the pair lie on the same flat regime of the logistic sigmoid. These saturated SLPs yield high accuracies across all error functions and training algorithms with the exception of E_S which yields surprisingly low accuracies (Figure 5.1f). In E_S, when dealing with $U^{\mathbf{w}}_{PN}$ values close to zero, increasing $U^{\mathbf{w}}_{PN}$ for few correctly classified pairs has a larger impact on the error than reducing $U^{\mathbf{w}}_{PN}$ for many incorrectly classified pairs. Consequently, the trained SLPs only classify correctly a small subset of the pairs sampled with small utility differences. This effect is not seen on the other error functions that depend strongly on $U^{\mathbf{w}}_{PN}$ because they present a smaller change around the boundary defined by $U^{\mathbf{w}}_{PN} = 0$.

Overall, both training algorithms are able to train SLPs that approximate the linear utility function to a good degree despite the noise added to the inputs. Even though the target utility function is simple, different object distributions have an effect on training. Spearman and sigmoidal rank-margin combined with neuroevolution always yield models with accuracies among the highest; results suggests that the independence of these error functions with respect to the difference between the trained model's output for preferred and non-preferred objects ($U^{\mathbf{w}}_{PN}$) among correctly classified pairs, allows NE to always find optimal solutions independently of the activation function used in the output neuron. For BP, regularized least-squares stands out as the most robust function. It appears that this function defines the gradient that best balances the effect of correctly and incorrectly classified pairs during training.



(a) Linear topology trained on $U^1S_{\mathcal{U}_{-1}^{+1}}$

(b) Logistic topology trained on $U^1S_{\mathcal{U}_{-1}^{+1}}$

(c) Linear topology trained on $U^1S_{\mathcal{N}_{\mu}^{0.1}}$

(d) Logistic topology trained on $U^1S_{\mathcal{N}_{\mu}^{0.1}}$

(e) Linear topology trained on $U^1S^d_{\mathcal{U}_{-1}^{+1}}$

(f) Logistic topology trained on $U^1S^d_{\mathcal{U}_{-1}^{+1}}$

Figure 5.1: Single-layer perceptrons trained on the synthetic linear datasets with uniformly distributed objects ($U^1S_{\mathcal{U}_{-1}^{+1}}$), clustered objects ($U^1S_{\mathcal{N}_{\mu}^{0.1}}$) and differentiated groups of utility differences ($U^1S^d_{\mathcal{U}_{-1}^{+1}}$). Bars represent the average accuracy of 10 SLPs in the testing data partition while the error bars represent the standard error.

Quadratic synthetic data

Two multi-layer topologies with one hidden layer are trained to learn the three synthetic datasets generated via a quadratic utility function. The first topology (MLP^2) contains two hidden neurons which are enough to predict with perfect accuracy pairwise preferences based on a quadratic utility; the second topology (MLP^{10}) contains 10 hidden neurons, the increased complexity can facilitate the training process by enabling several optimal solutions but can also be more prone to overfitting. Observing Figure 5.2 and Figure 5.3 it is clear that NE outperforms BP across all datasets and error functions examined, yielding the best results with E_{SRM} and E_{SP}.

In the uniformly distributed set ($U^2S_{\mathcal{U}_{-1}^{+1}}$), NE trains MLP^2 networks with linear output activation achieving testing accuracies above 80% with all error functions and over 90% with several of them as shown in Figure 5.2a. With the larger topology (MLP^{10}) and also linear output, E_{SRM} and E_{SP} present a decreased accuracy pointing out the insufficient number of generations (or number of individuals in the population) to train a topology of this size (Figure 5.3a). In the previous section, results suggested that these error functions lead to optimal solutions when error functions that depend on $U_{PN}^{\mathbf{w}}$ can not; however, they also generate a more *abrupt fitness landscape* that may slow down training, i.e. the error between solutions that are similar may change rapidly (due to the discontinuity on the error function) or not at all (due to the flat regions on the error function) requiring more generation to find the global optimum. For MLP^{10}, the best results are achieved with E_C and E_{RM} that, arguably, generate the smoothest landscape of all functions as they present no discontinuities, and they are monotonic with small slopes (unlike E_S).

When NE is used to train logistic MLPs on the uniformly distributed quadratic dataset, only E_{SRM}, E_{SP} and E_S maintain accuracies similar to the linear MLPs. It appears that the added complexity arising from bounding the maximization objective ($U_{PN}^{\mathbf{w}}$) results in poor training performances. While E_S still depends on the maximization of $U_{PN}^{\mathbf{w}}$ among correctly classified pairs, this dependency fades away after a small difference is reached.

The most striking result in the uniform dataset, is the poor performance of BP that yields accuracies above 70% only when training MLP^{10} with E_{RLS} (78.96% accuracy on average; see Figure 5.3a). It is worth mentioning, that attempts to train SLPs for this dataset yield, on average, accuracies around 68% which match the accuracies reached by BP with most error functions when training MLP^2 and MLP^{10}. This suggests that in most experiments gradient descent managed to learn, at best, a monotonic approximation of this utility function.

In the quadratic dataset with clustered pairs ($U^2S_{\mathcal{N}_{\mu}^{0.1}}$), E_S, E_{SRM} and specially E_{SP} (89.41% with the logistic activation) combined with NE yield the most accurate MLP^2, independently of the output activation function (see Figure 5.2c and Figure 5.2d). The other error functions train (in some experiments) linear MLP^2 that yield accuracies below chance level (as reflected by the low average and corresponding the high standard error). This is caused by a convergence of the genetic population towards monotonic functions that only maximize $U_{PN}^{\mathbf{w}}$ within the clusters of pairs that are located in one side of the parabola — determined by the synthetic quadratic

utility $U^2(\mathbf{w})$. When the logistic activation is used, the average accuracies drop below chance level, showing that only these monotonic solutions are found.

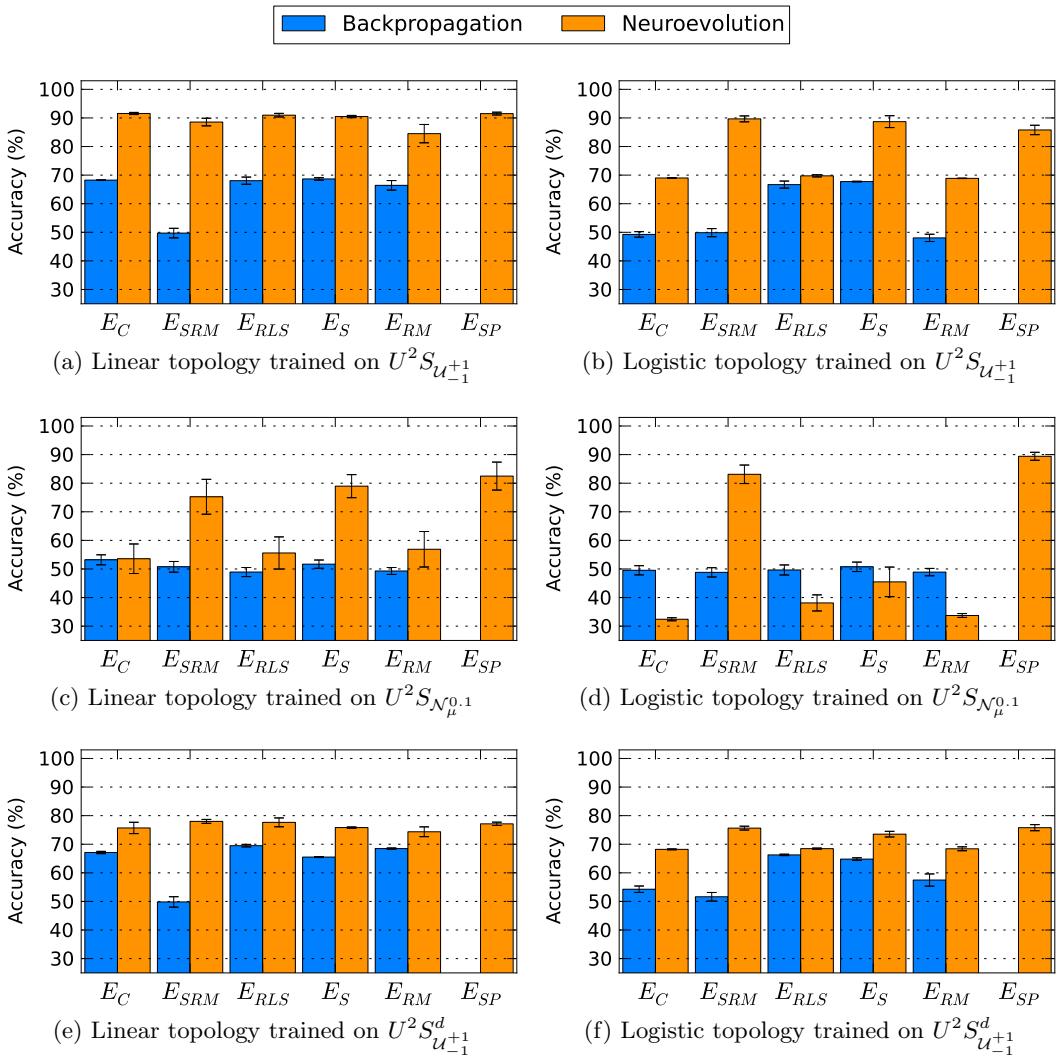
Experiments with MLP^{10} and NE confirm the result found on the uniform quadratic dataset, that shows that the performance of error functions that do not dependent on the value of $U^{\mathbf{w}}_{PN}$ among correctly classified pairs decreases when applied to larger networks (see Figure 5.3c and Figure 5.3d).

Backpropagation yields only baseline accuracies (around 50%) due to the inability of learning the training examples (training accuracies are also around 50%). It appears that the small differences between the clustered objects in the training pairs result in very low error gradients in the hidden neurons unable to modify the initial random weights significantly. Possibly, a variable learning rate (larger for the hidden neurons than the output neurons) could improve the performance of BP in clustered sets.

In the dataset with two groups of objects with differentiated utility differences ($U^2S^d_{\mathcal{U}^{+1}_{-1}}$), none of the training methods get, on average, accuracies above 80%, which amounts to a decrease of 10% in accuracy compared to the best results in the uniformly distributed set (see Figure 5.2e and Figure 5.3e). Most of the incorrectly classified pairs were sampled with short utility differences (unclear preferences), but unlike the results seen in the experiments with the linear utility, a number of pairs with large utility differences (clear preferences) are also not correctly classified. This finding suggests that a dataset with a significant amount of unclear preferences, even when they are correct (note that 50% of the pairs in the synthetic set contains unclear preferences but only 5% are incorrect), can drive training towards suboptimal solutions.

Finally, BP performs worse than NE and accuracies only reach 70% when training MLP^{10} with E_{RLS}, similarly to the experiments in the uniformly distributed set (but with a 10% decrement in the best accuracy).

In sum, learning the quadratic utility is, unexpectedly, a more difficult task compared to learning the linear utility. Given a small ANN topology, NE learned the function up to a good degree from the uniform and clustered distributions (best accuracies around 90%) using E_{SRM} and E_{SP} (and E_S for linear output activations only); the distribution with groups of differentiated utility differences posed a greater challenge as the synthetic unclear preferences appear to stir the training algorithm towards suboptimal solutions. Large topologies, on the other hand, could not be learned accurately with E_{SRM} and E_{SP}, presumably due to the relatively abrupt fitness landscape generated. The alternative training algorithm, backpropagation did not achieve as high accuracies in any of the datasets despite training on the same topologies. It appears that for the quadratic utility function, gradient decent with the proposed error functions always converges to suboptimal solutions with accuracies similar to monotonic functions (SLP). Only E_{RLS} outperformed those accuracies when applied for training the larger topology. In the clustered set backpropagation underperformed due to the proximity of the training objects within each pair, which produced a small gradient unable to train the logistic hidden neurons.



(a) Linear topology trained on $U^2S_{\mathcal{U}_{-1}^{+1}}$

(b) Logistic topology trained on $U^2S_{\mathcal{U}_{-1}^{+1}}$

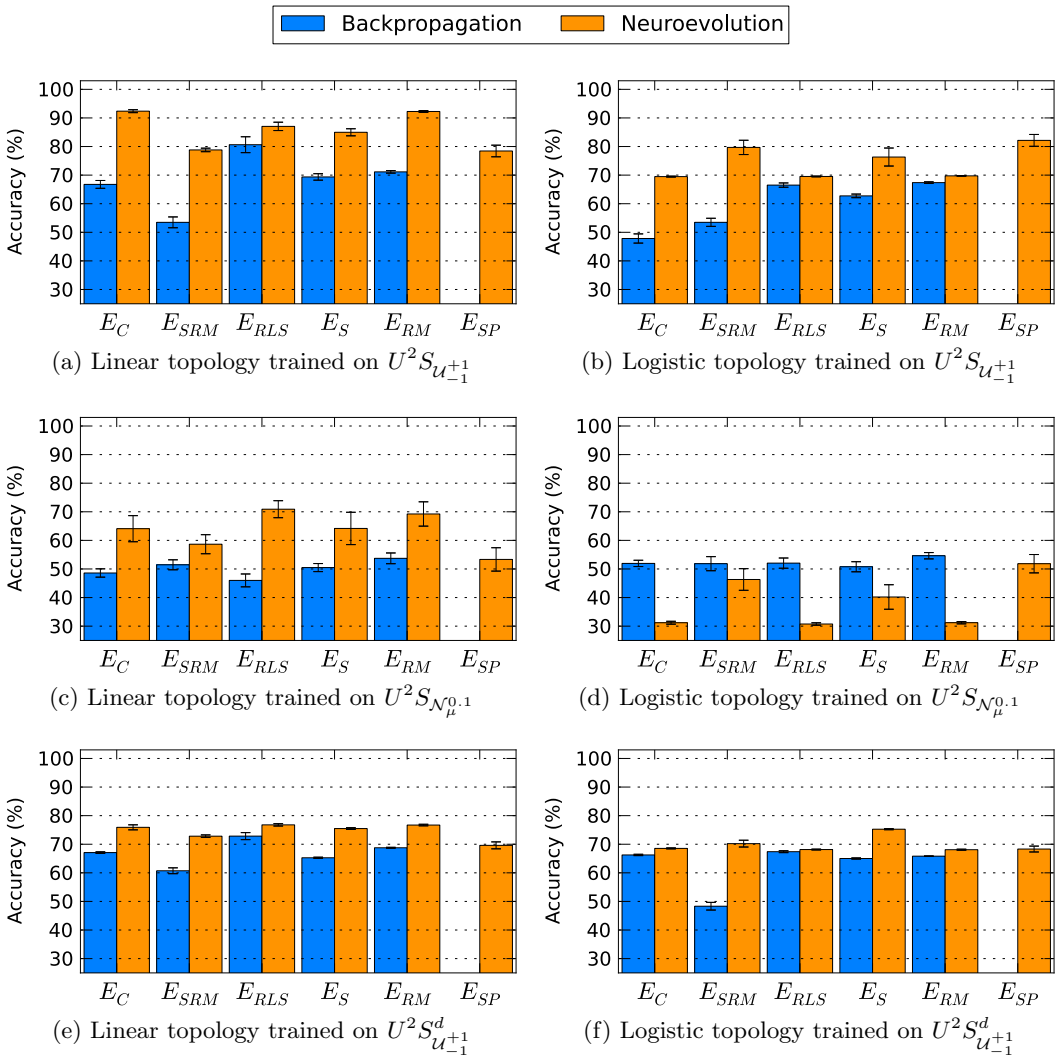
(c) Linear topology trained on $U^2S_{\mathcal{N}_{\mu}^{0.1}}$

(d) Logistic topology trained on $U^2S_{\mathcal{N}_{\mu}^{0.1}}$

(e) Linear topology trained on $U^2S^d_{\mathcal{U}_{-1}^{+1}}$

(f) Logistic topology trained on $U^2S^d_{\mathcal{U}_{-1}^{+1}}$

Figure 5.2: Multi-layer perceptrons with 2 hidden neurons (MLP^2) trained on the synthetic quadratic datasets with uniformly distributed objects ($U^2S_{\mathcal{U}_{-1}^{+1}}$), clustered objects ($U^2S_{\mathcal{N}_{\mu}^{0.1}}$) and differentiated groups of utility differences ($U^2S^d_{\mathcal{U}_{-1}^{+1}}$). Bars represent the average accuracy of 10 MLPs in the testing data partition while the error bars represent the standard error.



(a) Linear topology trained on $U^2S_{\mathcal{U}_{-1}^{+1}}$

(b) Logistic topology trained on $U^2S_{\mathcal{U}_{-1}^{+1}}$

(c) Linear topology trained on $U^2S_{\mathcal{N}_{\mu}^{0.1}}$

(d) Logistic topology trained on $U^2S_{\mathcal{N}_{\mu}^{0.1}}$

(e) Linear topology trained on $U^2S^d_{\mathcal{U}_{-1}^{+1}}$

(f) Logistic topology trained on $U^2S^d_{\mathcal{U}_{-1}^{+1}}$

Figure 5.3: Multi-layer perceptrons with 10 hidden neurons (MLP^{10}) trained on the synthetic quadratic datasets with uniformly distributed objects ($U^2S_{\mathcal{U}_{-1}^{+1}}$), clustered objects ($U^2S_{\mathcal{N}_{\mu}^{0.1}}$) and differentiated groups of utility differences ($U^2S^d_{\mathcal{U}_{-1}^{+1}}$). Bars represent the average accuracy of 10 MLPs in the testing data partition while the error bars represent the standard error.

Neural synthetic data

A two-hidden-layer MLP (MLP^5_{10}) with 5 and 10 neurons, respectively, is trained to learn the three synthetic datasets generated via a non-linear utility function. This utility function is a neural network with the exact same topology as MLP^5_{10}, thus the function can potentially be learned with perfect accuracy (around 95% due to the induced noise in the data).

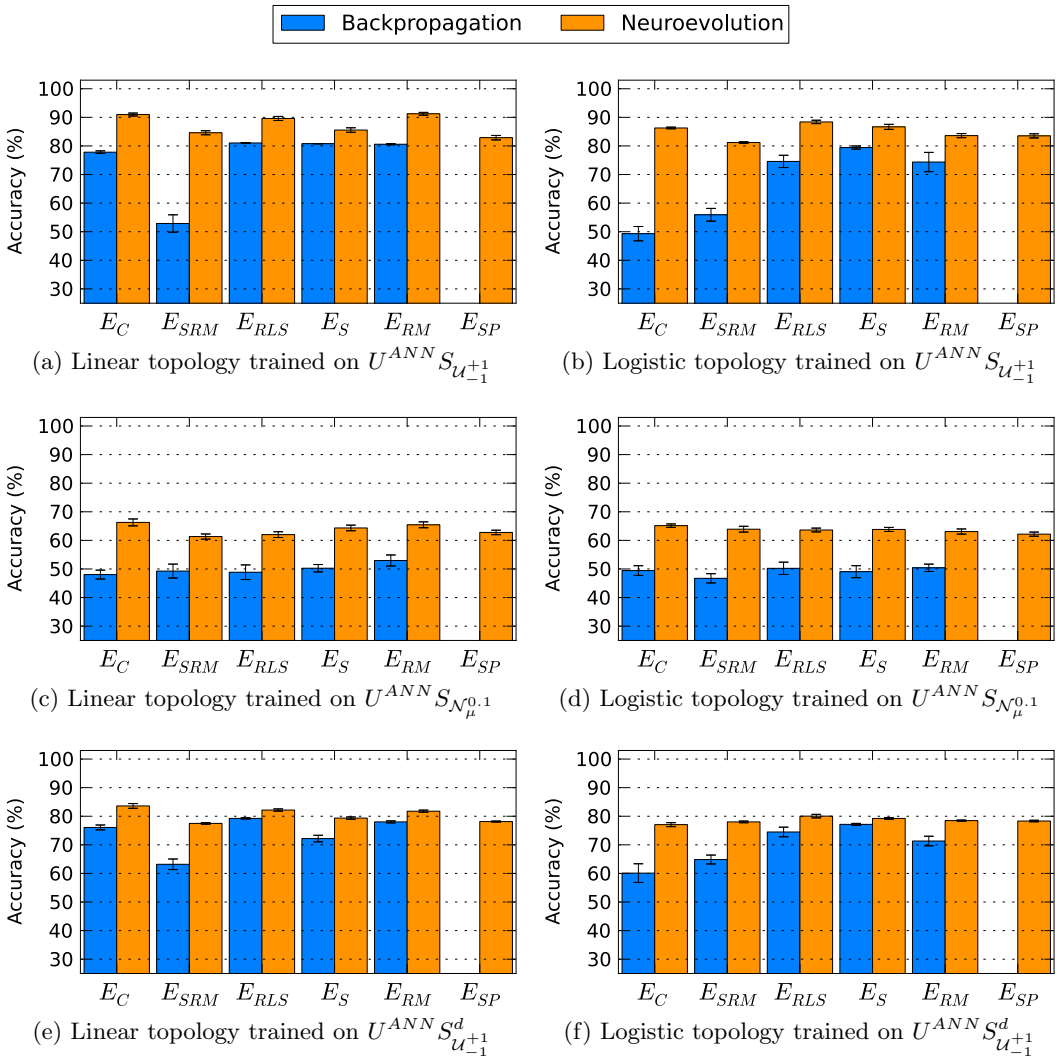
In the uniformly distributed dataset ($U^{ANN}S_{\mathcal{U}^{+1}_{-1}}$), NE yields again more accurate models than BP (see Figure 5.4a and Figure 5.4b) with maximum accuracies of 90.98% and 91.24% for MLPs with linear activation output trained using E_C and E_{RM}, respectively. E_{SRM}, E_S and E_{SP} yield slightly (but significantly) lower accuracies; given the relatively large size of the topology, this is consistent with the findings reported for the quadratic dataset.

BP reaches its highest accuracy with E_{RLS} training an MLP with linear output activation (81.04% accuracy); while this is not low, additional tests revealed that SLPs can achieve similar accuracies in this dataset, which suggests that gradient descent could not capture adequately the non-linearity of the utility. The low accuracies presented with E_{SRM} confirm results found in the other datasets showing the inability of this error function to drive gradient descend.

In the clustered dataset ($U^{ANN}S_{\mathcal{N}^{0.1}_{\mu}}$), testing accuracies are severely decreased (see Figure 5.4c and Figure 5.4d). NE trains models with testing accuracies above 60% for all error functions, with the highest average accuracy presented by E_C (66.29% accuracy); SLPs trained for this dataset present similar accuracies further showing that the utility was not appropriately learned. These low accuracies are in line with previous results that show that, on one hand, error functions that depend strongly on $U^{\mathbf{w}}_{PN}$ among correctly classified pairs find difficulties in clustered datasets and, on the other hand, the other error functions suffer performance decrements on large topologies. Both issues converge in this particular dataset, leaving no adequate error function to learn this non-linear synthetic utility from clustered data. As expected from previous results, BP does not improve baseline accuracies due to the proximity between training objects. This experiment further shows the difficulty of learning a complex non-linear function from scattered and clustered data pairs rather than uniformly sampled pairs.

The dataset with short and long utility differences ($U^{ANN}S^{d}_{\mathcal{U}^{+1}_{-1}}$) is modeled more accurately with linear output activation models when trained with NE and the error functions E_C, E_{RLS} and E_{RM} (83.61%, 82.16% and 81.76%, respectively) and BP with the error function E_{RLS} (79.24%) as seen in Figure 5.4f and Figure 5.4e. The drop in accuracy with respect to the uniform distributed set is lower than the one observed in experiments with quadratic utility datasets; nevertheless, the synthetic unclear preferences did not aid the learning task.

In sum, the experiments on synthetic neural datasets did not reveal new insights but confirmed some of the results found on the other synthetic datasets.



(a) Linear topology trained on $U^{ANN}S_{\mathcal{U}_{-1}^{+1}}$

(b) Logistic topology trained on $U^{ANN}S_{\mathcal{U}_{-1}^{+1}}$

(c) Linear topology trained on $U^{ANN}S_{\mathcal{N}_{\mu}^{0.1}}$

(d) Logistic topology trained on $U^{ANN}S_{\mathcal{N}_{\mu}^{0.1}}$

(e) Linear topology trained on $U^{ANN}S^{d}_{\mathcal{U}_{-1}^{+1}}$

(f) Logistic topology trained on $U^{ANN}S^{d}_{\mathcal{U}_{-1}^{+1}}$

Figure 5.4: Multi-layer perceptrons with two hidden layers with 5 and 10 hidden neurons (MLP^{5}_{10}) trained on the synthetic neural datasets with uniformly distributed objects ($U^{ANN}S_{\mathcal{U}_{-1}^{+1}}$), clustered objects ($U^{ANN}S_{\mathcal{N}_{\mu}^{0.1}}$) and differentiated groups of utility differences ($U^{ANN}S^{d}_{\mathcal{U}_{-1}^{+1}}$). Bars represent the average accuracy of 10 MLPs in the testing data partition while the error bars represent the standard error.

From the results of all datasets, we can conclude that NE trains more accurate ANNs than BP across synthetic datasets as BP encounters difficulties learning non-linear utilities. Backpropagation achieves the highest accuracies when combined with error functions that present a strong dependency with respect to $U^{\mathbf{w}}_{PN}$ — i.e. the difference between the trained model's output for the preferred and non preferred object — among correctly and incorrectly classified pairs (these are E_{RM}, E_{RLS} and E_C). E_{RLS} in particular stands out as in most experiments yields higher prediction accuracies; this error function seems to generate the gradient that best balances the pull towards correcting the outputs for incorrectly classified pairs and increasing $U^{\mathbf{w}}_{PN}$ for correctly classified pairs. On the other hand, NE achieves higher accuracies through the error functions that dependent mostly on the number of correctly classified pairs and $U^{\mathbf{w}}_{PN}$ only among incorrectly classified pairs (these are E_{SP} and E_{SRM}). This alternative strategy appears superior specially when there exist training pairs with small differences in synthetic utility (as with unclear preferences or clustered data). On the other hand, these error function are not suitable for backpropagation: E_{SP} creates a null gradient that cannot inform training and E_{SRM} creates a gradient that drives training towards models that do not separate the outputs between preferred and non-preferred objects. In addition, it appears that the performance of NE with these error functions decreases as the size of the model trained grows; it is unclear from the experiments reported in this chapter, whether by investing on a larger number of training generations and/or population size, these functions could always find optimal solutions, but results do not indicate otherwise.

It was also shown, that for datasets that feature objects grouped in compact clusters, BP cannot train logistic hidden neurons as consequence of the generated small gradient which may be corrected with the use of an enlarged learning rate for those layers. Additionally, the use of a logistic output activation did not seem to provide any significant benefits and, instead, created a more difficult optimization problem for error functions strongly dependent on $U^{\mathbf{w}}_{PN}$.

Finally, following the lower accuracies present in non-uniform datasets, it appears that learning a preference model from user reports can be facilitated by minimizing the number of unclear preferences and by including pairs of data expanded across the whole input space, rather than grouped into clusters. In the domain of affect modeling, annotation protocols such as the 4-alternative forced choice (see more details in Section 3.1.1) can be of great help with the first issue as users can explicitly indicate unclear preferences. The second issue, which may easily appear in physiological affect datasets when aggregating data from different users, activities or sessions, does not present a clear solution; while within-subject or within-session normalization may benefit learning, one must acknowledge that the preference function learned may variate with the normalization scheme applied, as the preference reports from different users (or sessions from the same user) will be overlapped at different relative positions in model's input space.

5.1.2 Margin tuning

The analysis presented in the previous section shows that not all error functions are appropriate for specific problems, topologies and training algorithms. As it was pointed out, the difference between the output of a model for the two objects in a training pair (denoted as $U^{\mathbf{w}}_{PN}$), is a fundamental element of the error functions examined and it has an important impact on the learning process. The margin parameter defined for all error functions mediates the relation between $U^{\mathbf{w}}_{PN}$ and the overall error, thus it is expected to have also a decisive

role in the training process. This section summarizes the outcome of tuning the value of the margin for each of the error functions across the synthetic datasets and topologies presented in the previous section. Four key values are selected for the margin: 0.0 and 1.0 as they are the *de facto* values used in the error functions, 0.5 for being the middle point and 0.01 representing a non-zero low value.

Regularized least-squares E_{RLS}

In this error function, the margin represents the desired value for $U^{\mathbf{w}}_{PN}$ across training pairs — its the local minima of a quadratic parabola so as $U^{\mathbf{w}}_{PN}$ increases or decreases its value from the margin value, the error raises quadratically. Non surprisingly, a margin equal to zero is not compatible to this error function as it trains a model that does not differentiate between compared objects (see Figure 5.5). In most experiments, 1.0 yields the highest accuracy with the exception of networks with a logistic output activation trained with neuroevolution. In that specific setting, 0.01 yields equally good or higher accuracies than other margins in the experiments with non-linear synthetic utilities. Note that with a margin of 1.0, that the only logistic networks that can reach the error global minimum must output 1.0 for all preferred objects and 0.0 for all non-preferred; as the margin is lowered, the minimum no longer requires extreme output values resulting in a smoother fitness landscape that facilitates training.

Backpropagation in the non-linear datasets requires a margin of at least 0.5 in order to achieve accuracies higher than the baseline which suggests that 0.01 generates a gradient too small for effective training at the critical region around $U^{\mathbf{w}}_{PN} = 0$ — i.e. the boundary that defines if a pair is correctly classified or not.

Rank-margin E_{RM}

While in E_{RLS} the margin specifies the exact desired value for $U^{\mathbf{w}}_{PN}$, in this error function it specifies the minimum desired value for $U^{\mathbf{w}}_{PN}$. This subtle difference generates better results for margins that equal zero but as in E_{RLS}, generally larger values of the margin are associated with higher accuracies — excluding the case of 0.01 for logistic topologies trained with NE (see Figure 5.6). Note that for zero margin, this function becomes more similar to E_{SRM} but its performance when combined with NE is much worse; this shows that the step in E_{SRM}, which stresses the difference between correctly and incorrectly classified, is essential for global-search training. On the other hand, this step is not relevant for BP because it does not affect the gradient — which is equal to zero with or without the step.

Sigmoid E_S

For this error function the margin is not the target value for $U^{\mathbf{w}}_{PN}$ as in E_{RLS} and E_{RM}; instead it defines the value around which $U^{\mathbf{w}}_{PN}$ variations have a significant impact on the error. The larger this value is, the greater the impact of changing $U^{\mathbf{w}}_{PN}$ among correctly classified pairs. As seen in Figure 5.7, lower values of the margin (0.0 and 0.01) tend to yield better results across datasets and training algorithms although occasionally larger values yield more accurate models (e.g. in the linear set with differentiated utility differences $S_{\mathcal{U}^{+1}_{-1}}U^{ANN}$). Large margins of this function appear to give not enough emphasis to incorrectly classified pairs which reflects negatively on the training performance. The original margin used along with this function is 0.0; however, this experiment shows that

increasing the margin to 0.01 can yield slight performance improvements which motivates for the inclusion of the margin as an additional hyper-parameter.

Sigmoidal rank-margin E_{SRM}

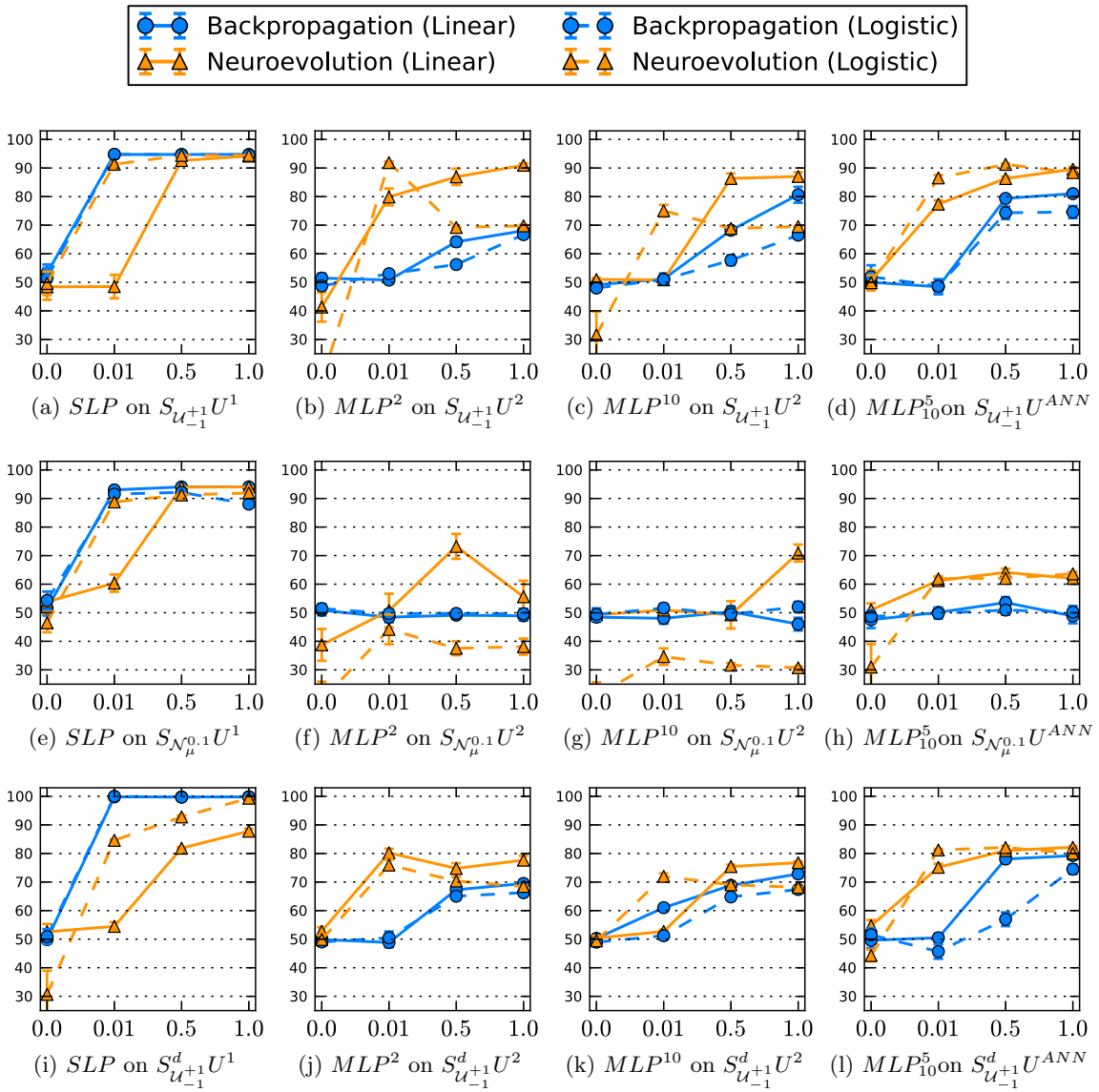
This error function was designed as a combination of E_S and E_{RM}; thus, the margin specifies the minimum desired $U^{\mathbf{w}}_{PN}$ (the error is constant after the value specified by the margin is reached) and only variations of $U^{\mathbf{w}}_{PN}$ within a close range of values around the margin have a significant impact on the overall error. For BP, a margin equal to 0.01 yields the highest accuracies in most datasets (see Figure 5.8). This small margin promotes differences slightly above zero, enough to avoid convergence towards ANNs with minimal weights that cannot separate training pairs. For NE, small values also yield better results but the differentiation between 0.0 and 0.01 is not as large given that global-search training can find optimal solutions with zero margin.

Cross-entropy E_C

This is a monotonic function of $U^{\mathbf{w}}_{PN}$ that presents a decreasing slope as the models' output between preferred and non-preferred increases. By increasing the margin, the error for every training pair increases, but this increment is larger for incorrectly classified pairs. Adjusting the value of the margin produced significant changes for several experiments, as higher values tend to provide higher accuracies (see Figure 5.9). It appears that for this error function, systematic tuning of the margin could yield better models as a general best value was not found.

Spearman E_{SP}

The margin for this simple function specifies the minimum $U^{\mathbf{w}}_{PN}$ required to consider a pair as correctly classified. For logistic models, a margin value of 1.0 yields baseline results as no model can be trained to surpass a difference of 1.0 (see Figure 5.10). For both linear and logistic models, it appears that a margin of zero value yields the most accurate models although 0.01 can lead to slightly more accurate models in some settings. Even though it could be expected that larger values yield higher accuracies as a result of a better generalization, the global-search nature of the genetic algorithm finds decent solutions with the minimum possible margin.

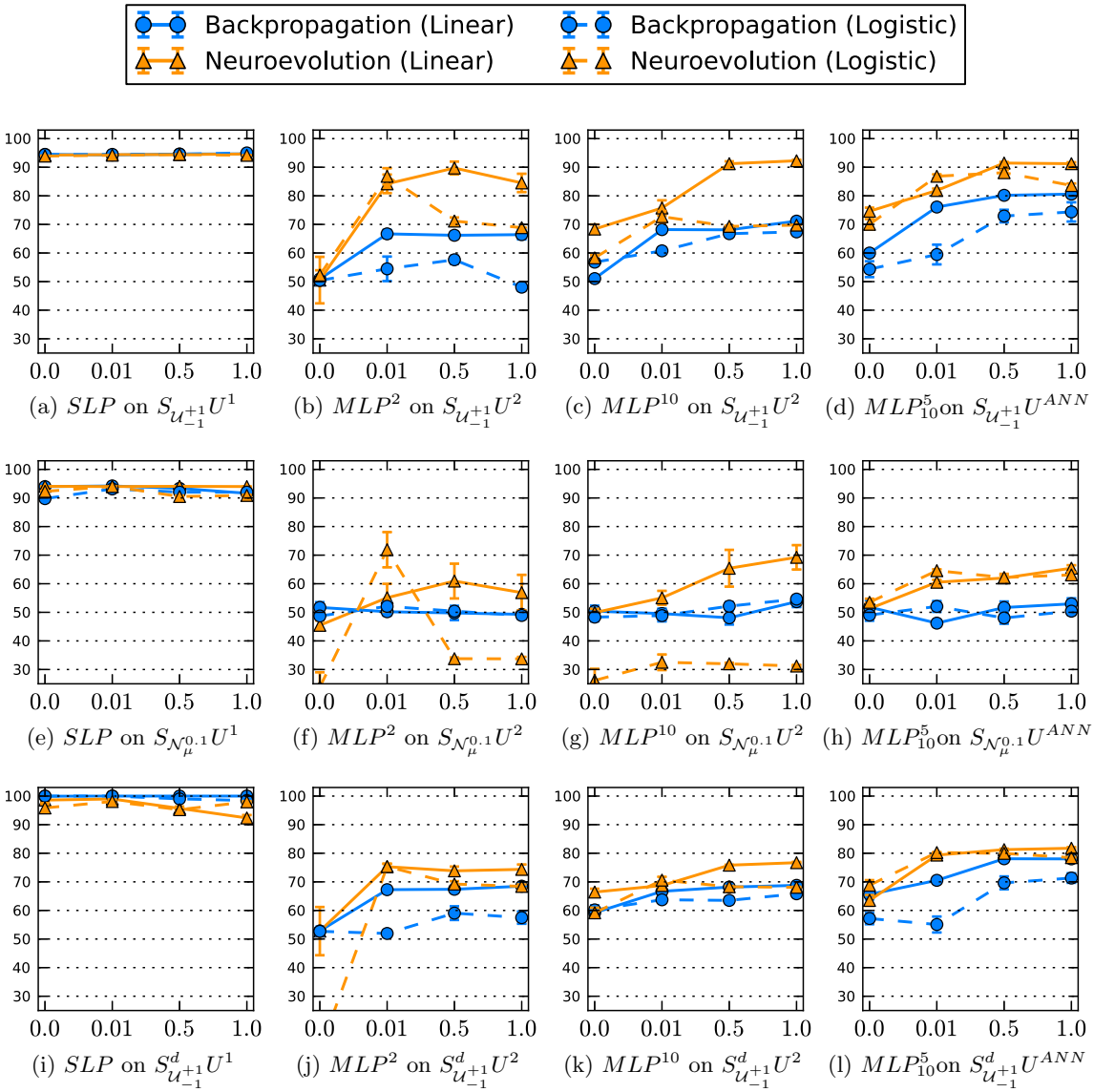


(a) SLP on $S_{\mathcal{U}_{-1}^{+1}U^1}$ (b) MLP^2 on $S_{\mathcal{U}_{-1}^{+1}U^2}$ (c) MLP^{10} on $S_{\mathcal{U}_{-1}^{+1}U^2}$ (d) MLP_{10}^5on $S_{\mathcal{U}_{-1}^{+1}U^{ANN}}$

(e) SLP on $S_{\mathcal{N}_{\mu}^{0.1}U^1}$ (f) MLP^2 on $S_{\mathcal{N}_{\mu}^{0.1}U^2}$ (g) MLP^{10} on $S_{\mathcal{N}_{\mu}^{0.1}U^2}$ (h) MLP_{10}^5on $S_{\mathcal{N}_{\mu}^{0.1}U^{ANN}}$

(i) SLP on $S^d_{\mathcal{U}_{-1}^{+1}U^1}$ (j) MLP^2 on $S^d_{\mathcal{U}_{-1}^{+1}U^2}$ (k) MLP^{10} on $S^d_{\mathcal{U}_{-1}^{+1}U^2}$ (l) MLP_{10}^5on $S^d_{\mathcal{U}_{-1}^{+1}U^{ANN}}$

Figure 5.5: Regularized least-squares: the average accuracy of 10 models trained using neuroevolution and backpropagation with E_{RLS} is tested on 4 margin values (0.0, 0.01, 0.5 and 1.0) for each of the synthetic datasets and selected topologies. The standard error is depicted as error bars.

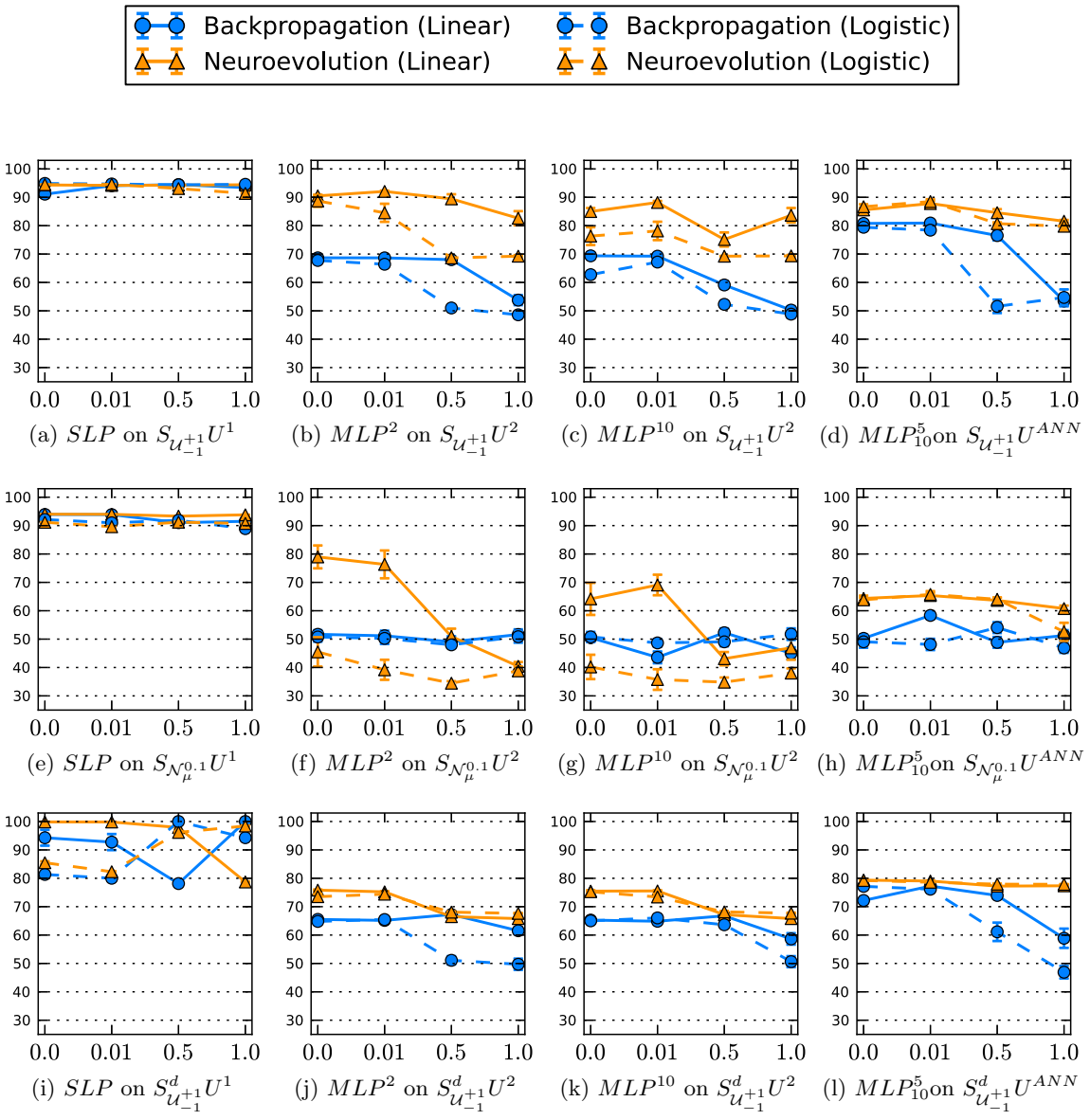


(a) SLP on $S_{\mathcal{U}_{-1}^{+1}}U^1$ (b) MLP^2 on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (c) MLP^{10} on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (d) MLP_{10}^5on $S_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

(e) SLP on $S_{\mathcal{N}_{\mu}^{0.1}}U^1$ (f) MLP^2 on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (g) MLP^{10} on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (h) MLP_{10}^5on $S_{\mathcal{N}_{\mu}^{0.1}}U^{ANN}$

(i) SLP on $S^d_{\mathcal{U}_{-1}^{+1}}U^1$ (j) MLP^2 on $S^d_{\mathcal{U}_{-1}^{+1}}U^2$ (k) MLP^{10} on $S^d_{\mathcal{U}_{-1}^{+1}}U^2$ (l) MLP_{10}^5on $S^d_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

Figure 5.6: Rank-margin: the average accuracy of 10 models trained using neuroevolution and backpropagation with E_{RM} is tested on 4 margin values (0.0, 0.01, 0.5 and 1.0) for each of the synthetic datasets and selected topologies. The standard error is depicted as error bars.

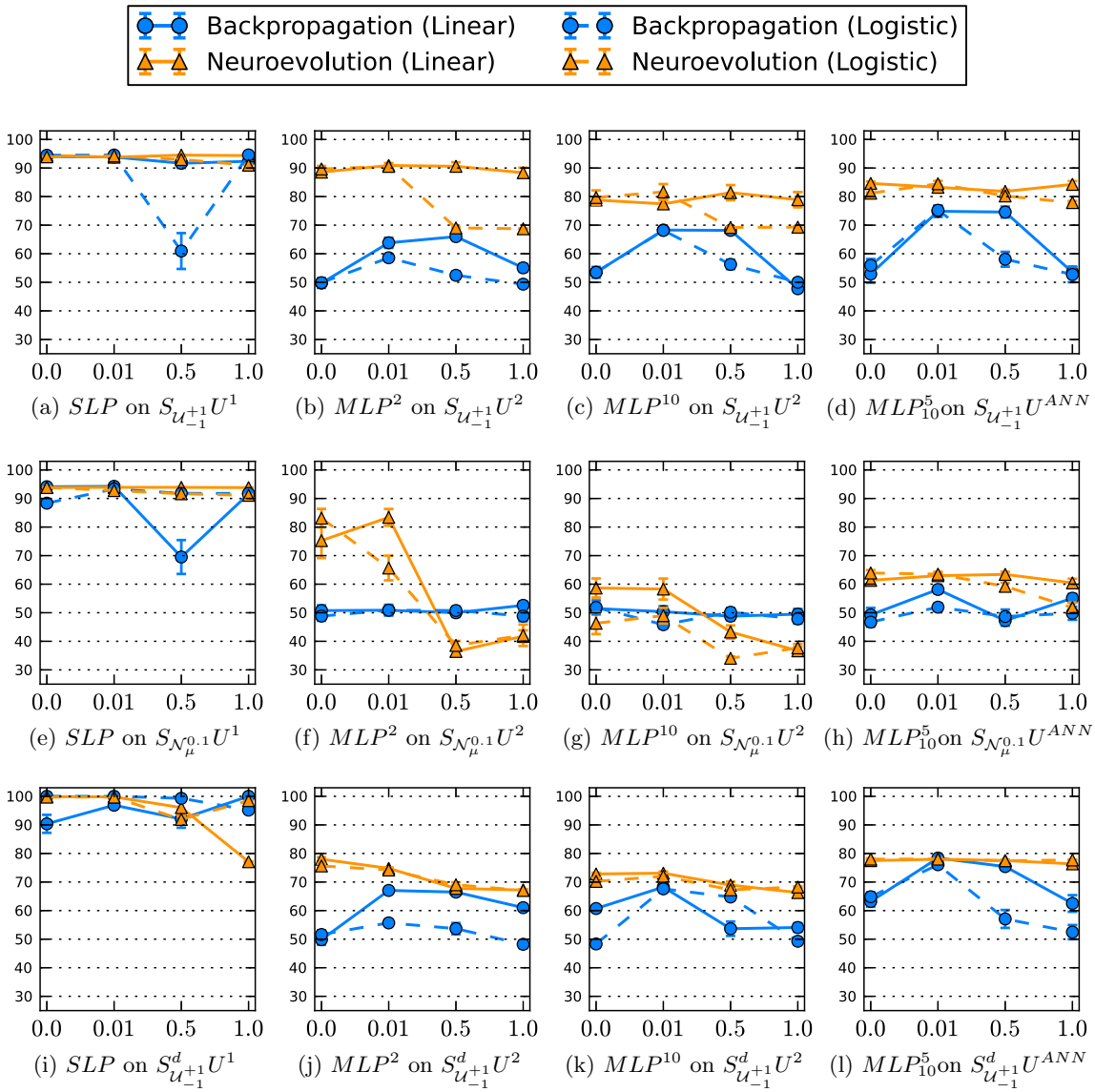


(a) SLP on $S_{\mathcal{U}_{-1}^{+1}}U^1$ (b) MLP^2 on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (c) MLP^{10} on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (d) MLP_{10}^5on $S_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

(e) SLP on $S_{\mathcal{N}_{\mu}^{0.1}}U^1$ (f) MLP^2 on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (g) MLP^{10} on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (h) MLP_{10}^5on $S_{\mathcal{N}_{\mu}^{0.1}}U^{ANN}$

(i) SLP on $S^d_{\mathcal{U}_{-1}^{+1}}U^1$ (j) MLP^2 on $S^d_{\mathcal{U}_{-1}^{+1}}U^2$ (k) MLP^{10} on $S^d_{\mathcal{U}_{-1}^{+1}}U^2$ (l) MLP_{10}^5on $S^d_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

Figure 5.7: Sigmoid: the average accuracy of 10 models trained using neuroevolution and backpropagation with E_S is tested on 4 margin values (0.0, 0.01, 0.5 and 1.0) for each of the synthetic datasets and selected topologies. The standard error is depicted as error bars.

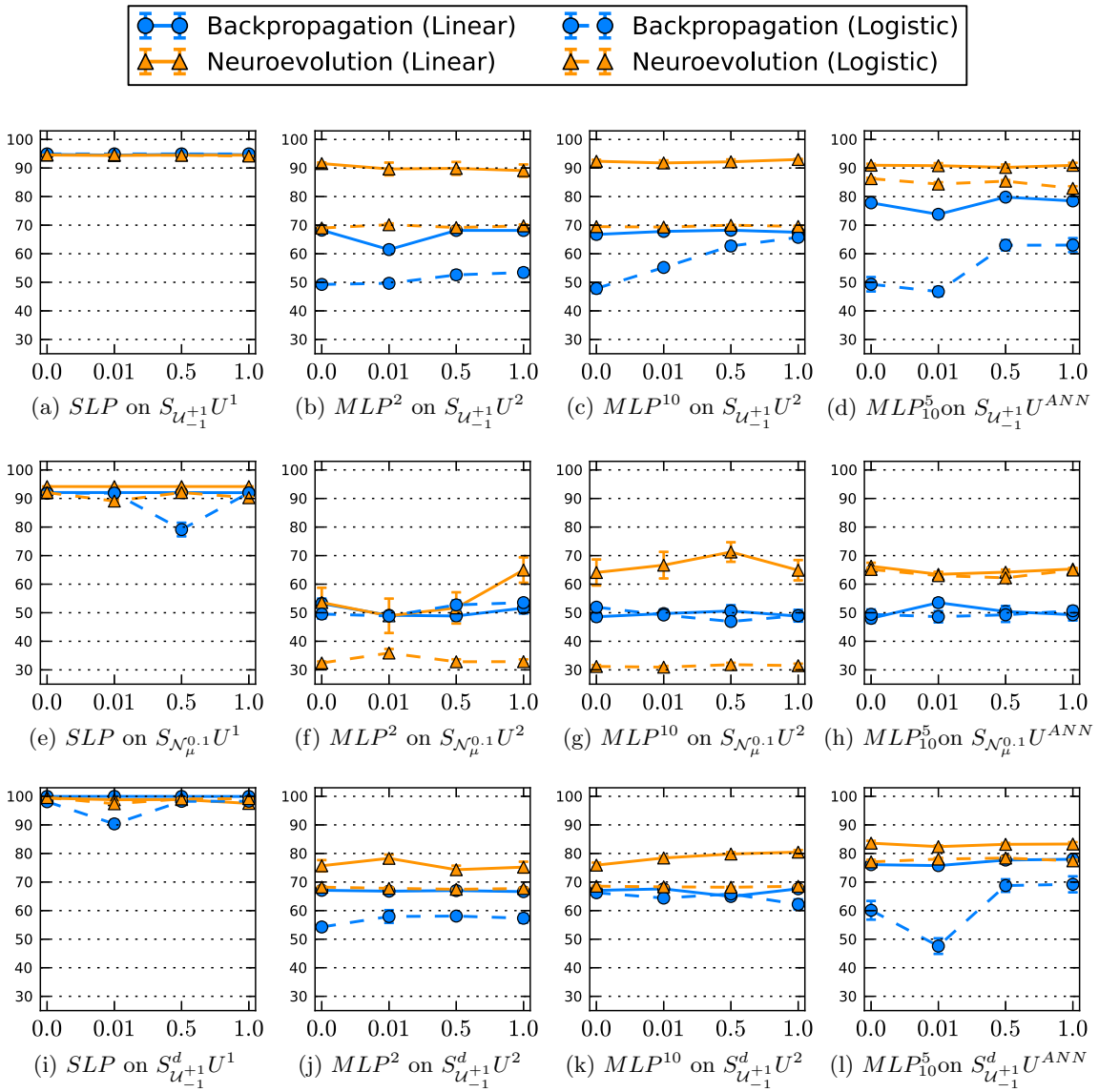


(a) SLP on $S_{\mathcal{U}_{-1}^{+1}}U^1$ (b) MLP^2 on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (c) MLP^{10} on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (d) MLP_{10}^5on $S_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

(e) SLP on $S_{\mathcal{N}_{\mu}^{0.1}}U^1$ (f) MLP^2 on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (g) MLP^{10} on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (h) MLP_{10}^5on $S_{\mathcal{N}_{\mu}^{0.1}}U^{ANN}$

(i) SLP on $S_{\mathcal{U}_{-1}^{+1}}^d U^1$ (j) MLP^2 on $S_{\mathcal{U}_{-1}^{+1}}^d U^2$ (k) MLP^{10} on $S_{\mathcal{U}_{-1}^{+1}}^d U^2$ (l) MLP_{10}^5on $S_{\mathcal{U}_{-1}^{+1}}^d U^{ANN}$

Figure 5.8: Sigmoidal rank-margin: the average accuracy of 10 models trained using neuroevolution and backpropagation with E_{SRM} is tested on 4 margin values (0.0, 0.01, 0.5 and 1.0) for each of the synthetic datasets and selected topologies. The standard error is depicted as error bars.

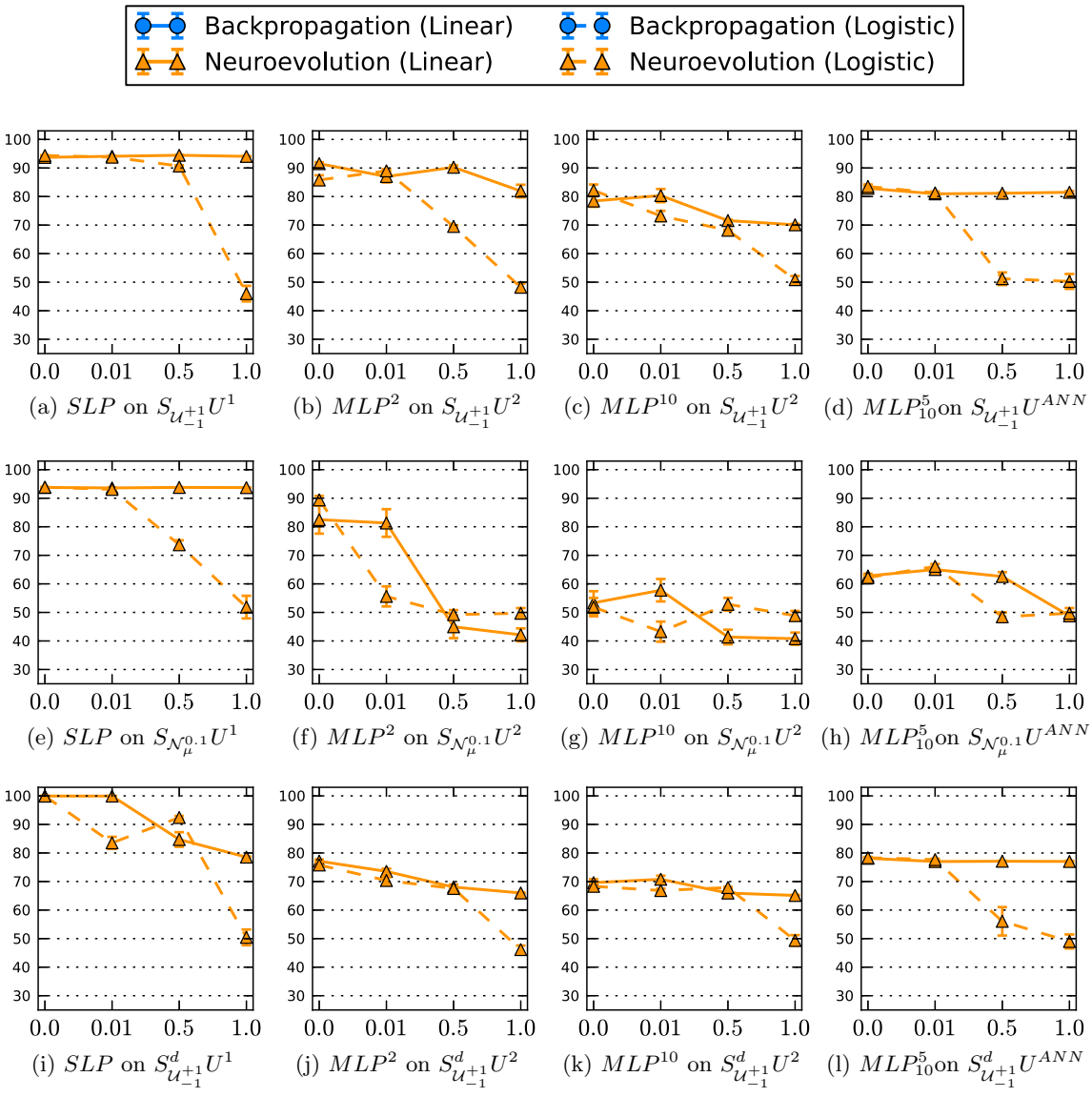


(a) SLP on $S_{\mathcal{U}_{-1}^{+1}}U^1$ (b) MLP^2 on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (c) MLP^{10} on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (d) MLP_{10}^5on $S_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

(e) SLP on $S_{\mathcal{N}_{\mu}^{0.1}}U^1$ (f) MLP^2 on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (g) MLP^{10} on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (h) MLP_{10}^5on $S_{\mathcal{N}_{\mu}^{0.1}}U^{ANN}$

(i) SLP on $S^d_{\mathcal{U}_{-1}^{+1}}U^1$ (j) MLP^2 on $S^d_{\mathcal{U}_{-1}^{+1}}U^2$ (k) MLP^{10} on $S^d_{\mathcal{U}_{-1}^{+1}}U^2$ (l) MLP_{10}^5on $S^d_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

Figure 5.9: Cross-entropy: the average accuracy of 10 models trained using neuroevolution and backpropagation with E_C is tested on 4 margin values (0.0, 0.01, 0.5 and 1.0) for each of the synthetic datasets and selected topologies. The standard error is depicted as error bars.



(a) SLP on $S_{\mathcal{U}_{-1}^{+1}}U^1$ (b) MLP^2 on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (c) MLP^{10} on $S_{\mathcal{U}_{-1}^{+1}}U^2$ (d) MLP^5_{10}on $S_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

(e) SLP on $S_{\mathcal{N}_{\mu}^{0.1}}U^1$ (f) MLP^2 on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (g) MLP^{10} on $S_{\mathcal{N}_{\mu}^{0.1}}U^2$ (h) MLP^5_{10}on $S_{\mathcal{N}_{\mu}^{0.1}}U^{ANN}$

(i) SLP on $S^d_{\mathcal{U}_{-1}^{+1}}U^1$ (j) MLP^2 on $S^d_{\mathcal{U}_{-1}^{+1}}U^2$ (k) MLP^{10} on $S^d_{\mathcal{U}_{-1}^{+1}}U^2$ (l) MLP^5_{10}on $S^d_{\mathcal{U}_{-1}^{+1}}U^{ANN}$

Figure 5.10: Spearman: the average accuracy of 10 models trained using neuroevolution with E_{SP} is tested on 4 margin values (0.0, 0.01, 0.5 and 1.0) for each of the synthetic datasets and selected topologies. The standard error is depicted as error bars.

5.1.3 Explorations on affect datasets

In addition to the analysis of the performance of ANN techniques on synthetic pairwise data, we describe in this section an initial analysis of their performance on real affect datasets where the utility function underlying the preferences is unknown. The different error functions, margin values, topologies and training algorithms explored for synthetic data are applied to learn models from Maze-Ball and DEAP. A subset of ad-hoc physiological features is selected by hand for each dataset; each feature is normalized using z-transformation — i.e. subtracting the mean and dividing by the standard deviation — over the complete sets (global normalization) and over individual participants (within-subject normalization). Unlike the experiments with synthetic data, the differences in accuracy across different parameter configurations were small and often statistically insignificant, therefore we only present, for each affective and cognitive state reported in each dataset and for the two normalization schemes, the configurations that yielded the highest average accuracy over 10 runs (see Table 5.1 and Table 5.2).

The first noticeable result is the rather low average accuracies obtained. These are expected, however, due to the lack of context features (e.g. gameplay features in MB) and the fixed subset of features (ANN inputs) used across reported states. As it is shown in Chapter 6 and Chapter 7, a careful feature selection mechanism is critical to obtain accurate models when the dataset used is small.

For globally normalized features, the highest accuracies for *boredom*, *excitement* and *frustration* are achieved via backpropagation of single-layer perceptrons while the highest accuracies for *anxiety*, *relaxation* and *fun* are achieved with multi-layer perceptrons (1 hidden layer with 10 neurons) trained via neuroevolution. Cross-entropy (with a margin value of zero), rank-margin (with a margin value of 0.5 for a linear SLP and of 0.01 for a logistic SLP) and regularized least-squares (with a margin value of 1.0) are the error functions. Reports of *challenge* in MB and the three reported states in DEAP (*arousal*, *valence* and *liking*) are predicted with low accuracies given the manually selected features. For within-subject normalized MB data, backpropagation with rank-margin ($m = 1.0$ or $m = 0.5$) or sigmoidal rank-margin ($m = 0.01$) yield the highest accuracies with the exception of *challenge* which finds the best model through the combination of the sigmoid error function ($m = 0.5$) and neuroevolution. DEAP reports are also predicted poorly with this normalization and only SLPs trained for *arousal* (backpropagation combined with E_C) present an average accuracy above baseline (57.05%).

At first, it may be surprising that BP yields better models than NE given that in the synthetic datasets the genetic search beats the gradient search in most experiments. It seems that, the higher training accuracies obtained by NE translate often in lower validation accuracies in these small affect datasets. On the other hand, for the globally normalized datasets NE trains MLPs that are significantly better than any other model trained by BP (for *fun* and *relaxation*). Although here only the best results are presented — they are not necessarily significantly more accurate than the second best — it appears that rank-margin, sigmoidal rank-margin and cross-entropy can train models that generalize better to unseen data than models trained with other error functions.

Data	Target	Training	Error	Margin	Topology	Activation	Accuracy
MB	Anxiety	NE	RLS	1.0	MLP^{10}	Linear	58.17 (1.07)
	Boredom	BP	C	0.0	SLP	Linear	66.67 (<0.01)
	Challenge	NE	SRM	1.0	SLP	Linear	54.85 (0.87)
	Excitement	BP	RM	0.01	SLP	Logistic	64.99 (<0.01)
	Fun	NE	C	0.0	MLP^{10}	Linear	64.44 (1.39)
	Frustration	BP	RM	0.5	SLP	Linear	67.33 (0.36)
	Relaxation	NE	C	0.0	MLP^{10}	Linear	60.45 (1.36)
DEAP	Arousal	NE	C	0.5	MLP^{5}_{10}	Linear	56.20 (0.42)
	Valence	NE	SRM	0.5	SLP	Linear	55.39 (0.34)
	Like	BP	C	0.0	MLP^{10}	Linear	55.63 (0.24)

Table 5.1: ANN configurations for globally normalized datasets: experiment configuration that yielded the highest validation accuracy (average and standard error in parenthesis of the 3-fold cross-validation accuracy from 10 independent runs) for each dataset. The configuration is defined by the training algorithm, error function, margin value, network topology and activation function of the last neuron.

Data	Target	Training	Error	Margin	Topology	Activation	Accuracy
MB	Anxiety	BP	SRM	0.01	SLP	Logistic	61.07 (0.7)
	Boredom	BP	RM	1.0	SLP	Linear	66.67 (0.37)
	Challenge	NE	S	0.5	SLP	Logistic	58.98 (0.25)
	Excitement	BP	SRM	0.01	MLP^{10}	Linear	65.24 (0.29)
	Fun	BP	SRM	0.01	MLP^{2}	Linear	66.42 (0.0)
	Frustration	BP	RM	1.0	SLP	Linear	61.00 (0.19)
	Relaxation	BP	RM	0.5	SLP	Linear	60.89 (0.26)
DEAP	Arousal	BP	C	0.01	SLP	Linear	57.05 (0.26)
	Valence	BP	RLS	0.01	SLP	Logistic	55.77 (<0.01)
	Like	BP	RM	1.0	SLP	Linear	55.33 (0.04)

Table 5.2: ANN configurations for within-subject normalized datasets: experiment configuration that yielded the highest validation accuracy (average and standard error in parenthesis of the 3-fold cross-validation accuracy from 10 independent runs) for each dataset. The configuration is defined by the training algorithm, error function, margin value, network topology and activation function of the last neuron.

5.2 Experiments with Support Vector Machines and Cohen's Method

In order to frame the performance of artificial neural networks, we also examine support vector machines and Cohen's method which represent, respectively, a widely popular machine learning algorithm and a method tailored for preference learning. This section compares the accuracies of these two methods over the same synthetic and real datasets explored on the previous section. Both training algorithms are deterministic, hence the accuracy of one run is shown. For Cohen's method the number of epochs and learning rate are adjusted systematically for each dataset. Five SVM kernels are tested (linear κ^1, sigmoid κ^S, Gaussian κ^G and polynomial of second κ^2 and third κ^3 degree) and the regularizer parameter C

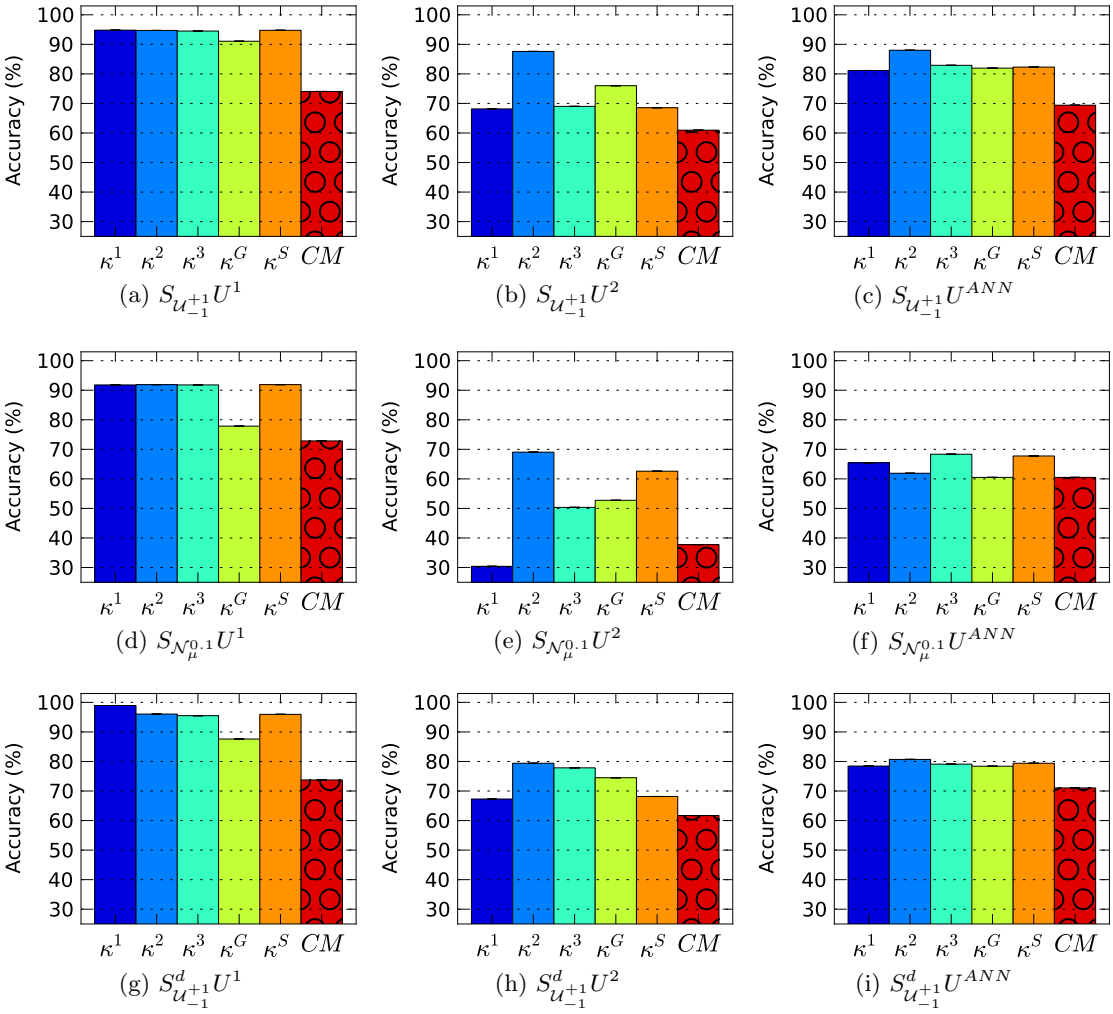


Figure 5.11: Support vector machines and Cohen's method: the testing accuracy of SVMs with different kernels — linear (κ^1), polynomial of 2^{nd} degree (κ^2) and 3^{rd} degree (κ^3), sigmoid (κ^s) and Gaussian (κ^G) — and Cohen's method (CM) are presented for each of the synthetic datasets.

and the kernel-dependent parameters γ and β are adjusted systematically for each kernel and dataset. As described in Section 3.4.2, the kernel defines the complexity of the function that the SVM can approximate, in a similar fashion to the topology on artificial neural networks; thus, different kernels are expected to perform best for different datasets. Figure 5.11 depicts the accuracies of the models trained on the synthetic datasets.

For the three linear datasets, the linear kernel yields equally high or higher accuracies than the other kernels. These accuracies (94.8%, 91.8% and 98.95% for the uniform, clustered and differentiated-utilities datasets, respectively) are fairly high, nevertheless slightly below the accuracies of the best single-layer perceptrons trained on the same datasets. Cohen's model only reaches accuracies around 70%; although the model consists of a linear combination of single-feature comparisons which could approximate these datasets perfectly, the training algorithm converges to maximize only one weight corresponding to the feature with a higher correlation on the training set.

For the three quadratic datasets, as expected, the SVM with polynomial kernel of second degree yields higher accuracies than the other kernels (87.6%, 69.05% and 79.4%). It appears that the same issues that afflict ANNs also generate problems in SVMs as these accuracies are not significantly higher than the best multi-layer perceptrons (with 2 hidden neurons average accuracies of 91.68%, 89.41% and 79.18% were achieved). In particular, the error functions that succeed for neuroevolution in the clustered dataset outperform, by an ample difference, the SVM kernels tested. Cohen's method yields again significantly lower accuracies.

For the datasets generated with a non-linear neural network, SVMs with polynomial kernels yield the highest accuracies. For the uniform and unclear preference datasets, the second degree polynomial kernel yields the highest accuracies (accuracies of 88.0% and 80.7%, respectively) which, as in previous experiments, do not outperform ANN approaches (91.24% and 83.61% highest accuracies among ANNs). On the clustered set, the polynomial kernel of 3^{rd} degree yields the highest accuracy which surpasses the best ANN by a small margin (68.35% and 66.28% for the SVM and ANN, respectively) although the accuracy is still relatively low. Cohen's models yield again lower accuracies than the other methods.

The experiments in synthetic data showcase that ANNs conform a reliable method to solve object ranking tasks that, given the adequate error function and parameters, outperforms a simple approach as Cohen's method and yields comparable — and in some scenarios significantly better — results to SVM, one of the most widespread machine learning methods.

As with ANNs, the experiments with real datasets did not produced large accuracy differences; therefore, we report the accuracies of the models that yielded higher prediction accuracies (see Table 5.3 and Table 5.4). On real datasets with features normalized across all participants, a polynomial kernel of 3^{rd} degree outperforms the neural network models for *relaxation*, *frustration* and *challenge* (see Table 5.3). The neural network training algorithms yield more accurate *fun* and *excitement* models and no differences are observed for *boredom* and *anxiety*. For the DEAP dataset, the sigmoid and Gaussian kernels yield the best results across all methods but close to baseline. With the data normalized within-subject, *challenge*, *excitement* and *fun* are predicted more accurately by SVMs (Gaussian and polynomial of 3^{rd} degree) while the other 4 reported states are predicted more accurately by ANN models (see Table 5.4). For DEAP reports, predictions are still below 60%; however, a polynomial of 3^{rd} degree achieves a 58.58% accuracy; the highest obtained in this dataset with manually selected features. In turn, Cohen's model yield accuracies below 60% for all states and datasets as none of the features is highly correlated with the preferences.

Altogether, the reported experiments show that ANNs can learn more accurately most synthetic datasets and several affect datasets than SVMs. In addition, ANNs present an advantage with respect to interpretation of the learned functions (Bengio, 2007). In brief, both models can be broken down into a linear model that imposes an order over a non-linear projection of the input space; in an ANN this projection is provided by one or more hidden layers that can provide meaningful insights on the nature of the modeled function through a hierarchy of learned features. On the other hand, the non-linear projection of an SVM is defined by a subset of training objects, each of them generating a dimension (or feature) of the projected space through the kernel function; as the parameters of each dimension are selected from the set of training objects (rather than learned), the expressivity of such representation is limited. In addition, since only one layer is created the number of projected features may explode as the complexity of the modeled function increases. An advantage of SVMs is that the training algorithm is deterministic; however, it may come with a larger

Data	Target	Kernel	SVM accuracy	CM accuracy
MB	Anxiety	Polynomial (degree 3)	58.21	46.67
	Boredom	Polynomial (degree 2)	66.67	55.56
	Challenge	Polynomial (degree 3)	61.90	55.71
	Excitement	Polynomial (degree 3)	62.61	58.99
	Frustration	Polynomial (degree 3)	66.67	47.78
	Fun	Polynomial (degree 3)	65.20	59.82
	Relaxation	Polynomial (degree 3)	64.45	56.67
DEAP	Arousal	Sigmoid	57.57	50.93
	Valence	Sigmoid	52.70	56.44
	Like	Gaussian	56.49	48.08

Table 5.3: SVMs and Cohen's models for globally normalized datasets: accuracies of Cohen's models and best SVM kernels.

Data	Target	Kernel	SVM accuracy	CM accuracy
MB	Anxiety	Gaussian	59.31	46.67
	Boredom	Gaussian	64.81	55.56
	Challenge	Polynomial (degree 3)	64.93	55.71
	Excitement	Gaussian	66.09	58.99
	Frustration	Polynomial (degree 2)	59.99	47.78
	Fun	Polynomial (degree 3)	65.34	59.82
	Relaxation	Gaussian	58.89	56.67
DEAP	Arousal	Sigmoid	57.38	51.11
	Valence	Polynomial (degree 3)	58.48	56.44
	Like	Sigmoid	56.31	48.08

Table 5.4: SVMs and Cohen's models for within-subject normalized datasets: accuracies of Cohen's models and best SVM kernels.

computational cost. While a careful analysis of computational effort is not performed, SVMs training time was significantly higher than the corresponding training times for the neural network approaches in the reported experiments. Using a Java implementation for neural networks, with no particular optimization mechanisms, and the *SVM-light* library (written in C) with the default optimization parameters (Joachims, 1999), training one model (ANN or SVM) in a dedicated node required, approximately less than 5 minutes for BP, less than 15 minutes for NE and between 20 minutes and several hours for SVMs (excluding linear kernels that trained in few seconds).

Finally, Cohen's method performs poorly as it trains models that predict preferences based only on the most correlated feature across the dataset. Potentially, an alternative training algorithm — such as a genetic algorithm guided by the Spearman error function — could yield more competitive results. Furthermore, one must note that the comparison over synthetic datasets may have benefited ANNs and SVMs as those methods are based on transitive preferences as the utilities used to create the data; Cohen's method on the other hand is designed to deal with intransitive preferences. Therefore, in highly intransitive datasets, Cohen's method could outperform ANNs and SVMs.

5.3 Summary

In this chapter we evaluated different methods to implement the last phase (preference modeling) of the affect modeling methodology introduced in this thesis. The experiments focused on artificial neural networks and in particular, on error functions and training algorithms. Results on synthetic data suggested the superiority of neuroevolution and highlighted the effects of two properties of the error functions, namely the dependency on the number of correctly classified pairs, and the impact of the difference between the trained model's output for preferred and non-preferred objects in each pair. Neuroevolution generated more accurate models with error functions that primed the first property and disregarded the output differences of pairs classified correctly, these are Spearman and sigmoidal rank-margin error functions. Backpropagation on the other hand performed better with error functions based on the output differences for correctly and incorrectly classified pairs. Specifically, the regularized least-squares error function features the relation between output differences and error that yielded the highest accuracies. Furthermore, we studied the effect of manually adjusting the impact of that difference on the error, enabled through the proposed margin parameter. Results showed that each error function performs better with different margins but in general, it appeared that the performance of ANN training algorithms can be improved by systematically tuning this hyper-parameter.

For a selection of real affect datasets, BP achieved better results in several settings, due to a greater overfitting of NE. As also derived by results in several experiments with the synthetic sets, it seems that the regularizer term — which prevents gradient descent from overfitting — did not achieve its intended purpose with global-search training.

Complementary, the efficacy of ANNs was validated against other popular machine learning techniques (support vector machines) and preference learning-specific methods (Cohen's model). Experiments showed that ANNs could outperform these methods in the examined preference modeling tasks. While SVMs showed slightly higher accuracies in few affect datasets, ANNs appear to be a more appropriate tool for affect modeling, given their enhanced expressivity that enables an easier interpretation of the learned model.

Finally, results on synthetic data suggested that the use of unclear preferences or subsets of data with different baselines complicates the learning task for the methods examined. Thus, it is expected that affect modeling can be facilitated by removing unclear user preference reports and by minimizing, whenever possible, across-users differences from the input features fed into the model. One may argue that, removing data can adulterate the results of an experiment; however, we are not proposing to remove data because it yields higher accuracies, but because the underlying function is best approximated (which in turn leads to higher accuracies); as finding the underlying function in affect reports is the main goal of affect modeling, our results suggest that using questionnaires that allow users to report unclear preferences (rather than force them to specify strong preferences) can help to draw more accurately models of affect.

In the next chapter, the focus switches to the first phases of the affect modeling methodology proposed in this thesis. Different feature extraction and selection techniques are applied in combination with the ANN training algorithms evaluated in this chapter to create models for several affective states and physiological datasets.

Chapter 6

Automatic Feature Extraction for Physiological Signals

In the preceding chapter we studied several preference learning methods that allow us to create models from ordinal annotations. As discussed in Chapter 3, these annotations allow us to reliably label affective experiences; by encoding these experiences as measurable data samples, PL becomes a particularly well suited method to create models of affect. However, the amount of information collected during each experience usually leads to large data samples that have to be reduced or otherwise PL methods (or more in general, machine learning methods) may not train meaningful models. In psycho-physiological studies, the data samples are composed by one or several physiological signals such as heart rate and skin conductance. These signals are commonly reduced through the extraction of a set of ad-hoc features (e.g. averages, variances and maximum values) proposed by an expert. Then, these features are used as the inputs of the models that estimate the target psychological states.

In this chapter we investigate state-of-the-art algorithms for automatic feature extraction offering an alternative to the expert-driven approach. We propose the utilization of convolutional auto-encoders to generate set of features that minimize the amount of information lost in the feature extraction process. CAEs represent an unsupervised training algorithm for convolutional neural networks; once trained, these networks take as input a data sample (e.g. a physiological signal) and transform it into a reduced representation. Thus, the outputs of the CNNs can be used as inputs for a predictive model in place of a set of ad-hoc features. While CAEs have shown a great potential on other domains like computer vision, they are characterized by a large number of parameters complicating their introduction to new domains. When applied to physiological signals, parameters used for computer vision are not applicable as in that domain data is arranged across 2-dimensional (images) or 3-dimensional (videos) arrays while physiological signals are 1-dimensional (signal values spanning along time). CAEs have also been applied to music where the processed data is 1-dimensional; however, those studies extracted features from the Fourier transformation of the original signal, that is, from the frequency domain (spectrogram) instead of the temporal domain (waveform). As our goal is not only to provide an algorithm that reduces the amount of information lost in the feature extraction phase but also remove, as much as possible, intermediate steps between data collection and modeling, we chose not to transform the signals into their frequency domain, however requiring an extensive parameter tuning. As the exhaustive empirical validation of all possible CAE parameters (including CNN topology) is an intractable task, the critical parameters (e.g. the patch

length, the number of layers, the number of neurons, learning rate and corruption rate) have been systematically tested, and a number of CNN parameters (e.g. pooling window length) have been fixed based on suggestions from the literature. In this chapter we report selected parameter sets which succeeded to a higher or lower degree in extracting relevant affect information from physiological signals, and summarize some of the patterns observed in the preliminary experiments.

In this chapter we showcase the expressive power of CNN features by analyzing several CNNs trained for reducing skin conductance, heart rate, blood volume pulse and blood volume signals on two dissimilar datasets, namely Maze-Ball and DEAP. We center the analysis of these CNN features on the weights of the neurons in the convolutional layers because these are the trainable parameters. Note that, as detailed in Chapter 3, each neuron evaluates every patch of the input signal (given by the output of the previous pooling layer). For each patch, the input samples are normalized (mean value within the patch equal to zero) and multiplied by the weights of the neuron; the resulting value is added the threshold of the neuron and processed by a logistic sigmoid function, producing the output of the neuron. Then, sequences of consecutive outputs are pooled together using an average or maximum function yielding one feature value. Thus, the feature value is directly linked to the multiplication between the weights and the normalized input signal, which is higher for sequences of input samples that follow a pattern similar to the sequence of weight values. Accordingly, the trained weights of a specific neuron can be visualized as a sequence that depicts the input patterns that produce maximal outputs. For each of the physiological signal and dataset examined, we present and discuss these visualizations for one selected network. As the goal of this thesis is neither to provide an extensive analysis of the affect models found on Maze-Ball and DEAP nor unveil new physiological features, we will present only the 1-convolutional-layer networks which allow for a simpler and more clear analysis than networks with a larger number of layers.

In addition to the examination of the expressivity of CNN features, we test their effectiveness by feeding them as inputs of computational models of affect and comparing them to models that take as inputs a complete set of ad-hoc statistical features common in affective computing studies. Using the PL methods studied in the previous chapter, models based on CNN features and ad-hoc features are trained to predict the MB game-related affective states (*anxiety*, *boredom*, *challenge*, *excitement*, *frustration*, *fun* and *relaxation*) and the DEAP affect dimensions (*arousal*, *valence* and *dominance*). Note that we are not concerned with the particular affect models found on these datasets or their maximum prediction accuracy, thus this chapter is focused on the evaluation of CNNs relative to standard feature extraction methods. Two different sets of CNN features and two variants of the set of ad-hoc features are evaluated allowing a more thorough evaluation and fair comparison (by balancing the number of attempts of each method). The ad-hoc features in each dataset are described in Chapter 4 and the two variants are created using two different normalization schemes. A z-transformation procedure generates one of the variants, where the mean and standard deviation of each feature across each dataset are equal to 0 and 1, respectively (within dataset normalization). The second variant is created using the same procedure but modified to achieve mean 0 and standard deviation 1 for each feature within the samples of each participant independently (within subject normalization).

For each set of features ten models were trained independently using the methods that showed more promise in Chapter 5, i.e. artificial neural networks trained using either the regularized least-squares error function with margin equal to 1.0 or the sigmoid rank-margin error function with margin equal to 0.01. Which error function is used, the training algo-

rithm (neuroevolution or backpropagation) and the neural network topology are systematically adjusted independently for each experimental condition (defined by physiological signal, affective state and feature selection method). The rest of parameters are either fixed or adjusted systematically following the same procedure as in the previous chapter. The predictive power of each feature set is given by the average 3-fold cross-validation accuracy (percentage of correctly classified pairs) of the 10 models. Furthermore, as neither the ad-hoc features nor the CNN features are necessarily relevant for predicting each of the specific affective states, sequential forward and genetic search feature selection are applied to reduce every set of features studied. Following the same procedure, SFS and GFS run 10 times, thus each producing 10 models for each feature set with potentially different features selected. T-tests are used to assess the significance of accuracy differences among pairs of experiments; significant results are considered with p-values below 0.05.

6.1 Skin Conductance

In this section we show the utility of CNNs as feature extractors for skin conductance. We present experiments with four different CNNs, two trained on the MB dataset and two on the DEAP dataset (see Table 6.1). The first CNN trained for MB, labeled $CNN^{SC}_{20\times11}$, contains two convolutional layers with 5 logistic neurons at each layer, as well as an average-pooling layer over non-overlapping windows of size 3. Each of the neurons in the first and second convolutional layer has 20 and 11 inputs, respectively. The second network for MB, labeled as CNN^{SC}_{80}, contains one convolutional layer with 5 logistic neurons of 80 inputs each. The first network trained on DEAP data, labeled $CNN^{SC}_{10\times15}$, contains two convolutional layers with 5 and 15 logistic neurons at each layer, as well as an average-pooling layer over non-overlapping windows of size 5. Each of the neurons in the first and second convolutional layers has 10 and 15 inputs, respectively. The second CNN for DEAP, labeled as CNN^{SC}_{30}, contains one convolutional layer with 15 logistic neurons, with 30 inputs at each neuron.

All topologies are built on top of an average-pooling layer over non-overlapping windows of the raw SC signals; for both MB networks the window length is fixed at 20 samples while for the DEAP networks, CNN^{SC}_{30} and $CNN^{SC}_{10\times15}$, a window length of 512 and 256 samples is selected, respectively. This initial layer reduces through an average function, the resolution of the raw SC signal from 31.25 Hz to 1.56 Hz in MB and from 512 Hz to 1 Hz (in DEAP with CNN^{SC}_{30}) and to 2 Hz (in DEAP with $CNN^{SC}_{10\times15}$). Although SC is usually sampled at high frequencies (e.g. 512 Hz), related studies have shown that the most affect-relevant information contained in the signal can be found at a lower time resolution as even rapid arousal changes (i.e. a phasic change of SC) can be captured with a lower resolution and at a lower computational cost (Ravaja et al., 2006; Yannakakis et al., 2010). For that purpose, the selection of this initial pooling stage aims to facilitate feature learning at low resolutions. Moreover, preliminary experiments with dissimilar pooling layers showcased that features extracted on higher SC resolutions do not necessarily yield models of higher accuracy.

All four CNNs examined here are selected based on a number of criteria. The number of inputs of the first convolutional layer of the CNNs considered were selected to extract features at different time resolutions and, thereby, give an indication of the impact the time resolution might have on performance. In MB 20 and 80 inputs correspond to 12.82 and 50.56 seconds, respectively; in DEAP 10 and 30 inputs correspond, in turn, to 5 and 30 seconds. Extensive preliminary experiments with smaller and larger time windows did not seem to affect the model's prediction accuracy. For the 2-convolutional-layer networks, five

		Maze-Ball		DEAP	
		$CNN^{SC}_{20\times11}$	CNN^{SC}_{80}	$CNN^{SC}_{10\times15}$	CNN^{SC}_{30}
Pooling layer 1	Function	Average	Average	Average	Average
	Window	20	20	512	256
Convolutional layer 1	Neurons	5	5	5	15
	Inputs	20	80	10	30
Pooling layer 2	Function	Average	Average	Average	Average
	Window	3	*	5	*
Convolutional layer 2	Neurons	5	—	15	—
	Inputs	11	—	15	—
Pooling layer 3	Function	Average	—	Average	—
	Window	*	—	*	—

Table 6.1: Convolutional neural network topologies for skin conductance. The pooling function and the window length of each pooling layer, and the number of neurons and the number of inputs per neuron of each convolutional layer are specified. The window length in the output layers (*) is adjusted to generate a total of 15 outputs.

neurons were selected for the first layer as a good compromise between expressivity and dissimilarity among the features learned: a low number of neurons derived features with low expressivity while a large number of neurons generally resulted in features being very similar. The small window on the intermediate pooling layer was chosen to minimize the amount of information lost from the output of the first convolutional layer (i.e. feature maps) while the number of inputs to the neurons in the next layer was adjusted to cover about a third and above half of the pooled feature maps in $CNN^{SC}_{20\times11}$ and $CNN^{SC}_{10\times15}$, respectively. Five output neurons were used for MB and 15 for DEAP; a final pooling layer reduces the number of outputs per neuron to 3 samples in MB networks and 1 in DEAP networks. Hence, the networks trained for MB produce 15 CNN features which correspond to the output of 5 neurons averaged over 3 contiguous time windows along the 90 second experience; similarly, the networks trained for DEAP generate 15 CNN features but these correspond to the output of 15 neurons averaged over one single time window over the 60 second experience. The selection of the number of neurons and pooling in the last layer was made to achieve the exact number of ad-hoc statistical features of SC (i.e. 15) which allow us to make a more fair comparison. The number of output neurons is larger for DEAP networks because this dataset contains a larger number of training samples, thus a wider range of different patterns could potentially be found and therefore a larger number of distinct features could be learned.

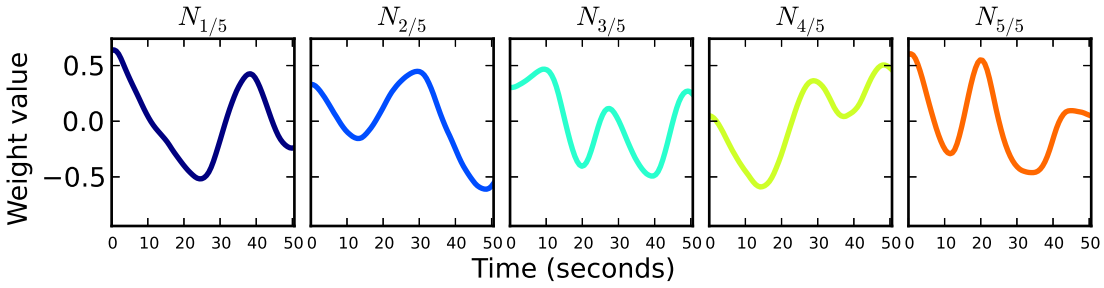
6.1.1 Deep Learned Features

This section showcases the expressivity potential of CNN features by analyzing some of the convolutional networks learned from SC data; as mentioned in the beginning of this chapter, this analysis spans only the 1-convolutional-layer networks and is centered on the weights of the neurons in this layer. Figure 6.1a depicts the values of the 80 connection weights of the five neurons in CNN^{SC}_{80}. These weights cover SC signal patches of 51.2 seconds (0.64 seconds per weight). The first neuron ($N_{1/5}$) outputs a maximal value for areas of the SC signal in which a long decay is followed by 10 seconds of an incremental trend and a final decay. The second neuron ($N_{2/5}$) shows a similar pattern but the increment is detected

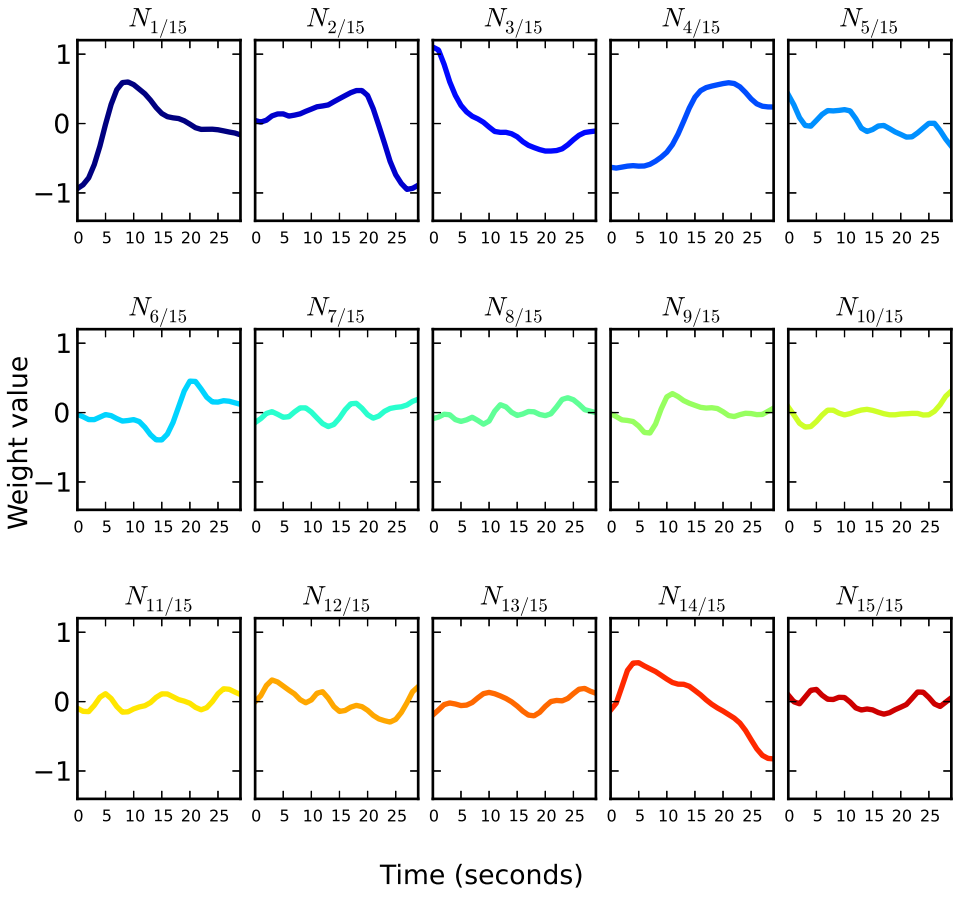
earlier in the time window and the follow-up decay is longer. A high output of these neurons would suggest that a change in the experience elicited a heightened level of arousal that decayed naturally seconds after. The forth neuron ($N_{4/5}$) in contrast, detects a second incremental trend in the signal that elevates the SC level even further. The fifth neuron ($N_{5/5}$) also detects two increments but several seconds further apart. Finally, the third neuron ($N_{3/5}$) detects three consecutive SC increments. These last three neurons could detect changes on the level of arousal caused by consecutive stimuli presented few seconds apart. Overall, this convolutional layer captures long and slow changes (10 seconds or more) of skin conductance. These local patterns cannot be modeled with the same precision using standard statistical features related to variation (such as standard deviation and average first/second absolute differences), which further suggests that dissimilar aspects of the signal are extracted by learned and ad-hoc features.

Figure 6.1b depicts the value of the 30 connection weights of the 15 neurons in CNN_{30}^{SC}. These neurons cover 30 seconds of the SC signal (1 second per weights) on each evaluation. By minding the shorter (in seconds) input patch, a large number of similarities with CNN_{80}^{SC} are observed. In particular, the first ($N_{1/15}$) and fourth ($N_{4/15}$) neurons output a maximal value for areas of the SC signal characterized by a sudden increment during 10-15 seconds, followed by a decay that leaves the signal in a heightened level; these features resemble the central pattern captured by $N_{1/5}$ and $N_{4/5}$. The fourteenth neuron ($N_{14/15}$) highlights areas marked by a long decay (20 seconds) following to an increment; this pattern, in turn, is analogous to the final and intermediate segments of $N_{2/5}$ and $N_{5/5}$, respectively. The third neuron ($N_{3/15}$) presents also a pattern similar to that observed in $N_{2/5}$ and $N_{5/5}$ but with a 5-second-long offset with respect to $N_{14/15}$ (starting at the highest point). Finally, the second neuron ($N_{2/15}$) detects an identical component to the initial segment of $N_{3/5}$, i.e. a slow increasing trend followed by a strong decrement. These similarities suggest that, first and unsurprisingly, similar components construct the SC signals in both datasets despite the different stimuli, hardware and data collection protocols; and second, that auto-encoders were able to find the same patterns despite the different number of training samples of each dataset, and different patch lengths and time resolutions of each CNN.

The remaining neurons captured additional patterns. Specifically, the sixth neuron ($N_{6/15}$) and ninth neuron ($N_{9/15}$) detect a short increment in the SC signal (approximately 5 seconds) while the rest of the neurons react to different bursts of 2 or 3 small increments. These are components of higher frequency that were not captured by the longer time window (and lower number of neurons) of CNN_{80}^{SC}.



(a) CNN_{80}^{SC}



(b) CNN_{30}^{SC}

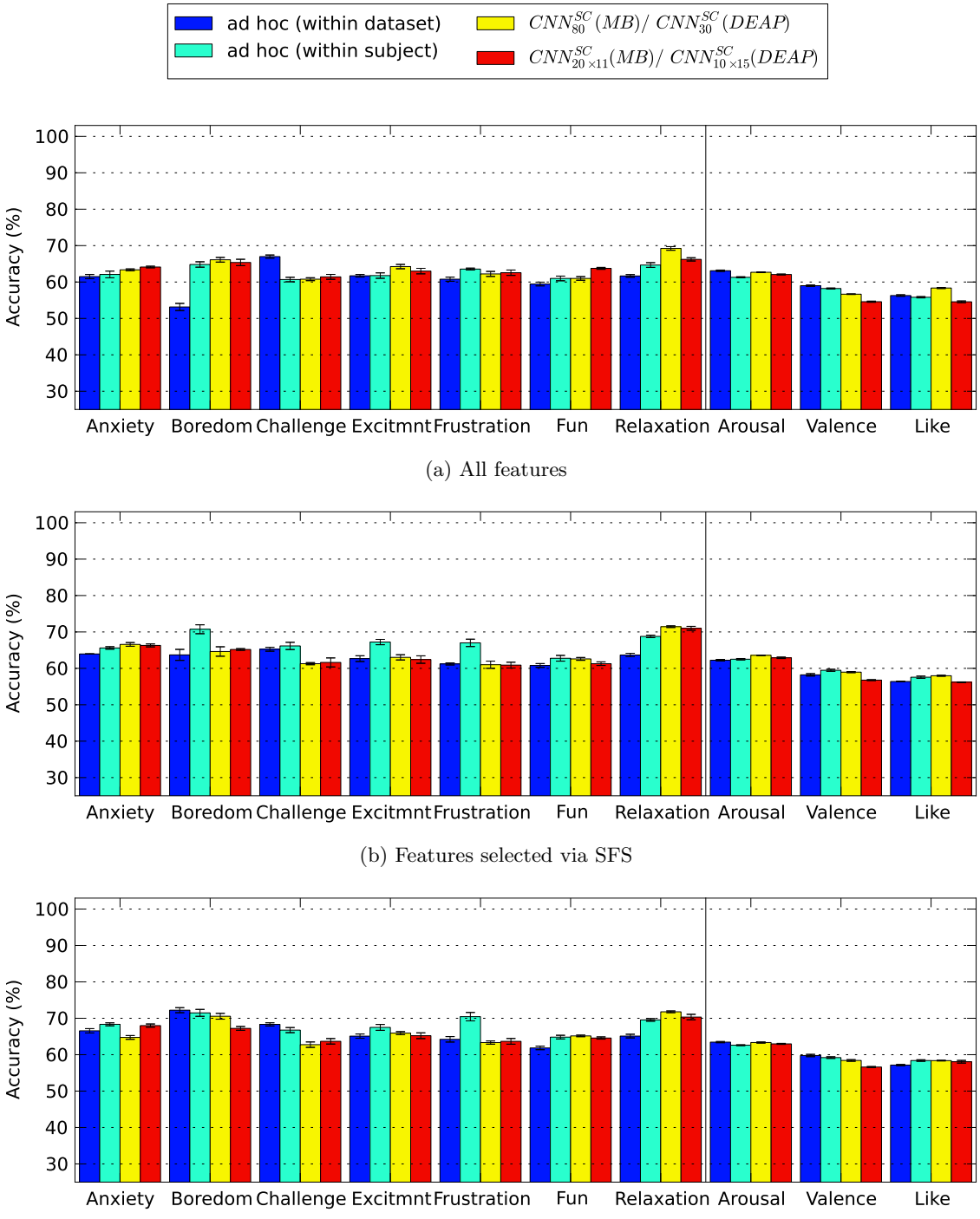
Figure 6.1: Learned features of the 1-layer convolutional neural networks trained for SC. Lines are plotted connecting the values of consecutive connection weights for each neuron $N_{x/total}$. The x axis displays the time stamp (in seconds) of the samples connected to each weight within the input patch.

6.1.2 Deep Learning vs. Ad-hoc Feature Extraction

In this section we provide a systematic evaluation of the skin conductance CNN features by comparing their prediction power against ad-hoc features. Figure 6.2a depicts the average prediction accuracies of ANNs trained on all the outputs of the CNNs compared to the corresponding accuracies obtained by ANNs trained on the complete set of ad-hoc statistical features. CNN features yield models with a significantly higher accuracy than ad-hoc features for the MB states of *relaxation*, *fun*, *excitement* and *anxiety*. Given the performance differences among these networks, it appears that learned local features could detect aspects of SC that were more relevant to the prediction of these particular affective states than the set of ad-hoc statistical features proposed. Figure 6.2b and 6.2c show that the highest accuracies achieved using automatically selected features maintain a significant difference for *relaxation* while for the remaining 3 states the accuracies meet at similar values. Furthermore, the *relaxation* and *fun* models trained on selected ad-hoc features, despite the benefits of FS, yield accuracies which are not significantly different than the models trained on the complete sets of learned features. This suggest that CNNs can extract general information from SC that is more relevant for affect modeling than ad-hoc features selected specifically for the task.

On the other hand, models trained on ad-hoc features outperform CNN-based models across feature selection schemes on the prediction of *frustration* and *challenge*. This significant difference suggests that certain aspects of the signals were not adequately captured by the convolutional networks. In particular, the normalization the input signals within each patch facilitates the learning of *relative* or *local* features at the expense of discarding *absolute* or *global* information. This information is, however, captured by simple ad-hoc features such as maximum $\max\{SC\}$ and initial SC_{ini} values. These features with absolute information are found in the best frustration model built on GFS-selected features (specifically, we find among its inputs SC_{ini}, $\max\{SC\}$ and time when the minimum value is recorded $t_{\min}\{SC\}$) and the best challenge model also built in GFS-selected features (receives as inputs SC_{ini}, $\max\{SC\}$, minimum value $\min\{SC\}$, difference between maximum and minimum values of SC D^{SC}). It appears that despite their simplicity, these features yield more relevant information for the prediction of some affective and cognitive states than the CNN topologies used here. For the remaining MB state, *boredom*, models trained on selected features are no significantly different between ad-hoc and learned features. For the DEAP dataset, only models for *arousal* present accuracies above 60% and no significant difference is observed between ad-hoc and CNN models.

Despite the difficulty in predicting complex affective states based solely on SC, these results suggest that unsupervised CNNs trained as a stack of denoising auto-encoders form a promising method for automatically extracting features from this modality, as competitive prediction accuracies were achieved when compared against a well-defined set of ad-hoc statistical features. Results also show that there are particular affective states (*relaxation* and *fun*), in which DL is able to automatically extract features that are beneficial for their prediction. On the other hand, the within-patch normalization used prevents DL from capturing certain absolute characteristics of the SC signals (such as initial and maximum values) which appear to be useful to predict some affective states (*frustration* and *challenge*). With an alternative normalization (e.g. normalize the complete signals within each participant prior presentation to the network), CNNs could capture those simple and possibly other more advanced absolute patterns; thus, it is expected that combining CNNs with different normalizations can reach and surpass the accuracies shown by the ad-hoc



(a) All features

(b) Features selected via SFS

(c) Features selected via GFS

Figure 6.2: Skin conductance: average accuracy of ANNs trained on ad-hoc features normalized across the complete dataset (within dataset) and within each participant independently (within subject), and features generated by each of the CNN topologies ($CNN_{20\times11}^{SC}$ and CNN_{80}^{SC} for MB, and $CNN_{10\times15}^{SC}$ and CNN_{30}^{SC} for DEAP). The black bar displayed on each average value represents the standard error (10 runs).

feature-based models. One could argue that the same procedure could be applied on the opposite direction, i.e. ad-hoc features could be designed to match the accuracies of DL in the previously presented states. In fact, DL can be used to find new ad-hoc feature extractors by coding the patterns learned in the networks. However, an ad-hoc set of features is by definition finite and may not contain the appropriate indexes for a given dataset. Creating a massive repository would not yield optimal solutions either; not even when using dimensionality reduction methods. Unsupervised feature learning (if adequately tuned) has the potential of identifying automatically every relevant component of any given dataset in a compact and homogeneous set of features, yielding more accurate predictors without the intervention of a human designer.

6.2 Blood Volume and Blood Volume Pulse

Following the same systematic approach for selecting CNN topology and parameter sets, two convolutional networks for blood volume pulse (in Maze-Ball) and other two for blood volume (in DEAP) are presented and evaluated in this section. These CNN topologies are summarized in Table 6.2. Those used for BVP feature the following: (1) one max-pooling layer with non-overlapping windows of length 30 followed by a convolutional layer with 5 logistic neurons and 45 inputs at each neuron (CNN_{45}^{BVP}); and (2) two convolutional layers with 10 and 5 logistic neurons, respectively, and an intermediate max-pooling layer with a window of length 30. The neurons of each layer contain 30 and 45 inputs, respectively ($CNN_{30\times45}^{BVP}$). Both topologies are topped up with an average-pooling layer that reduces the number of outputs from each of the 5 output neurons down to 3 — i.e. the CNNs output 5 feature maps of length 3 which amounts to 15 features. The initial pooling layer of the first network collects the maximum value of the BVP signal every 0.96 seconds, which results in an approximation of the signal's *upper envelope* — i.e. a smooth line joining the extremes of the signal's peaks. Decrements in this function are directly linked to increments in heart rate, and further connected to increased arousal and corresponding affective states (e.g. excitement and fun as shown in the studies of Yannakakis et al. (2010); Yannakakis and Hallam (2008). Neurons with 45 inputs were selected to capture long patterns (i.e. 43.2 seconds) of variation, as sudden and rapid changes in heart rate were not expected during the game survey experiment. The second network follows the same rationale but the first pooling layer — instead of collecting the maximum of the raw BVP signal — processes the outputs of 10 neurons that analyze signal patches of 0.96 seconds, which could operate as a beat detector mechanism.

The two CNN architectures presented for BV (in DEAP) feature an initial max-pooling layer with non-overlapping windows of length 512. In both topologies, one convolutional layer is stacked on top with 9 logistic neurons and 20 and 30 inputs at each neuron, respectively, to each topology (labeled CNN_{20}^{BV} and CNN_{30}^{BV}). Finally, an average-pooling with length 11 (for CNN_{20}^{BV}) and max-pooling layer with length 10 (for CNN_{30}^{BV}) reduces the number of outputs of each neuron down to 3 yielding a total of 27 outputs per network. The initial pooling layer yields a smoother signal at 1 Hz that should facilitate learning patterns at low frequencies (above heart beat frequency). High frequency information may also be important; however, reducing the initial time resolution of the signal speeds up CNN training (as the number of training patches is reduced) while a large part of the relevant information in those high frequencies is present in the heart rate signal (and can be extracted from that modality). The two different topologies are designed to capture patterns

		Maze-Ball		DEAP	
		CNN_{45}^{BVP}	$CNN_{30\times45}^{BVP}$	CNN_{20}^{BV}	CNN_{30}^{BV}
Pooling layer 1	Function	Maximum	—	Maximum	Maximum
	Window	30	—	512	512
Convolutional layer 1	Neurons	5	10	9	9
	Inputs	45	30	20	30
Pooling layer 2	Function	Average	Maximum	Average	Maximum
	Window	*	30	*	*
Convolutional layer 2	Neurons	—	5	—	—
	Inputs	—	45	—	—
Pooling layer 3	Function	—	Average	—	—
	Window	—	*	—	—

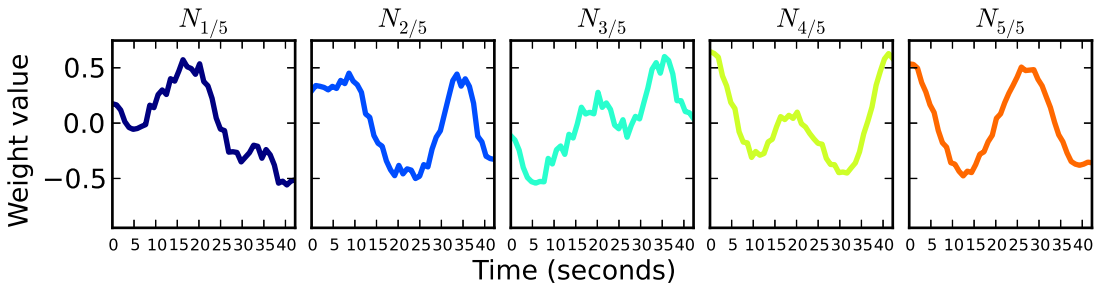
Table 6.2: Convolutional neural network topologies for blood volume and blood volume pulse. The pooling function and the window length of each pooling layer, and the number of neurons and the number of inputs per neuron of each convolutional layer are specified. The window length in the output layers (*) is adjusted to generate a total of 15 and 27 outputs for the MB and DEAP networks, respectively.

of different length (20 and 30 seconds) and the number of neurons and pooling is adjusted to approximate the number of BV ad-hoc features (i.e. 26).

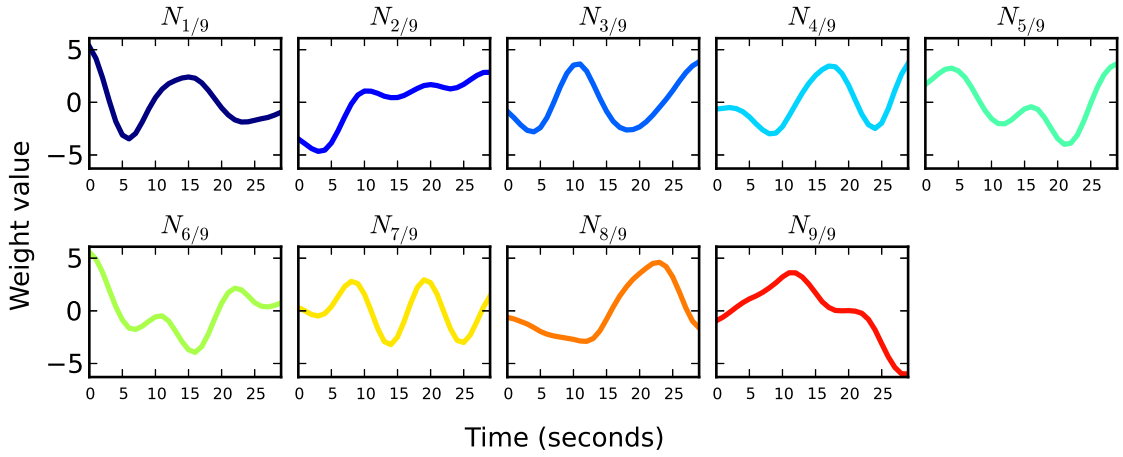
6.2.1 Deep Learned Features

In this section we examine one of the networks trained for BVP and another trained for BV. Figure 6.3a depicts the 45 connection weights of each neuron in CNN_{45}^{BVP} which cover 43.02 seconds of the BVP signal's upper envelope. Given the negative correlation between the trend of the BVP's upper envelope and heart rate, neurons would output maximal values when consecutive decreasing weight values are aligned with an area in the BVP signal corresponding to a HR increment and consecutive increasing weight values with HR decays. On that basis, the second ($N_{2/5}$) and fifth ($N_{5/5}$) neurons detect two 10-second-long periods of HR increments, which are separated by a HR decay period. The first ($N_{1/5}$) and the forth ($N_{4/5}$) neuron detect two overlapping increments on HR, followed by a decay in $N_{4/5}$. The third neuron ($N_{3/5}$), on the other hand, detects a negative trend on HR with a small peak in the middle. This convolutional layer appears to capture dissimilar local complex patterns of BVP variation which are, arguably, not available through common ad-hoc statistical features.

Figure 6.3b depicts the 30 connection weights of each neuron in CNN_{30}^{BV}. While a rather wide variety of patterns is observed in this network, none of them appears to resemble the BVP patterns learned from MB. Despite the similarity of the recording protocols — i.e. the same sensor technology (*plethysmography*) applied to the finger tips — this result only confirms the dissimilarity between these two signals (Lidberg et al., 1974). In particular, each of the pairs $N_{3/9}$-$N_{4/9}$ and $N_{5/9}$-$N_{6/9}$ detects one similar pattern with an offset of approximately 4 seconds, $N_{7/9}$ presents its maximal output for signals with a period of 11 seconds, $N_{2/9}$ detects a fast and slow accumulative increments, $N_{9/9}$ an increment for 11 seconds and decrement for the remaining 19 seconds, and $N_{1/9}$ and $N_{8/9}$ feature a 15 seconds peak (7.5 seconds increment, 7.5 decrement) preceded by a fast and strong or slow and mild decrement, respectively. These are complex variation patterns that are unlikely



(a) CNN_{45}^{BVP}



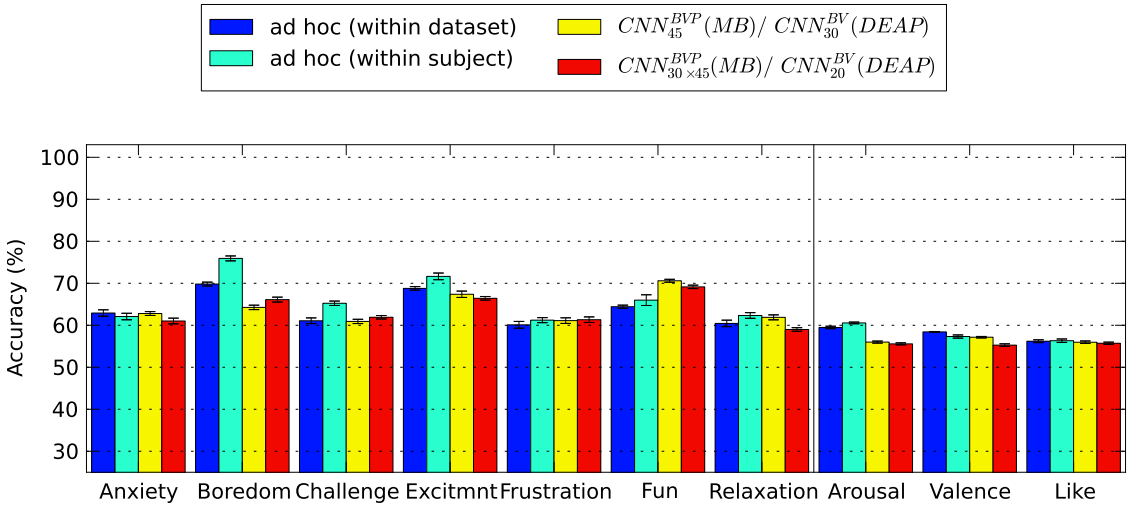
(b) CNN_{30}^{BVP}

Figure 6.3: Learned features of selected convolutional neural networks trained for BVP and BV. Lines are plotted connecting the values of consecutive connection weights for each neuron $N_{x/total}$. The x axis displays the time stamp (in seconds) of the samples connected to each weight within the input patch.

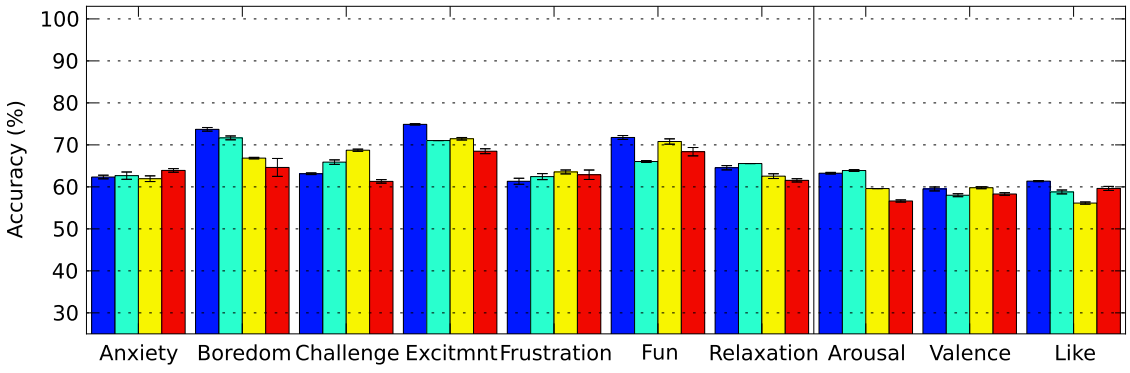
to be well represented by standard ad-hoc features.

6.2.2 Deep Learning vs. Ad-hoc Feature Extraction

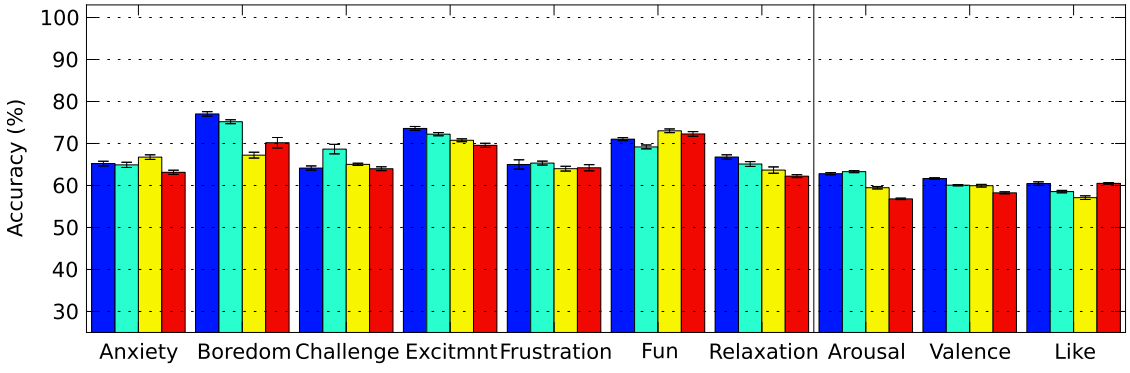
In this section we compare the prediction accuracies of ANN models trained on any of the four CNNs described above against ANN models trained on BVP and BV ad-hoc features. Predictors of *fun* trained on features extracted with CNN_{45}^{BVP} outperformed the ad-hoc feature sets without feature selection by a large margin — accuracies of 70.6% and 65.99%, respectively — (see Fig. 6.4a). Automatic feature selection generates a larger improvement for the ad-hoc models yielding a non significant difference between the best models (see Fig. 6.4). Given the reported links between fun and heart rate (Yannakakis and Hallam, 2008), this result suggests that CNN_{45}^{BVP} effectively extracted HR information directly from the BVP signal — i.e. without beat extractors — to predict reported *fun*. The efficacy of CNNs is further supported by the results reported in (Martínez et al., 2011) where SLP predictors of fun trained on ad-hoc statistical features of the HR signal (in Maze-Ball) do not outperform the DL models presented here. For reported fun, CNN-based feature extraction demonstrates a great advantage of extracting affect-relevant information from



(a) No feature selection



(b) Sequential feature selection



(c) Genetic feature selection

Figure 6.4: Blood Volume Pulse and Blood Volume: average accuracy of ANNs trained on ad-hoc features normalized across the complete dataset (within dataset) and within each participant independently (within subject), and features generated by each of the CNN topologies ($CNN^{BVP}_{30\times45}$ and CNN^{BVP}_{45} for MB, and CNN^{BV}_{20} and CNN^{BV}_{30} for DEAP). The black bar displayed on each average value represents the standard error (10 runs).

BVP bypassing beat detection and heart rate estimation.

On the other hand, for *arousal*, *boredom* and *excitement*, models trained on all and selected ad-hoc features significantly outperform CNN based models. Furthermore, a similar difference is observed among predictors of *relaxation*, *like* and *valence* trained on automatic selected features. Observation of the subsets of selected features reveals that at least one of the inter-beat amplitude features ($E\{IBAmp\}$ and $\sigma\{IBAmp\}$) were present in every of the best models as well as the power frequency spectrum features (e.g. high frequency power HF) in a smaller number of models. While precise beat information appears to be crucial to build accurate models of several affective states from BVP and BV, CNN_{45}^{BVP}, CNN_{20}^{BV} and CNN_{30}^{BV} have no access to any trace from the heart beats as consequence of the initial pooling layer, explaining the lower accuracies of CNN features in certain affective states. Even though $CNN_{30\times45}$ was designed to include heart beat information, it appears that the topology chosen was not sufficient to capture information about the beat amplitude. Despite missing this information, *excitement* and *boredom* still show high accuracies around 70%. Finally, non significant differences are observed for the remaining MB states — *frustration*, *challenge* and *anxiety* — among the best feature selection results suggesting that in absence of a strong link between beat amplitude and target affective states, convolutional networks can extract appropriate features from BVP for predicting affect.

In all, results presented in this section show that CNNs can extract relevant information for affect prediction from the raw BVP and BV signals. Models of several affective states trained on these features are comparable to models that rely on advanced ad-hoc features that build on beat detectors (e.g. inter-beat amplitude) and Fourier transformations (e.g. high frequency power of BVP). It was clear, however, that information regarding inter-beat amplitude could not be captured by the topologies tested leading to models with significant lower accuracies than models relying on the average and standard deviation of $IBAmp$.

6.3 Heart Rate

As with the previous modalities, two CNN topologies are chosen for each HR dataset (see the detailed parameters in Table 6.3). All four topologies include one single convolutional layer and one average-pooling layer that extracts exactly 15 outputs. For MB the two networks designed feature: (1) 15 neurons and 30 inputs at each neuron (CNN_{30}^{HR-MB}) and (2) 3 neurons and 15 inputs at each neuron (CNN_{15}^{HR-MB}). Analogously for DEAP the networks contain: (1) 5 neurons per patch location and 30 inputs at each neuron ($CNN_{30}^{HR-DEAP}$) and (2) 3 neurons per patch location and 15 inputs at each neuron ($CNN_{15}^{HR-DEAP}$). Each of the neurons in these networks captures 15-second or 30-second long patterns in the HR signals, depending on the number of inputs (1 second per input as HR is sampled at 1 Hz). That length amounts to a third and half of the experience in MB and DEAP, respectively; 15 and 5 neurons are chosen to find a variety of distinctive patterns in the long 30 seconds intervals while a lower number of neurons (i.e. 3) is chosen for the 15-second long windows as a small number of different patterns are expected.

6.3.1 Deep Learned Features

Two dissimilar topologies are selected for analysis in this section. For the HR signal in MB we examine a network with 15 features (CNN_{30}^{HR-MB}). By observing Figure 6.5a it appears that, the total number of 30-second-long distinct patterns could be captured with less than 15 neurons as several pairs of neurons in CNN_{30}^{HR-MB} learned similar weight vectors

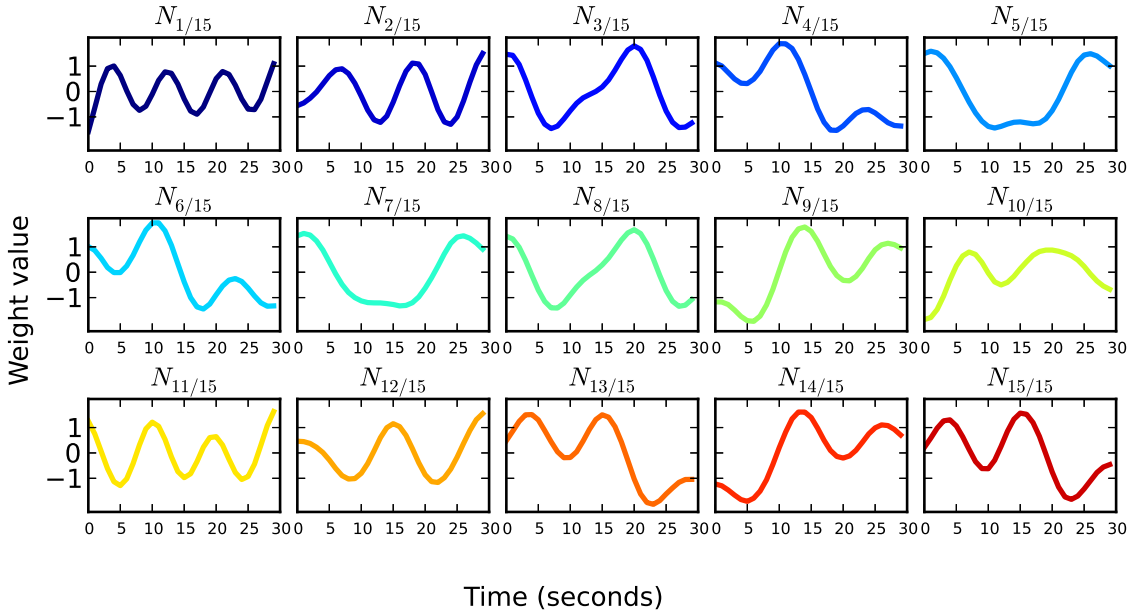
		Maze-Ball		DEAP	
		CNN_{30}^{HR-MB}	CNN_{15}^{HR-MB}	CNN_{30}^{HR}	CNN_{15}^{HR}
Convolutional layer 1	Neurons	15	3	5	3
	Inputs	30	15	30	15
Pooling layer 2	Function	Average	Average	Average	Average
	Window	*	*	*	*

Table 6.3: Convolutional neural network topologies for heart rate. The pooling function and the window length of each pooling layer, and the number of neurons and the number of inputs per neuron of each convolutional layer are specified. The window length in the output layers (*) is adjusted to generate a total of 15 outputs.

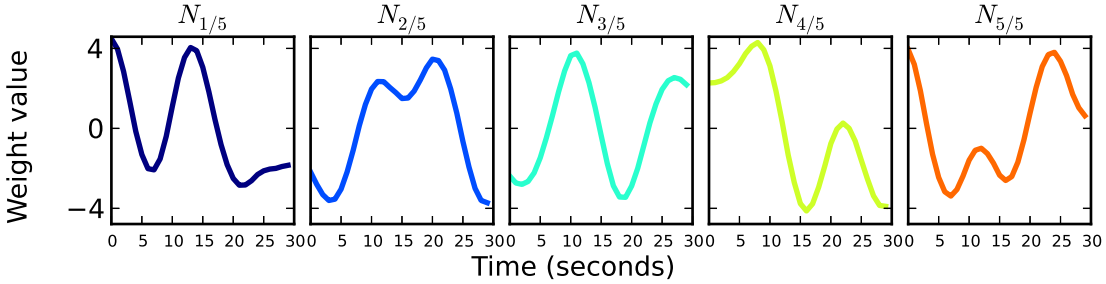
($N_{3/15} \leftrightarrow N_{8/15}$, $N_{4/15} \leftrightarrow N_{6/15}$, $N_{5/15} \leftrightarrow N_{7/15}$, $N_{9/15} \leftrightarrow N_{14/15}$ and $N_{13/15} \leftrightarrow N_{15/15}$). While these features may be redundant for training a classifier from a small number of data samples, a predictor could be trained to extract very detailed information from subtle changes in the signals given a large dataset; for example, the pair $N_{4/15}$-$N_{6/15}$ can be used to distinguish between the relative magnitude of a small peak compared to a preceding larger peak. Another group of similar patterns can be identified in $N_{1/15}$, $N_{11/15}$, $N_{2/15}$ and $N_{12/15}$: these four neurons show an almost periodic signal with an increasing period across them (from 8 seconds in the $N_{1/15}$ to 12 seconds in $N_{12/15}$). Applied together to the same input signal, these neurons estimate different frequency components contained between 0.125 Hz and 0.083 Hz, within the low frequency band (0.04 to 0.15 Hz, LF); the amount of variation within this band conforms a popular index of heart-rate variability as several studies have drawn links to parasympathetic activity (Goldberger et al., 2001). This result highlights the power of convolutional auto-encoders as they could automatically learn a well-known HR feature, very relevant in affective computing research. It is thus expected that this approach can also discover new interesting features not used before.

Most of the other neurons show a combination of more dissimilar variations combining increments and decrements that last between 5 and 10 seconds each. $N_{3/15}$ and $N_{8/15}$ capture longer sustained increments (13 seconds approximately) while $N_{7/15}$ and $N_{5/15}$ react to periods of small changes (during 10 seconds) followed by an increment (about 8 seconds). Overall, these neurons appear to capture different non-periodic patterns of variation characterized by changes that are sustained for at least 5 seconds. Changes in periods below 5 seconds were not expected to be significant as the HR signal was calculated using a 5-second long sliding window. While ad-hoc features of HR variability are usually extracted from RR intervals, CNNs discovered some popular indexes on the HR signal and other complex features possibly not represented in typical HR and HRV feature sets.

Figure 6.5b depicts the weights of a $CNN_{30}^{HR-DEAP}$ which features only 5 neurons but the same number of inputs at each neuron than the network analyzed above. Each of the five neurons would output a maximal value for a different sequence of alternating increments and decrements each of them lasting between 5 and 10 seconds, approximately. Variations with similar lengths of time were observed in most of the neurons in CNN_{30}^{HR-MB}, suggesting the recurrence of these HR patterns across dissimilar activities of low physical demand (playing a computer game and watching music videos).



(a) CNN_{30}^{HR-MB}

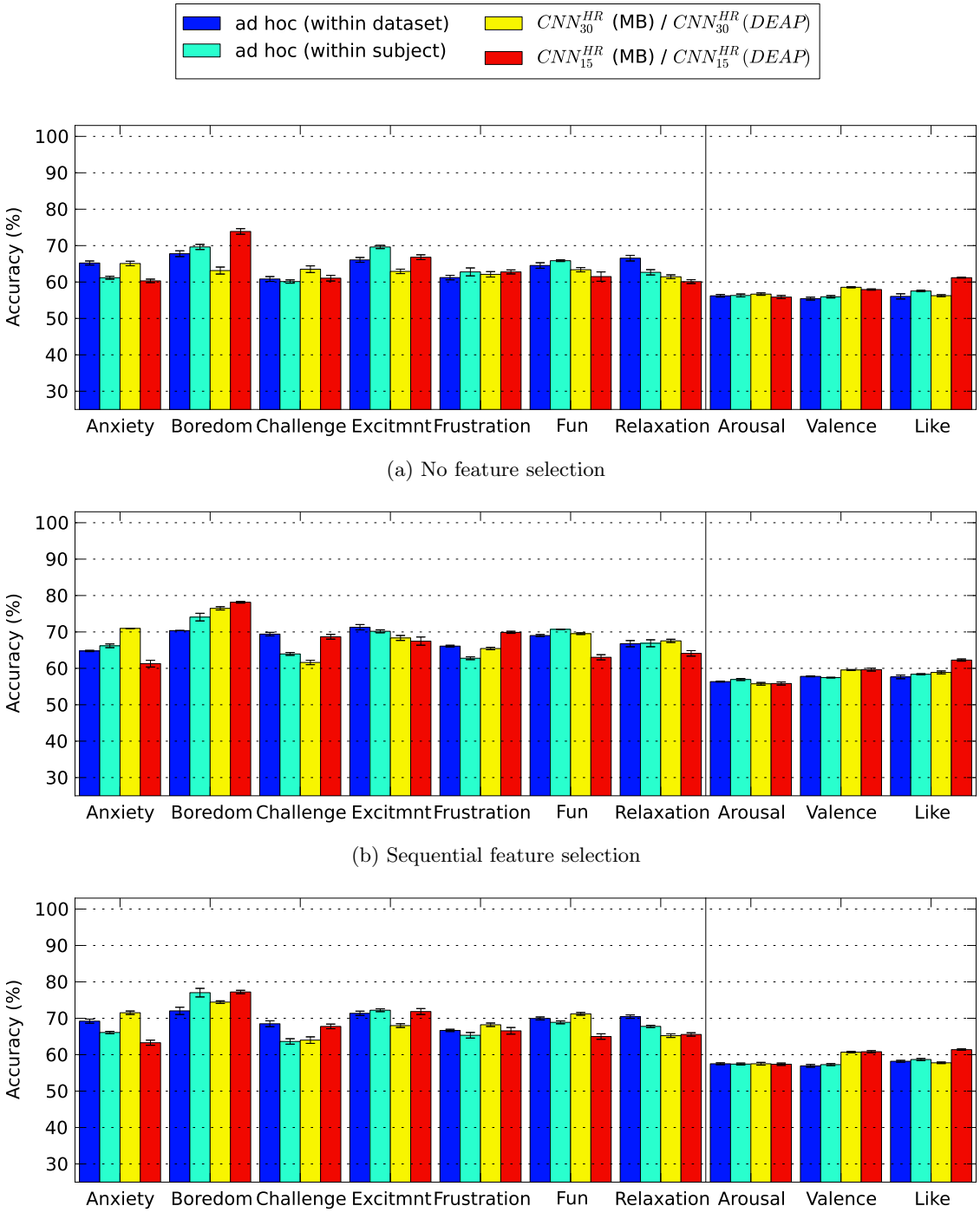


(b) $CNN_{30}^{HR-DEAP}$

Figure 6.5: Learned features of selected convolutional neural networks trained for HR. Lines are plotted connecting the values of consecutive connection weights for each neuron $N_{x/total}$. The x axis displays the time stamp (in seconds) of the samples connected to each weight within the input patch.

6.3.2 Deep Learning vs. Ad-hoc Feature Extraction

In this section we compare the prediction accuracies of ANN models trained on the HR CNNs against ANN models trained on HR ad-hoc features. In these experiments reported CNN features of HR yield significantly more accurate models than ad-hoc features in several MB states and DEAP dimensions. In particular *boredom* and *challenge* models trained on CNN features yield significantly higher accuracies than the corresponding ad-hoc models when no feature selection is applied (see Figure 6.6a). Feature selection increases the accuracies of these models dissimilarly, yielding non significant differences between best CNN and ad-hoc selected feature sets as seen in Figure 6.6b and Figure 6.6c. Models for *frustration* and *anxiety* , in turn, do not present a significant difference in accuracy across the complete set of extracted features, while feature selection pushes the accuracy of CNN



(a) No feature selection

(b) Sequential feature selection

(c) Genetic feature selection

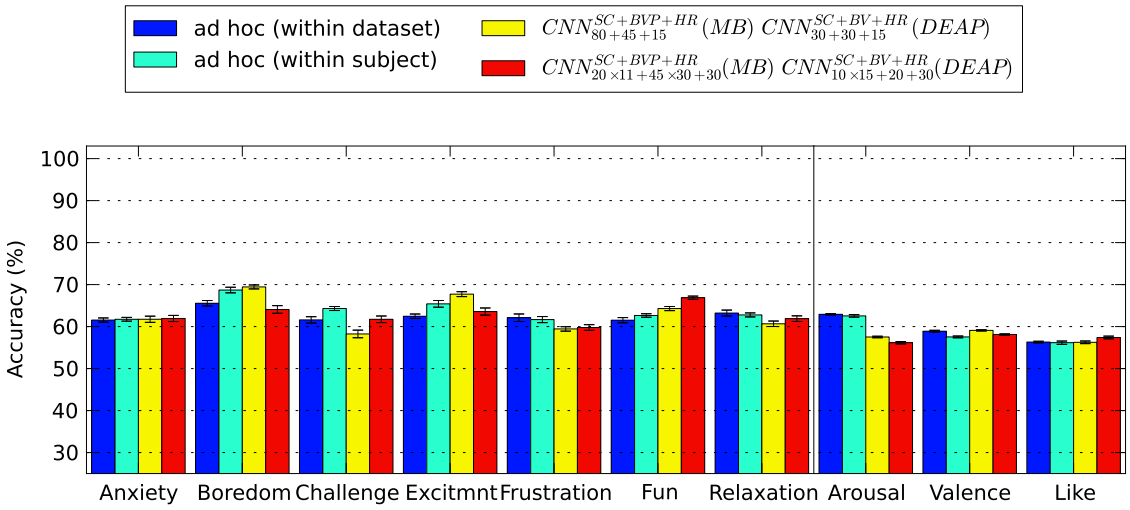
Figure 6.6: Heart Rate: average accuracy of ANNs trained on ad-hoc features normalized across the complete dataset (within dataset) and within each participant independently (within subject), and features generated by each of the CNN topologies (CNN_{30}^{HR-MB} and CNN_{15}^{HR-MB} for MB, and $CNN_{30}^{HR-DEAP}$ and $CNN_{15}^{HR-DEAP}$ for DEAP). The black bar displayed on each average value represents the standard error (10 runs).

features to a significantly higher value than the best ad-hoc model . Within DEAP, CNN features yield more accurate models of *valence* and *like* compared to the accuracies obtained from the ad-hoc features both when all the features are concerned and when they are automatically selected via SFS and GFS. From these results it appears that CNNs can extract features from HR that are more relevant for the prediction of several affective constructs than popular HR ad-hoc features.

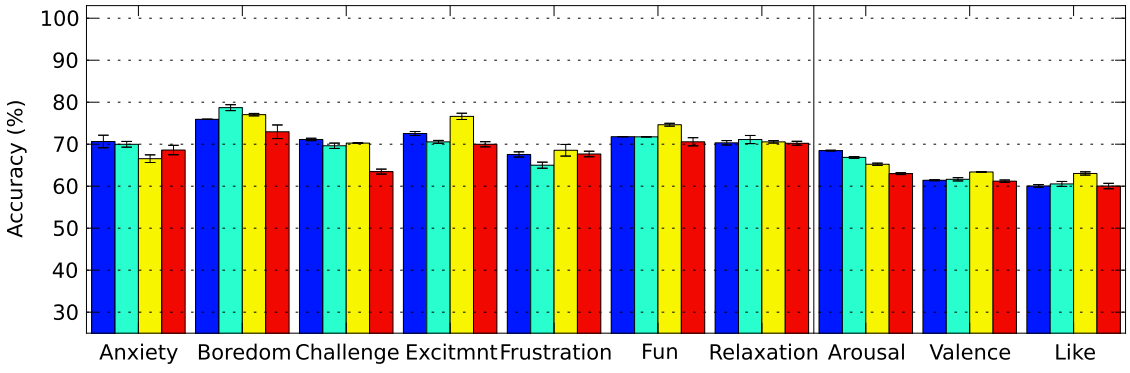
On the other hand, CNN feature models are outperformed by ad-hoc feature models in a few settings. In particular, *excitement* and *fun* models built on all ad-hoc features yield significantly better results than CNN feature sets but accuracies are evened up after features are automatically selected with GFS. This result highlights the importance of feature selection even for CNN extracted features. Finally, *relaxation* models yield better or similar results with ad-hoc features across selection mechanisms. The best *relaxation* model, trained on features selected via GFS, is an MLP with a single input feature: time when minimum HR was detected. The CNN that yields better results contains neurons that cover 30 seconds of the raw HR signal (out of 90 in MB). With the patch-wise normalization applied in our experiments, it would have been impossible to distinguish whether the minimum is detected in the first, second or third segment of the game. A larger time window or a different normalization scheme (as discussed in Section 6.1) would help CNNs to capture these global patterns.

6.4 Fusion

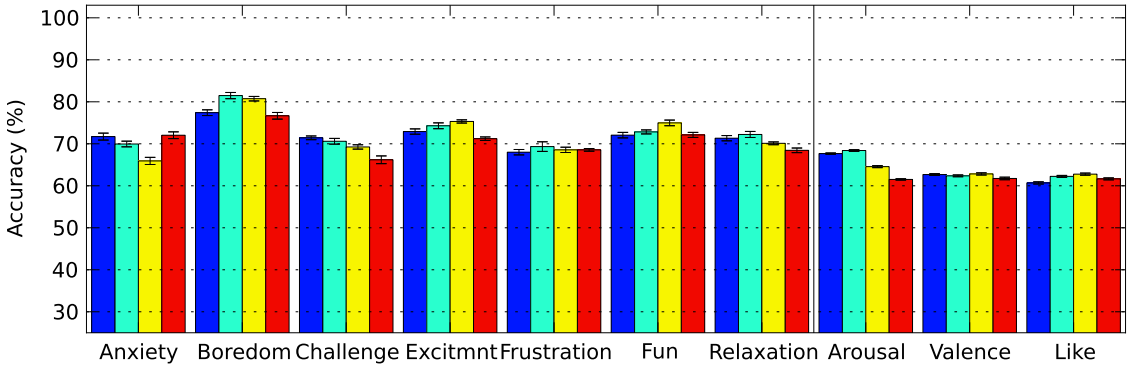
To test the effectiveness of learned features in models fusing several input modalities, the outputs of the BVP/BV, SC and HR CNN networks presented earlier are combined into one ANN and its performance is compared against a combination of all ad-hoc BVP/BV, SC and HR features. As depicted in Figure 6.7, differences observed in this experiment confirm the high relevance of CNN-learned features for predicting several affective states and the limitations of the approach reported previously in the study of single modalities. Specifically, the fusion of CNNs from all signals generates models that yield higher prediction accuracies than models built on ad-hoc features in the MB states of *fun* and *excitement*. In both states the models built on the complete set of features and the best performing automatically selected set yielded significantly more accurate models than the corresponding ad-hoc feature sets. Furthermore, automatic selected CNN features (via SFS) yield the highest performing prediction model for *valence* reported so far (average accuracy of 63.3%), which is significantly higher than the best model built on any selected set of ad-hoc features (best accuracy obtained with GFS selected features is 62.68%). On the other hand, *challenge* and *arousal* are predicted with significantly higher accuracies when using sets of ad-hoc features. The most accurate *arousal* model, built on 8 selected features, uses as inputs $\sigma\{IBAmp\}$ and $E\{IBAmp\}$ which are not captured by CNNs due to the imposed initial max-pooling layer, and $t_{\min}\{BV\}$, SC_{last} and D_t^{SC} which are hindered by the patch-wise normalization used. Similarly, the best *challenge* model uses four inputs two of which are $E\{IBAmp\}$ and $t_{\min}\{SC\}$. For the remaining affective states and dimensions, no significant differences are observed which suggests that the lack of certain aspects of the signals by CNNs were covered by information from other modalities.



(a) No feature selection



(b) Sequential feature selection



(c) Genetic feature selection

Figure 6.7: Physiological Fusion: average accuracy of ANNs trained on ad-hoc features from every modality (BVP/BV, HR and SC) normalized across the complete dataset (within dataset) and within each participant independently (within subject), and features generated by a triple of the CNNs ($CNN^{SC+BVP+HR-MB}_{20\times11+30\times45+30}$ and $CNN^{SC+BVP+HR-MB}_{80+45+15}$ for MB, and $CNN^{SC+BV+HR-DEAP}_{10\times15+20+30}$ and $CNN^{SC+BV+HR-DEAP}_{30+30+15}$ for DEAP). The black bar displayed on each average value represents the standard error (10 runs).

6.5 Summary

This chapter presented an empirical evaluation of convolutional auto-encoders as a method to extract features for affect prediction from physiological signals. We have applied this method to skin conductance, heart rate, blood volume pulse and blood volume and showed that, through a simple visualization mechanism, the extracted features are easily analyzed and capture well known affect features and other physiological patterns. Furthermore, these features yield predictive affect models of equal or greater accuracy than ad-hoc methods for a number of modalities and affective states present on the two datasets examined, suggesting that unsupervised feature learning can reveal affect-relevant information hindered by standard ad-hoc features. On the other hand, ad-hoc features outperformed automatically learned features on some states and modalities, pointing out two weaknesses of the proposed method, namely inaccurate synthesis of global information and difficulties for capturing amplitude information on periodic signals, outlining some clear directions for future work. Aside of these contributions, the results also served to showcase the importance of automatic feature selection and further validate neural network preference learning as a technique for affect modeling. In the next chapter a completely different feature extraction approach is evaluated, namely frequent sequence mining, used specifically to fuse physiological modalities of inputs with contextual information.

Chapter 7

Automatic Feature Extraction for Context and Physiology Fusion

In the previous chapter we showed how an unsupervised learning method can be used in place of ad-hoc features to reduce the dimensionality of the input signals of an affect model. Despite of years of research on psycho-physiology, results showed that physiological features learned automatically can yield more accurate models of affect than sets of standard ad-hoc features. This finding represents an improvement for psycho-physiology but more in general defines a new paradigm that can generate advances in other affect modeling scenarios, specially those in which the body of research around ad-hoc features is limited. One of these scenarios consists of modeling affect using *context* as an input modality. We refer to context as information about the system that the user is interacting with in the target application domain. Information within this context modality could be a mouse click on an advertisement while browsing the web, an actor that starts to cry on a movie scene or a player scoring a goal on a soccer video-game. The context modality depends on the specific system and thus generic ad-hoc features are not easily designed. Another scenario in which ad-hoc features have been scarcely studied consists of modeling affect using several different modalities; in fact, most studies only consider single-modality features which then are fed to the predictive model.

In this chapter we evaluate the efficacy of an unsupervised learning method on these two scenarios: to extract features that combine several modalities including context. We utilize the *generalized sequential patterns* sequence mining algorithm to find patterns across sequences of discrete events as those commonly found on a context modality (e.g. sequence of actions performed on the interface). Then, these patterns (frequent sequences) can be directly converted into features that feed an affect model. In particular, we apply GSP to mine frequent sequences across discrete events of skin conductance, blood volume pulse and game events (that provide a context for the physiological responses) within the Maze-Ball dataset (all the events are described in Chapter 4). The first section of this chapter presents a detailed analysis of the sequences extracted and discusses the effect of different parameters on the extraction algorithm. In the second section, two alternative methods for feature creation from frequent sequences are applied, and the resulting feature sets are compared to two variants of a set of ad-hoc features as inputs to predictors of the affective states reported in Maze-Ball. The ad-hoc features are described in Chapter 4 and include the physiological features used in the previous chapter together with a set of gameplay features. Similarly to the previous chapter, the analysis of the results covers the expressivity of the

unsupervised learned features and the relative performance between those features and an ad-hoc set; the thorough analysis of the affect models is left out as the focus of the chapter is on the method and not the Maze-Ball dataset.

7.1 Sequence Mining

As described in Chapter 3, the GSP algorithm features 3 tunable parameters: minimum support, maximum time window and minimum gap, denoted as S_{min}, W_{max} and G_{max}, respectively; this section discusses the effects of these parameters in the task of mining sequences from multimodal datasets and later analyzes the frequent sequences found in the Maze-Ball dataset.

7.1.1 Parameter Tuning

The S_{min} is set up to 100 sequences which forces a sequence pattern to occur in at least 44.64% of the samples (100 out of 224 samples in total) to be considered frequent. This high threshold is selected because of the low *specificity* of the events that yields a high frequency of each event within the data-sequences. For example, consider the event of picking any of the ten available pellets ($) and an alternative representation with ten different events associated to a specific pellet in the game $\{\$_0, \ldots, \$_9\}$. The support count for each $\$_x$ is expected to be lower than the support count of $ as different players will most likely pick different pellets. Consequently a lower minimum support would be required to consider sequence patterns containing the 10 specific $\$_x$ events frequent.

The W_{max} is chosen as a trade-off between the frequency of the events within and across signals. On one hand, a low window is required to not consider simultaneous events that within the same signal are clearly not occurring at the same time (e.g. in a 90 second-long game, two gameplay events of more than a few seconds apart should not be processed as two simultaneous events). On the other hand, a certain window must be allowed to consider events from different asynchronous signals to occur at the same moment in time. For example, a variation on the RR signal is linked to the time interval between two heart beats; a gameplay event happening during this interval should be considered simultaneous to the RR variation even though the time stamps of both events are not identical. In all experiments reported in this chapter W_{max} is 1 second.

The G_{max} parameter has a direct effect on the number of events that can be concatenated in a sequence and the support count of those sequences. Note that the definition of a sequence pattern does not require two consecutive events to occur immediately one after the other in the data-sequences. Consequently when G_{max} is much larger than the frequency of events in one modality, a high number of events in other modalities can be skipped when matching a pattern with a data-sequence. For example, if we select a G_{max} of 5 seconds and we consider the sequence $(\blacktriangleright)(\blacktriangleright)$, the number of data-sequences containing those two events in a 5 second interval is very high (arrow keys are pressed every 1-3 seconds) even though none of the players pressed the right arrow key twice in a row. This effect is enhanced by the fact that the target sequences combine events from different modalities. For instance, if the pattern $(\blacktriangleright)(s^{\Uparrow})(\blacktriangleright)$ is found with a 5 second G_{max}, this sequence is supported by any data-sequence in which a player pressed the right key in a 5 second interval before and after his SC raised — independently of other keys being pressed before and after $s^{\Uparrow}$ or other events such as $ and E occurring. This has not only an effect on the informative value of the sequences but also increases greatly the number of frequent patterns (see Table 7.1). On

# events	G_{max}		
	1.5	3	5
1	19	19	19
2	161	188	232
3	785	1650	2113
4	505	8387	18259
5	28	18985	118192
6	0	21704	545667
7	0	11251	NaN
8	0	1954	NaN
9	0	36	NaN

Table 7.1: Amount of frequent sequential patterns for different values of G_{max}. Search was stopped for $G_{max} = 5.0$ for lengths above 6 as the computational cost of finding every pattern became intractable.

the other hand, a large G_{max} may allow higher level sequences — i.e. sequences of high level events that do not occur very frequently — to emerge such as (\$)(\$)(\$). Thus, when setting up this parameter, it is necessary to take into account the frequency of each event and the desired target sequences. As this chapter focuses on the extraction of features that fuse physiology with context, G_{max} is set up to small values to capture the relationship between gameplay events and physiological responses. In particular, results with G_{max} of 1.5 and 3 seconds, which are both close to the frequency of the key presses (the most frequent event in the dataset), are reported. The two selected G_{max} values also approximate the physiological time responses to game events reported in several studies — e.g. see (Ravaja et al., 2005, 2006).

7.1.2 Sequence Analysis

We found a large amount of frequent sequences in the dataset; specifically, looking at the 3 and 5 second G_{max} values (see Table 7.1) one may observe the exponential increase of frequent sequential patterns with regards to the number of events considered. The large amount of patterns is caused, in part, by the right and left key press events which are so frequent in the data-sequences that result in almost *wild card* events (i.e. between any two events is very likely that either the right or the left key is pressed).

The more restrictive G_{max} of 1.5 seconds did not produce any frequent sequence combining the sector events with the key presses making more difficult, if not impossible, to map a sequence of key presses to an area in the maze (e.g. (◄)(►)(►)(◄)(◄)). Even though specific paths cannot be inferred from the key presses, some interesting results can be observed. For 1.5 and 3 second G_{max} values the sequences of maximum length found consist only of the events {◄, ►, $s^{\Uparrow}$, $s^{\Downarrow}$}; neither other gameplay events nor RR variations are included in the most frequent sequences (see Table 7.2). These sequences suggest that the combination of many key press patterns in the Maze-Ball game and SC variation are frequent and should be considered in the analysis of player experience. Furthermore, in the longest sequences that combine key presses with the events {E, \$} (see Table 7.2), the latter always appear in the last position of the sequences suggesting that players follow similar strategies to approach pellets and enemies but more dissimilar behaviors are presented after picking a pellet or being hit by an enemy.

Sequences	G_{max}	
	1.5	3
$(s^{\Uparrow})(s^{\Downarrow})(\blacktriangleright)(\blacktriangleleft)(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)(\blacktriangleleft)$	< 100	108
$(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)$	< 100	108
$(\blacktriangleleft)(s^{\Uparrow})(s^{\Downarrow})(s^{\Uparrow})(s^{\Downarrow})(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)(\blacktriangleleft)$	< 100	101
$(\blacktriangleright)(\blacktriangleleft)(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)(\blacktriangleleft)(\$)$	< 100	104
$(\blacktriangleleft)(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)(\blacktriangleleft)(\$)(\blacktriangleleft)$	< 100	104
$(\blacktriangleleft)(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)(\$)$	< 100	149
$(\blacktriangleright)(\blacktriangleleft)(m^{7})(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)$	< 100	105
$(\blacktriangleleft)(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)(E)$	< 100	104
$(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(E)(\blacktriangleright)(\blacktriangleleft)$	< 100	102
$(\blacktriangleleft)(s^{\Uparrow})(s^{\Downarrow})(\blacktriangleright)(E)(\blacktriangleleft)$	< 100	101
$(\blacktriangleleft)(\blacktriangleleft)(\blacktriangleright)(\blacktriangleright)(\blacktriangleleft)$	115	202
$(\blacktriangleright)(s^{\Uparrow})(\blacktriangleright)(s^{\Downarrow})(\blacktriangleleft)$	106	192
$(s^{\Uparrow})(\blacktriangleright)(s^{\Downarrow})(\blacktriangleright)(\blacktriangleleft)$	101	187

Table 7.2: Support counts of a subset of frequent sequences containing keyboard events.

Table 7.3 shows some of the most frequent 2-sequences and 3-sequences that combine the main performance events, \$ and E, with physiological events. With the more restrictive value for G_{max} (1.5 second), all the frequent 3-sequences contain the subsequence ($s^{\Uparrow}$)($s^{\Downarrow}$) with the event \$ or E in any position, being the most frequent sequence in which the gameplay event occurs simultaneously with an increase of SC, followed by a decrease of SC. The 3 second G_{max} on the other hand, produced almost any combination of two physiological events with one of the gameplay events. This might indicate that the threshold is too large and does not capture a meaningful fusion of the modalities.

The frequent 2-sequences correspond to all possible combinations of one of the gameplay events with one physiological event showing more occurrences when combined in the same element. Note that the count support indicates only the number of data-sequences in which the sequence pattern appears and not the number of occurrences within each data-sequence. Opposite to long sequences, these short patterns are expected to occur more than once in each sequence. Therefore, the number of occurrences of the patterns within each data-sequence is required for a full analysis of the physiological responses to game events. This study, however, is too detailed for this chapter as the focus is on the method rather than the psycho-physiological models of Maze-Ball players.

It is worth mentioning that none of the other high level gameplay events occur frequently with the physiological responses. While this is not entirely surprising for the sector events — given that there is no visual or audio feedback when changing sectors in the maze — one would expect that the count down event would trigger a change on the player that would have been reflected on her physiology. However, such a relationship is not observed frequently via event sequences.

7.2 Affect Modeling with Frequent Sequences

For the affect modeling experiments reported here, the frequent sequences included were found using $G_{max} = 1.5$, $S_{min} = 100$ and $W_{max} = 1$ that did not contain the keyboard and *Stop* events. This last constrain reduces substantially the number of frequent patterns from

3-sequences	G_{max}		2-sequences	G_{max}	
	1.5	3		1.5	3
($ $s^{\Uparrow}$)($s^{\Downarrow}$)	141	168	($ $s^{\Uparrow}$)	184	184
(E $s^{\Uparrow}$)($s^{\Downarrow}$)	131	163	($ r^{-50})	178	178
($s^{\Uparrow}$)($) ($s^{\Downarrow}$)	123	164	(E $s^{\Uparrow}$)	175	175
($s^{\Uparrow}$)(E)($s^{\Downarrow}$)	116	164	(E $s^{\Downarrow}$)	174	174
($s^{\Uparrow}$)($s^{\Downarrow}$)($)	112	181	(E r^{+50})	174	174
(E)($s^{\Uparrow}$)($s^{\Downarrow}$)	109	175	($ $s^{\Downarrow}$)	170	170
($s^{\Uparrow}$)($s^{\Downarrow}$)(E)	106	180	($s^{\Uparrow}$)($)	166	194
($)($s^{\Uparrow}$)($s^{\Downarrow}$)	105	186			
($s^{\Uparrow}$)($ $s^{\Downarrow}$)	105	158			
($s^{\Uparrow}$ $s^{\Downarrow}$)($)	102	139			

Table 7.3: Support counts of a subset of the most frequent sequences including physiological and performance events. Events enclosed in the same parentheses occur simultaneously (in any order within an interval of 1 second).

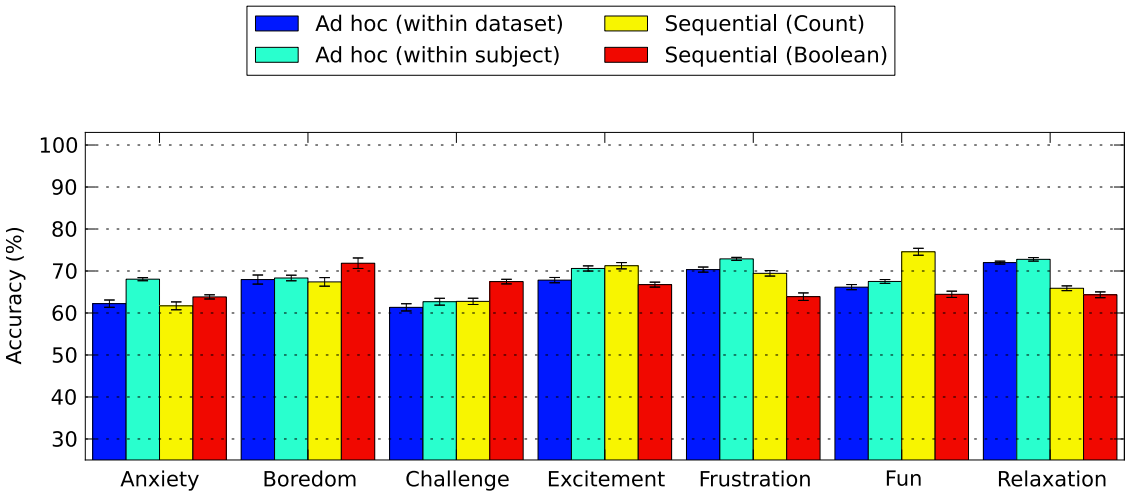
1498 to 140 in an effort to lower the dimensionality of the input space and the computational cost of training. Preliminary studies showed that using the full set of frequent patterns did not improve the prediction accuracy of the models of affect. Models for the seven states in Maze-ball are created based on two representations of these sequential patterns, namely *boolean* and *count*. With the count representation, the features for each data-sequence (game played) correspond to the number of occurrences of each frequent sequence within that sample (values normalized between 0 and 1 across the complete set); with the boolean representation every feature with a value above 0 is converted to 1 (each feature is a detector of whether a particular frequent sequence occurs at least once within the data-sequence). Alternatively, models are trained using a set of ad-hoc features that combines physiological data — 15 features extracted from each signal (heart rate, skin conductance and blood volume pulse) — and gameplay metrics — 31 features extracted.

Analogously to the methodology used in the previous chapter, for each set of features and particular affective state, sequential forward and genetic search feature selection run 10 times using the methods that showed more promise in Chapter 5, and the average 3-fold cross-validation accuracy (percentage of correctly classified pairs) across the 10 resulting feature subsets is reported. The average 3-fold cross-validation accuracy of 10 models trained on the complete sets (without feature selection) of features is also reported. T-tests are used to assess the significance of accuracy differences among pairs of experiments; significant results are considered with p-values below 0.05.

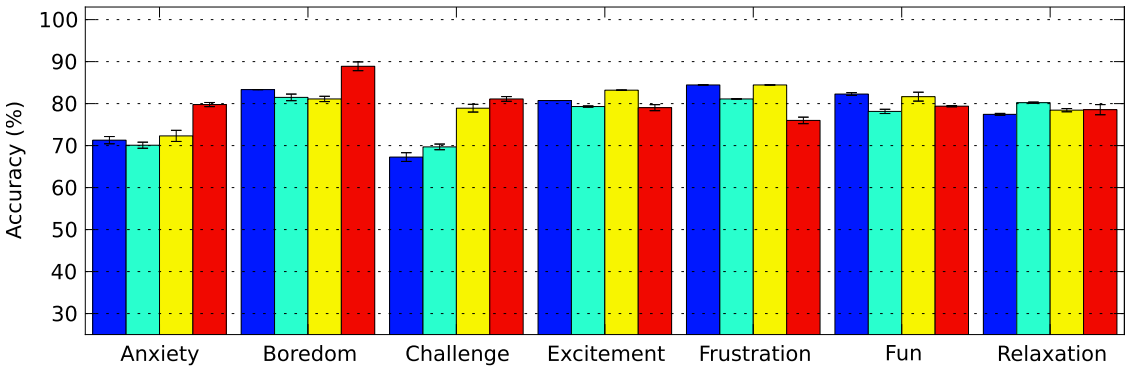
The training algorithm (neuroevolution or backpropagation), error function (regularized least squared with margin equal to 1.0 or sigmoid rank margin with margin equal to 0.01) and neural network topology are systematically adjusted independently for each condition (target affective state, feature selection method and feature extraction scheme). The remaining parameters are either fixed or adjusted systematically following the same procedure as in previous chapters.

7.2.1 Sequences Input to User Preference Models

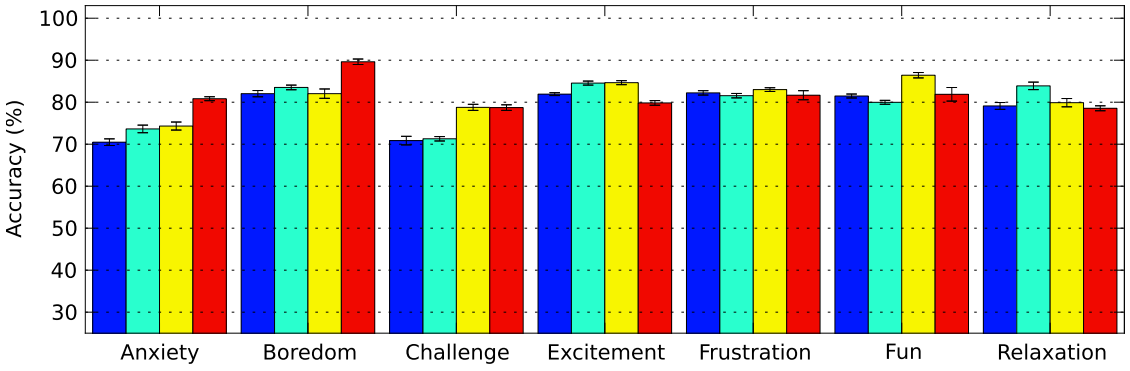
For the dataset examined, in which the frequent patterns are short and expected to occur a variant number of times across samples, the count representation is expected to yield



(a) All features



(b) Features selected via SFS



(c) Features selected via GFS

Figure 7.1: Sequential vs ad-hoc features: average accuracy of ANNs trained on ad-hoc features and features extracted from GSP sequences as real (Count) or binary (Boolean) features. The black bar displayed on each average value represents the standard error (10 runs).

			Anx	Brd	Chl	Exc	Frs	Fun	Rlx
SFS	Sequential	Count	4.9	5.8	6.3	5	11	6.5	6.3
		Boolean	9.4	9.3	9.5	6.5	10	11.3	8.3
	Ad-hoc	Dataset	4	3	2.1	6	7	5.8	5.6
		Subject	3.3	3.9	3.9	4.8	5	4.8	7.2
GFS	Sequential	Count	18.6	13.8	16	22.5	18.2	27.3	16.8
		Boolean	25.3	25.8	24	24.2	25.9	25.8	26
	Ad-hoc	Dataset	8.7	12.1	9.5	11.2	14.4	10.8	18.3
		Subject	12.4	12.7	13.8	17.3	11.6	12.1	21.1

Table 7.4: Number of selected features. The average number of features selected across 10 runs of SFS and GFS for each of the affective states in MB — *anxiety* (Anx), *boredom* (Brd), *challenge* (Chl), *excitement* (Exc), *furstration* (Frs), *fun* (Fun) and *relaxation* (Rlx) — and four sets of features: sequential with boolean and count representation (140 features in total) and ad-hoc with within dataset and within subject normalization (76 features in total).

higher accuracies since the boolean representation is far less informative. Models trained in the complete sets of features show that this hypothesis is valid for the states of *fun*, *excitement* and *frustration* while boolean features yield significantly higher accuracies than count features for *anxiety*, *boredom* and *challenge* (see Figure 7.1a). When SFS and GFS are applied, the same results are observed with the exception of several differences becoming insignificant — for *challenge* (both GFS and SFS), *frustration* (GFS) and *fun* (SFS).

Models of affect were trained on pairs of data, as boolean features collapse a range of values into 1, a larger number of pairs present the same value for each feature when the boolean representation is used, yielding to a lower differentiation than features based on the count representation — count features show a different value between the samples of each pairwise report on 76.77% of the pairs on average (standard deviation equals to 11.1) while with boolean features this average percentage drops to 36% (standard deviation of 11.57). Nevertheless, models built on boolean features can learn to differentiate preferences with comparable accuracies as count features by using a larger number of inputs (see Table 7.4). Furthermore, by collapsing every value above 0 into 1, more pronounced differences are observed than with features normalized by the maximum number of occurrences (e.g. a pair that presents values 0.0 and 0.1 for a given count feature, yields, respectively, 0.0 and 1.0 the boolean version). This form of aggressive outlier removal, could explain the higher accuracies for *anxiety* and *boredom* models.

7.2.2 Comparison Between Sequential and Ad-hoc Features

Feature sets based on frequent sequence patterns, when reduced by automatic feature selection, yield equally or more accurate predictors of affect than ad-hoc features in all the states investigated with the exception of *relaxation* (see Figure 7.1b and Figure 7.1c). In particular, with the *count* representation, the best selected models yield higher accuracies for the states of *fun*, and *challenge*. With the boolean representation, sequential features also yield significantly more accurate models than ad-hoc features for *anxiety* and *boredom*. Models for *frustration* and *excitement* do not present significant differences while *relaxation* is predicted more accurately with ad-hoc features. While ad-hoc features contain much more information about physiology (e.g. maximum heart rate and the high frequency com-

ponent of RR intervals) and gameplay (e.g average distance to enemies and percentage of map explored), information about discrete events occurring in short time intervals provides equivalent or more valuable information about the user experience for the prediction of all self-reports of affect except *relaxation*. One could argue that a greater number of ad-hoc gameplay features could have been created reducing for instance information about the trajectory followed by the player (which is present in the sequential features through the section events); however, ad-hoc gameplay features were carefully designed when MB was created and the parts of the maze were not considered relevant (Martínez et al., 2010). Note that any information captured by automatically learned features can be ultimately coded as an ad-hoc feature forcing sequential features to yield performances equal or lower than ad-hoc features; however, the main advantage of these automatic feature extraction methods is that they can find features that could have been disregarded by human designers, as it was the case with Maze-Ball trajectory information.

If feature selection is not applied, *frustration* and *anxiety* (in addition to *relaxation*) models yield higher accuracies with ad-hoc features (see Figure 7.1a). While sequential features seem to not provide additional information for the prediction of *frustration* and *relaxation*, *anxiety* models on selected sequential features outperform ad-hoc features; hence, it appears that the larger total number of features (140 sequential features vs. 76 ad-hoc features), unsurprisingly, can have a negative effect on training. Nevertheless, count features for *fun* and boolean features for *challenge* and *boredom* still maintain a significant improvement over ad-hoc features despite the larger set of features, highlighting the efficacy of unsupervised feature extraction on creating relevant input features.

7.2.3 Expressivity of Sequential Features

This section exemplifies the expressivity of models that rely on sequential features. Table 7.5 depicts a model built on boolean sequential features (best *anxiety* model) and a model built on count sequential features (best *fun* model), corresponding to the highest accuracies found using single-layer perceptrons. As both models are single-layer perceptrons, we can very easily analyze the effect of each feature by examining the connection weights (features with low absolute connection weight are omitted to better highlight the relevant effects).

The *anxiety* model combines 2 multimodal features and 2 physiological features. The sequences (r^{-50})(\$) and (E)($s^{\Uparrow}$) associated with high negative weights suggest that games in which an increment on SC is never detected after being hit by an enemy or an increment on HR — connected to the decrement of RR intervals — is never detected right before collecting a pellet are related to more *anxious* experiences. This could be explained by an overall heightened level of SC and HR during an anxious level leading to less noticeable or non-existent reactions to the most relevant gameplay events. Along the same interpretation, an increased variability of sympathetic activity in absence of gameplay events would indicate a more anxious experience, as suggested by the positive connections with ($s^{\Uparrow}$ $s^{\Downarrow}$)($s^{\Uparrow}$) and ($s^{\Uparrow}$ r^{+50}). The changes could be caused by difficulty of approaching pellets or changes on the camera perspective which are not represented in the sequence events.

Observing the best-performing ANN model of *fun* (see Table 7.5) it seems that the total number of pellets picked (\$) has a positive impact on reported fun as well as when a sudden peak in SC is generated just before or just after picking a pellet — ($s^{\Uparrow}$)($s^{\Downarrow}$)(\$) and (\$ $s^{\Uparrow}$)($s^{\Downarrow}$) — which could be related to a heightened arousal state when the player is about to pick a pellet. Sudden increases on the RR intervals length followed by a sudden change on SC — (r^{+50}) ($s^{\Uparrow}$) and (r^{+50})($s^{\Downarrow}$) — have a negative impact on reported fun attenuated

Anxiety		Fun	
(r^{-50})(\$)	-0.27	(E $s^{\Downarrow}$)	-0.31
(E)($s^{\Uparrow}$)	-0.15	(\$ $s^{\Uparrow}$)($s^{\Downarrow}$)	0.18
($s^{\Uparrow}$ r^{+50})	0.24	($s^{\Uparrow}$)($s^{\Downarrow}$)(\$)	0.19
($s^{\Uparrow}$ $s^{\Downarrow}$)($s^{\Uparrow}$)	0.19	(\$)	0.25
		(r^{+50})($s^{\Uparrow}$)	-0.19
		(r^{+50})($s^{\Downarrow}$)	-0.19
		(r^{+50})($s^{\Uparrow}$)(r^{-50})	0.17
		(m_5)	-0.17

Table 7.5: Input features and corresponding connection weight values for the highest performing ANN models for *anxiety* and *fun*.

when followed by a sudden increase on RR — (r^{+50}) ($s^{\Uparrow}$) (r^{-50}). Additionally, enemy hits accompanied by a sudden decrease on SC (E $s^{\Downarrow}$), not surprisingly, seem to decrease the level of fun. Note, that a single event E would not necessarily have a negative effect on fun as it is a fundamental part of the game; however, a decrease on SC might indicate a lowered level of the player's arousal as consequence of the game event. Finally, it appears that exploring the 5^{th} sector of the maze contributes to less *fun* experiences. As shown by these two models, sequential frequent patterns used as boolean or real-valued inputs for models of affect can reveal relevant elements of affective experiences obscured by standard ad-hoc features, especially across modalities of user input.

7.3 Summary

This chapter evaluates sequential pattern mining as a method to extract automatically multimodal features. The GSP sequence pattern mining algorithm was applied to physiological signals (blood volume pulse and skin conductance) and context-based metrics (e.g. in-game events and key board presses) in a game dataset. The extracted frequent sequences and algorithm parameters were analyzed to highlight the main advantages, limitations and practical considerations of this approach when applied to the analysis of multimodal interactions.

Additionally, as an alternative to ad-hoc features, the frequent sequences mined were presented as inputs of affect detectors assisting the process of finding more accurate models of user affect and experience. With automatic feature selection, sequences outperform ad-hoc features in four (*anxiety*, *boredom*, *challenge*, and *fun*) out of the seven affective states examined in this study, yield similar accuracies in other two states (*frustration* and *excitement*), and showcase slightly lower accuracies for *relaxation*. Furthermore, an analysis of the models of affect trained on the frequent sequences reveals relationships between sudden arousal level changes across physiological signals, critical game events and reported affect, which demonstrates the expressive power of this feature extraction mechanism.

Overall, despite its simplicity, sequence mining shows a great potential for automatically extracting features for modeling affect from discrete (or discretized) input modalities. Despite the discretization of physiological signals discards interesting information such as heart rate variability, the results show that these simple events in combination with game information suffice to yield multimodal features of great prediction power. In the final chapter, the main findings and contributions of this thesis are summarized, the limitations of the method and evaluation are discussed and the directions for future work are outlined.

Chapter 8

Conclusions

This thesis proposed a methodology for enhancing current affect modeling practices. The methods that integrate this methodology are aligned to the three research questions outlined in Chapter 1: 1) *what are the most adequate preference learning methods to train models of affect based on ordinal annotations of affect*, 2) *can automatic feature extraction capture physiological components that are more relevant for prediction of affect than popular ad-hoc designed features* and 3) *can automatic feature extraction capture the interplay across modalities of input and produce more accurate models of affect compared to ad-hoc features.*

Towards answering the first research question, we analyze the performance of a collection of artificial neural network training algorithms as well as support vector machines and Cohen's method. Several synthetic datasets are generated to test the performance of the different algorithms with respect to the complexity of the target function (i.e. the mapping between physiology and affective state in psycho-physiological modeling). Within those synthetic datasets, two key patterns — common in physiological and affect datasets — are investigated: input data distributed around separate clusters (to resemble physiological data from several users with dissimilar baselines) and pairs of data with different degrees of preference (to simulate self-reports over pairs of experiences that elicit very similar affective states). The algorithms are also compared across two affect datasets containing a number of fixed physiological features that are used as inputs of the models, and pairwise self-reports and ratings of several affective states that are used as target outputs. Additionally, a detailed evaluation of the impact of the error function and training algorithm in ANN prediction accuracy is presented. In particular, the two training algorithms for neural networks dominant in different artificial intelligence communities are evaluated, namely backpropagation from machine learning studies and neuroevolution from computational intelligence studies; both algorithms are compared across a collection of error functions proposed in the related literature — rank-margin, cross-entropy and regularized least-squares from ML research, and sigmoid from CI and affective computing studies — and also introduced here — sigmoidal rank-margin and Spearman — to cover characteristics not explored before.

In order to answer the second research question, a method based on recent advances in deep learning (Bengio et al., 2007) is proposed as an automatic feature extractor and compared to an extensive set of ad-hoc features which are commonly (and traditionally used) in AC research. Specifically, convolutional neural networks are trained using denoising auto-enconders to extract features from four physiological signals (HR, SC, BV and BVP). These features are compared against ad-hoc signal-specific features as inputs to ANN-based models trained to predict pairwise self-reports in two affect datasets.

An answer to the third research question was sought by introducing a frequent sequence mining method borrowed from data mining studies, as a technique for multimodal feature extraction. In particular, the generalized sequence patterns algorithm is applied to extract features from temporal sequences, that fuse interaction events (game context) and physiological responses, in an affect game-based dataset. The automatically generated features are compared against ad-hoc features as inputs to affect models.

Results obtained from the experiments presented in this dissertation suggest that more accurate models of affect can emerge from the combination of ANNs for preference learning with automatic feature extraction techniques, both for physiological signals and multimodal fusion. First, ANNs outperform SVMs in most datasets examined and CMs in all of them. Considering that ANNs also require a lower computational effort and that they present a higher expressivity, it is suggested that they conform a more suited method for affect preference modeling. In addition, within ANN variants experiments on synthetic data showed that NE is able to train more accurate ANNs than BP especially when the data is not uniformly distributed, either because the input samples present variable baselines (clustered data) or unclear preferences are present in the dataset. For NE, the error functions that yield more accurate models do not depend on the ANN's output for sample pairs classified correctly, being the sigmoidal rank-margin the error function that yields the best results from the examined error functions. On the other hand, BP requires error functions that depend on correctly classified pairs, being regularized least-square the examined function that yields best results, as it defines the most appropriate balance between maximizing the difference between ANN's output of samples in correctly classified pairs, and correcting the outputs of samples in incorrectly classified pairs. Second, CNNs yield models that outperform ad-hoc features for several input modalities and several affective states; furthermore, limitations revealed in the experiments suggest that alternative configurations would show larger benefits of automatic feature extraction. Third, results demonstrated that SM, and in general automatic feature extraction, can reveal relationships across sudden arousal level changes in physiological signals, critical game events and reported affect, and that those relations can guide the construction of more accurate predictors of affect compared to standard ad-hoc physiological features and game metrics.

8.1 Contributions

This section summarizes the contributions of this thesis which advance affect modeling practices in AC and HCI. The main achievements also contribute to the field of AI and games and technology enhanced learning technologies that incorporate user modeling components (such as intelligent tutoring systems and game-based learning approaches). Finally, contributions have also been made to the fields of machine and preference learning with new validations of the strengths and weaknesses of several supervised and unsupervised learning techniques. More specifically, this thesis has contributed the following:

- Evaluation of a generic methodology for creating models of affect based on physiological and contextual information from ordinal annotations. Reduction of input signals through feature extraction and selection, and model training are performed by automatic algorithms, reducing information loss and biases introduced by human modelers.

- Extensive empirical testing of a set of artificial neural network training algorithms,

support vector machines and Cohen's method for pairwise preferences.

- Introduction of deep learning for the construction of accurate models of affect based on physiological manifestations of affect. The method is tested in two affect datasets and a number of recommendations are suggested for its application in affect modeling at large.
- Introduction of sequential pattern mining as a method for exploring the relationship between asynchronous signals from different modalities. Sequence mining has demonstrated its ability in extracting relevant multimodal features that are critical for accurate affect detection.
- Confirmation of the importance of automatic feature selection in affect modeling.

8.2 Limitations

This section outlines the key limitations and drawbacks of the methodology proposed and the evaluation presented. The main limitations can be traced back to the difficulty of collecting large and reliable affect datasets, which are common limitations to any affect modeling methodology that relies on user data. The remaining limitations — as the results revealed — arise from the application of the new methods introduced in this thesis.

8.2.1 Data-driven Affect Modeling

Method Evaluation

While the Maze-Ball and DEAP datasets include key components for affect modeling and are representative of a typical affect modeling scenario, the methodology presented needs to be tested on diverse datasets. Moreover, to be able to demonstrate robustness of the algorithms, more and dissimilar modalities of user input need to be considered, and different domains (beyond games and music videos) need to be explored. The accuracies obtained across different affective states and modalities of user input, however, already provide sufficient evidence that the methodology would generalize well in dissimilar domains and modalities. Although low accuracies are obtained in the prediction of particular affective states, specially within the DEAP dataset, the proposed methodology is promising as all the results are compared against well-established and validated methods that do not achieve higher accuracies.

Model Generality

A limitation inherent to data-driven modeling approaches is that the quality of a learned model is subject to the quality of the data used to train it. If emotion elicitation fails but users still report experiencing specific affective states, noise is introduced into the dataset which may drive data-driven methods towards flawed or suboptimal models of affect as shown in Chapter 5. Several mechanisms are used in this thesis to reduce the number of incorrect reports and minimize their effect on training. Maze-Ball experimental survey featured a 4-alternative choice questionnaire that offers experiment participants two explicit options to report unclear differences in affective states, which then were dropped before training the models of affect. DEAP, in turn, implements regular ratings (participants rate each video after watching it) but two considerations were taken when converting them into

pairwise preferences. First, only consecutive reports were converted into preferences as it is expected that relative ratings across non-consecutive videos are unreliable — as consequence of subjective scales rapidly becoming inconsistent along time. Second, rating differences below 1 point (in a real-valued 7-point scale) are considered unclear and dropped before training the models. Then a *regularizer* is included in the training algorithms to enforce *smooth* models (i.e. small changes in the inputs yield to small changes in the outputs) that potentially do not learn spurious and incorrect patterns. Finally, cross-validation is used to estimate the generality of the models by calculating their prediction accuracy over data not used during training. However, note that parameter tuning and feature selection are guided by cross-validation accuracy which negatively affects the generality of the parameters used, features selected and models.

Even though we consider all the datasets used in this dissertation to be representative of the problem under examination, ideally, model generality can only be adequately evaluated in completely different datasets from the one used for training (as in Martínez et al., 2011). Alternatively, large datasets would improve the quality of the models by providing a population sample that better represents the complete population and reduces the number of patterns appearing by chance.

Ambiguity of physiology and additional components of affect

It has been suggested that certain dissimilar affective states may be linked to the same physiological responses (Cacioppo and Tassinary, 1990) which would prevent the creation of computational predictors that can differentiate among them while relying solely on physiological data. This theory would explain, in part, the relative low accuracies of the physiological models presented in Chapter 6. This limitation is alleviated by learning models that fuse context information and physiology, which capture more complex and meaningful affective relations (Barrett et al., 2007) and present higher prediction accuracies as physiological patterns are disambiguated by context information. However, it is not absurd to expect that other factors — not manifested through physiology or context — could also influence affective states generating other effects and biases. For instance, personality could determine different affective responses even though the context and the physiological responses appear similar (although some links between personality and physiology have been suggested (Kagan et al., 1987)). We can, in part, overcome such limitations by including additional information (e.g. personality, moods, preferences, gender) as inputs to models of affect, thereby, enhancing the expressivity and completeness of the models and improving the accuracy of the predictions. The methodology introduced in this thesis could potentially create models with such additional information and help towards investigating its relevancy or, contrarily, redundancy. However, those investigations would require a larger dataset containing a population sample representative of the variance of all the investigated factors.

Post-experience reports and comparative reports

The models of affect constructed across this thesis are trained on quantitative post-experience self-reports; thus, models depend on the ability of the experiment participants to remember their affective experiences and express them through a preference (MB) or a rating (DEAP) report. According to different theoretical models of emotional self-report, when humans report past emotional experiences they have to retrieve specific thoughts, event-specific details or beliefs that relate to the past experience (Robinson and Clore, 2002; Ross, 1989; Chris-

tianson and Safer, 1996; Kahneman, 1999). From these theories follows that the self-reports may be related only to certain parts of the experience monitored and, in turn, those parts of the experience are more relevant for the prediction of the reports. Martínez and Yannakakis (2011a) have shown that by calculating ad-hoc physiological features on particular intervals of the experience can increment the prediction accuracies for certain affective states. CNN may implicitly help improving the accuracy of the models by finding a set of patterns at different time intervals of the experience; SM may also exploit this characteristic implicitly by providing features that are not tied to particular moments of the experience. Thus, it appears that both methods proposed can extract information from the specific events that motivate the self-reports. However, a problem remains related to the user not recalling experiences of particular affective states that were not linked to specific events or thoughts. In comparative reports, especially, two separate experiences must be remembered, although an implicitly similar process could happen with single ratings if users attempt to maintain a consistent scale. The length of the experiences was kept short (90 seconds in MB and 60 seconds in DEAP) to minimize this problem but limitations derived from memory can only be completely bypassed by reporting affective states while they are being felt (e.g. think-aloud protocols) which on the other hand disrupts the experience. In addition to problems induced by recall of emotion, other limitations to self-reports exist such as *primacy* and *recency* effects related to the experiment order (Yannakakis and Hallam, 2011) (i.e. the participant having a tendency towards the first or last experience, respectively) or to the order of the items or direction of the scale within the questionnaire (Chan, 1991) (i.e. selecting the first or last item of a questionnaire, respectively). Alternatively, self-reports can be replaced by expert annotators that can pinpoint the exact time interval during which the affective state is displayed. This method, however, relies more on the user displaying clearly their affective state (and it is also more expensive and time-consuming). In summary, affect annotation is arguably the toughest challenge in affect modeling as there exists no method that is completely trustworthy and reliable.

Loss of information from ratings to preferences

In Chapter 3 we argued that rating reports contain ordinal information as the distance between items on the scale is unknown and variable (as it is subjective). Furthermore, in order to reduce biases due to the variability of the scale along time and across participants, we converted pairs of consecutive ratings into pairwise preferences. While this transformation is likely to reduce the amount of noise introduced in the dataset, it also leads to a loss of information related to the intensity of the emotion. Note that if for instance, a participant rates three consecutive experiences as $\{1, 4, 5\}$, these ratings yield the same order (and pairwise preferences) as $\{1, 2, 5\}$; however, the second experience presents a larger intensity in the first set of ratings than in the second. Alternative preference learning methods can be used to introduce this information into the learning process, as for instance through an extension of the regularized least-squares error function (Pahikkala et al., 2009). On the other hand, it is not certain whether introducing that discarded information in the modeling process yields more precise affect models or yields flawed models due to reporting biases.

8.2.2 Automatic Feature Extraction

Parameter tuning and training time

While DL and SM can automatically provide a more complete and appropriate set of features for certain affective states, ad-hoc features are *off-the-shelf* solutions that do not require training or tuning for well-studied signals; however, for new signals such as actions or events in a new game, new features must be hand-crafted. For SM the number of parameters is small (3) and can be selected taking into account few considerations (see Chapter 7 for more details) although the computational cost for finding all frequent patterns can be high when the input signals are *dense*. DL generally requires a lower training time but the number of parameters is larger and tuning demands more expert knowledge. This thesis introduces a number of CNN topologies that performed well on the SC and HR signals and to a lesser extend on the BV and BVP signals. Regarding the similarities in the features learned between datasets, it is expected that similar configurations would yield good performances in alternative datasets reducing the effort of parameter tuning. Alternatively, these methods can be applied to create new ad-hoc features — i.e. they could be trained once on a large dataset and the features then reused in further datasets.

Discretization of signals for frequent sequence mining

While DN is applicable to continuous signals — i.e. temporal sequences of real values such as physiological signals — and to discrete signals — i.e. temporal sequences of discrete labels such as interaction events (e.g. clicking a button) — a limitation of SM is that it can only handle discrete signals. The *discretization* of continuous signals involves information loss and it may require a designer to choose the important components. In this thesis, two discretization methods were utilized to generate simple events from heart rate variability and skin conductance, respectively. The resulting multimodal features that combined these events and contextual information yield high prediction accuracies despite of large amount of physiological data lost in the process; however, it is expected that alternative methods (hand-crafted or learned) able to inject a larger amount of information into the discretized signal could yield better results.

Global patterns and signal amplitude in convolutional networks

Two limitations emerged from the experiments reported in Chapter 6 regarding CNN capabilities: the topologies used in this thesis could not capture global information (e.g. time when minimum value is recorded) and could not properly represent the peak-to-peak amplitude of periodic signals such as BVP. The first limitation appears to be caused by the patchwise normalization of the approach; thus it is expected that using alternative normalization schemes can eliminate this problem. The second limitation however, seems to arise from an incomplete exploration of the possible topologies as, in most cases, the periodic nature of the signals (BVP and BV) was removed (via initial pooling layers). It is thus expected that alternative topologies can capture these components overcoming this problem as well. Alternatively, related studies have made use of Fourier transformation to extract the frequency components and feed them directly into the CNN (e.g. Hamel et al., 2011); such an approach can potentially yield good results for periodic physiological signals as well.

Expressivity of deep learned features

An advantage of ad-hoc statistical features resides in the simplicity to interpret the physical properties of the signal as they are usually based on simple statistical metrics. Therefore, prediction models trained on ad-hoc features can be analysed with low effort providing insights in affective phenomena. Artificial neural networks have traditionally been considered as *black boxes* that oppose their high prediction power to a more difficult interpretation of what has been learned by the model. Results presented in Chapter 6 have shown, however, that appropriate visualization tools can ease the interpretation of neural-network based features. Moreover, learned features derived from DL architectures may define data-based extracted patterns, which could lead to the advancement of our understanding of emotion manifestations via physiology (and beyond).

8.3 Extensibility

The methodology introduced and evaluated throughout this thesis showed great promise in two affect datasets. This section reviews additional domains where the same methodology is expected to perform reliably and a set of extensions for the methods that can lead to the creation of more accurate and reliable models of affect.

8.3.1 Applicability to Other Domains

Given that the computational methods used directly are generic, it is expected that the methodology can be used to train reliable and accurate affect models beyond the experimental datasets used here.

Automatic feature extraction beyond physiology and game context

As it has been mentioned before, CNNs can process any continuous signal. While this thesis applied CNNs on heart rate, skin conductance, blood volume pulse and blood volume, the method is directly applicable for affect detection in single-dimensional, time-series, input signals such as electroencephalograph (EEG), electromyograph (EMG), skin temperature (ST), respiration rate (RSP) and speech, but also in two-dimensional input signals such as images (Rifai et al., 2012; Asteriadis et al., 2009) (e.g. for facial expression and head pose analysis). It has been also discussed that CNNs could process discrete events if converted to a continuous function which also opens up the possibility of extracting contextual features via DL and even multimodal features by using neurons that collect inputs across modalities.

With appropriate mechanisms for discretization, SM is expected to yield excellent results with additional 1-dimensional continuous signals. The discretization mechanisms used for SC and BVP could be tuned to extract events from ST and EMG, and RSP, respectively, as the characteristics of the signals are similar. The method is also directly applicable to facial expression and body movement via *action units* which are discrete events typically used for the recognition of emotion from these modalities (Ekman and Friesen, 1978 among others).

Affect modeling beyond games and films

The key findings of the thesis indicate that the affect modeling methodology proposed is independent of the user activity and the system used to elicit the affective states. Thus,

the method is directly applicable to any real-time human computer interaction sub-domain (involving physiological signals or any type of time-series user input modality).

Affect modeling beyond self-reports

Although all the user experiments presented in this thesis contain post-experience self-reports, the methods are directly applicable to any other form of reports that can be represented ultimately as comparisons, even if the compared experiences last for different time lengths (e.g. an external observer could annotate frustration episodes of variable length and rank them by intensity). Pooling layers of a CNN can be adjusted to yield the same number of features independently of changes to the length of the raw signals. SM features are also applicable to variable-length experiences without any adjustments. Furthermore, with small modifications — i.e. use of different error functions such as sum of square errors (Widrow and Lehr, 1990) — the methodology proposed can be applied to affect classification (i.e. prediction of discrete emotions) and regression (i.e. prediction of target affect intensities) tasks across any type of input signal.

8.3.2 Method Extensions

This dissertation introduced automatic feature extraction to affect modeling from physiology and context, and surveyed several modeling methods. The evaluation covers the basic components of deep learning and frequent sequence mining, and three modeling methods but a different number of research directions can extend this work.

Alternative modeling methods and intransitive preferences

We presented an extensive study of three computational methods for preference learning to figure out which techniques are best for affect modeling. However, additional methods have to be added to the comparison to completely answer this question. On one research direction, further standard methods that solve the preference learning problem by approximating a global utility function can be explored; candidates popular on different research areas are Bayesian networks and decision trees. On a different direction, the approach defined by Cohen's method — i.e. two data samples are fed simultaneously to the model and the probability of the first being preferred is approximated — has to be examined more carefully. In this thesis, only synthetic transitive preferences where used for testing while Cohen's method is likely to stand out when learning intransitive preferences. Furthermore, any standard classification method (e.g. decision trees) can be directly applied to predict preferences in this manner, by using two samples as input to the model and one binary class as output (first sample is preferred or second sample is preferred).

Deep models of affect

Deep learning was used to automatically extract features that could be fed to any computational model (e.g. support vector machine). Choosing an artificial neural network as the representation of the model, creates the opportunity of further refining the learned features through supervised learning (i.e. *fine-tuning* the model (Bengio, 2009)). This method could yield better results by explicitly imbuing information about the target affective state into the features; however, large datasets would be required to prevent overfitting.

Larger topologies and features sets

The reduced size of the datasets used in the evaluation limited the number of features that could be learned via deep learning. Currently, deep architectures are widely used to extract thousands of features from large datasets, which yields models that outperform other state-of-the-art classification or regression methods (e.g. (Krizhevsky et al., 2012; Farabet et al., 2013)). It is expected that the application of DL to model affect in large physiological datasets would show larger improvements with respect to ad-hoc features and provide new insights on the relationship between physiology and affect. Additionally, as mentioned above, a wider range of topologies and normalization schemes could enable CNNs to capture a wider range of signal components providing a larger difference with respect to ad-hoc feature extraction. Finally, a combination of several CNNs with dissimilar topologies to detect patterns of different time resolution, would surely yield to more accurate models (as seen in Rifai et al., 2012).

8.4 Summary

This thesis has presented a complete methodology for learning models of affect from monitored experiences annotated with post-experience ordinal self-reports. The models are based on preference learning methods — artificial neural networks, support vector machines and Cohen's method — that estimate the affective state of the user relying on physiological and contextual features extracted automatically through unsupervised machine learning and data mining techniques (deep learning and frequent sequence mining). The methodology presents a number of limitations inherent to the basic problems of learning models from observations of affect and a few drawbacks of the algorithms used. The promising results reported in this dissertation suggest that the methodology would create accurate models of affect in different domains and a number of extensions to the method are expected to enhance results even further.

Bibliography

E. Abbasnejad, E. Bonilla, and S. Sanner. Sparse gaussian processes for learning preferences. In *Proceedings of Workshop on Choice Models and Preference Learning (CMPL)*, 2011. — Cited on pages 37 and 61.

O. AlZoubi, R. Calvo, and R. Stevens. Classification of eeg for affect recognition: An adaptive approach. In *AI 2009: Advances in Artificial Intelligence*, pages 52–61. Springer, 2009. — Cited on pages 30, 32, and 35.

Omar Alzoubi, Md. Sazzad Hussain, Sidney D'Mello, and Rafael A. Calvo. Affective modeling from multichannel physiology: Analysis of day differences. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 4–13. Springer, 2011. ISBN 978-3-642-24599-2. — Cited on pages 30 and 35.

J.L. Andreassi. *Psychophysiology: Human Behavior and Physiological Response*. Psychology Press, 2000. — Cited on pages 32 and 49.

I. Arroyo, D.G. Cooper, W. Burleson, B.P. Woolf, K. Muldner, and R. Christopherson. Emotion sensors go to school. In *Proceedings of Conference on Artificial Intelligence in Education (AIED)*, pages 17–24. IOS Press, 2009. — Cited on page 32.

S. Asteriadis, P. Tzouveli, K. Karpouzis, and S. Kollias. Estimation of behavioral user state based on eye gaze and head poseapplication in an e-learning environment. *Multimedia Tools and Applications*, 41(3):469–493, 2009. — Cited on pages 32 and 153.

H. Aviezer, R.R. Hassin, J. Ryan, C. Grady, J. Susskind, A. Anderson, M. Moscovitch, and S. Bentin. Angry, disgusted, or afraid? studies on the malleability of emotion perception. *Psychological Science*, 19(7):724–732, 2008. — Cited on page 50.

A. Bahamonde, J. Díez, J.R. Quevedo, O. Luaces, and J.J. del Coz. How to learn consumer preferences from the analysis of sensory data by means of support vector machines (svm). *Trends in Food Science & Technology*, 18(1):20–28, 2007. — Cited on page 37.

B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger. Learning to rank with (a lot of) word features. *Information retrieval*, 13 (3):291–314, 2010. — Cited on page 38.

J.N. Bailenson, E.D. Pontikakis, I.B. Mauss, J.J. Gross, M.E. Jabon, C.A.C. Hutcherson, C. Nass, and O. John. Real-time classification of evoked emotions using facial feature tracking and physiological responses. *International journal of human-computer studies*, 66(5):303–317, 2008. — Cited on pages 30, 35, and 37.

L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra. Effective codebooks for human action representation and classification in unconstrained videos. *IEEE Transactions on Multimedia*, 14(4):1234–1245, 2012. — Cited on page 39.

R. Banse and K.R. Scherer. Acoustic profiles in vocal emotion expression. *Journal of personality and social psychology*, 70(3):614, 1996. — Cited on page 32.

L.F. Barrett, B. Mesquita, K.N. Ochsner, and J.J. Gross. The experience of emotion. *Annual review of psychology*, 58:373, 2007. — Cited on pages 50 and 150.

A. Ben-Hur and D. Brutlag. Remote homology detection: a motif based approach. *Bioinformatics*, 19(suppl 1):i26, 2003. ISSN 1367-4803. — Cited on page 39.

Y. Bengio. On the challenge of learning complex functions. *Progress in Brain Research*, 165:521–534, 2007. — Cited on page 113.

Y. Bengio. Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2(1):1–127, 2009. — Cited on pages 39, 51, 71, and 154.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems (NIPS)*, volume 19, page 153. MIT; 1998, 2007. — Cited on pages 33, 39, 54, and 147.

Y. Bengio, A. Courville, and P. Vincent. Unsupervised feature learning and deep learning: A review and new perspectives. Technical Report Arxiv report 1206.5538, Université de Montréal, 2012. — Cited on pages 46 and 54.

N. Bianchi-Berthouze and C.L. Lisetti. Modeling multimodal expression of users affective subjective experience. *User Modeling and User-Adapted Interaction*, 12(1):49–84, 2002. — Cited on page 32.

C.M. Bishop. *Neural networks for pattern recognition*. Clarendon press Oxford, 1995. — Cited on pages 36, 62, and 63.

N. Bohnen, N. Nicolson, J. Sulon, and J. Jolles. Coping style, trait anxiety and cortisol reactivity during mental stress. *Journal of psychosomatic research*, 35(2-3):141–147, 1991. — Cited on page 49.

Margaret M Bradley and Peter J Lang. Affective reactions to acoustic stimuli. *Psychophysiology*, 37(02):204–215, 2000. — Cited on page 30.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of International conference on Machine learning (ICML)*, pages 89–96. ACM, 2005. ISBN 1-59593-180-5. — Cited on pages 38 and 61.

C. Busso, Z. Deng, S. Yildirim, M. Bulut, C.M. Lee, A. Kazemzadeh, S. Lee, U. Neumann, and S. Narayanan. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of international conference on Multimodal interfaces (ICMI)*, pages 205–211. ACM, 2004. — Cited on pages 32 and 34.

J.T. Cacioppo and L.G. Tassinary. Inferring psychological significance from physiological signals. *American Psychologist*, 45(1):16–28, 1990. — Cited on page 150.

R. Calvo, I. Brown, and S. Scheding. Effect of experimental factors on the recognition of affective mental states through physiological measures. In *AI 2009: Advances in Artificial Intelligence*, pages 62–70. Springer, 2009. — Cited on pages 30 and 32.

R.A. Calvo and S. D'Mello. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1(1):18–37, 2010. — Cited on pages 19 and 29.

R.A. Calvo and S.K. D'Mello. *New perspectives on affect and learning technologies*, volume 3. Springer, 2011. — Cited on page 42.

G. Caridakis, S. Asteriadis, K. Karpouzis, and S. Kollias. Detecting human behavior emotional cues in natural interaction. In *Proceedings of International Conference on Digital Signal Processing (DSP)*, pages 1 –6, july 2011. doi: 10.1109/ICDSP.2011.6004962. — Cited on page 33.

R. Caruana, S. Baluja, and T. Mitchell. Using the future to" sort out" the present: Rankprop and multitask learning for medical risk evaluation. In *Advances in neural information processing systems (NIPS)*, pages 959–965. Morgan Kaufmann Publishers, 1996. — Cited on pages 37 and 38.

H. Cecotti and A. Graser. Convolutional neural networks for p300 detection with application to brain-computer interfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):433 –445, march 2011. — Cited on page 37.

J.C. Chan. Response-order effects in likert-type scales. *Educational and Psychological Measurement*, 51(3):531–540, 1991. — Cited on pages 31, 47, and 151.

Marcela Charfuelan and Marc Schröder. Investigating the prosody and voice quality of social signals in scenario meetings. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, volume 6974, pages 46–56. Springer, 2011. ISBN 978-3-642-24599-2. — Cited on page 34.

S.Å. Christianson and M.A. Safer. Emotional events and emotions in autobiographical memories. In *Remembering our past: Studies in autobiographical memory*, pages 218–243. Cambridge University Press, 1996. — Cited on page 150.

W. Chu and Z. Ghahramani. Preference learning with gaussian processes. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 137–144. ACM, 2005. — Cited on pages 37 and 61.

G.L. Clore. Why emotions are never unconscious. In *The nature of emotion: Fundamental questions*, pages 285–290. 1994. — Cited on pages 31 and 47.

W.W. Cohen, R.E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999. — Cited on pages 61 and 71.

C. Conati and H. Maclaren. Modeling user affect from causes and effects. *User Modeling, Adaptation, and Personalization*, pages 4–15, 2009. — Cited on pages 32, 41, and 42.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. — Cited on pages 36 and 69.

H.L. Costner. Theory, deduction, and rules of correspondence. *American Journal of Sociology*, pages 245–263, 1969. — Cited on pages 23, 32, and 48.

K. Crammer and Y. Singer. Pranking with ranking. In *Advances in neural information processing systems (NIPS)*, volume 14, pages 641–647, 2001. — Cited on pages 38 and 61.

M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1(1-4): 131–156, 1997. — Cited on page 46.

R.J. Davidson. On emotion, mood, and related affective constructs. In *The nature of emotion: Fundamental questions*, pages 51–55. Oxford University Press New York, 1994. — Cited on page 21.

R.J. Davidson, P. Ekman, N.H. Frijda, HH Goldsmith, J. Kagan, R. Lazarus, J. Panksepp, D. Watson, and L.A. Clark. How are emotions distinguished from moods, temperament, and other related affective constructs? In *The nature of emotion: Fundamental questions*. Oxford University Press, 1994. — Cited on page 21.

O. Delalleau, E. Contal, E. Thibodeau-Laufer, R. Chandias, Y. Bengio, and F. Zhang. Beyond skill rating: Advanced matchmaking in ghost recon online. *IEEE Transactions on Computational Intelligence and AI in Games*, (99):1–1, 2011. — Cited on page 61.

J. Dennerlein, T. Becker, P. Johnson, C. Reynolds, and R.W. Picard. Frustrating computer users increases exposure to physical factors. In *Proceedings of the International Ergonomics Association (IEA)*, 2003. — Cited on page 32.

L. Devillers and L. Vidrascu. Real-life emotions detection with lexical and paralinguistic cues on human-human call center dialogs. In *Proceedings of Conference of the International Speech Communication Association (Interspeech)*, pages 801–804, 2006. — Cited on page 30.

E. Diener. Subjective well-being: The science of happiness and a proposal for a national index. *American psychologist*, 55(1):34, 2000. — Cited on pages 31 and 47.

S. D'Mello and A. Graesser. Automatic detection of learner's affect from gross body language. *Applied Artificial Intelligence*, 23(2):123–150, 2009. — Cited on pages 30 and 32.

A. Drachen, L. Nacke, G. N. Yannakakis, and A. L. Pedersen. Correlation between heart rate, electrodermal activity and player experience in first-person shooter games. In *Proceedings of the SIGGRAPH Symposium on Video Games*. ACM-SIGGRAPH Publishers, 2010. — Cited on pages 34 and 40.

H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *Advances in neural information processing systems (NIPS)*, pages 155–161. MORGAN KAUFMANN PUBLISHERS, 1997. — Cited on page 36.

H. Drucker, D. Wu, and V.N. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999. — Cited on page 36.

P. Ekman and W.V. Friesen. Facial action coding system: A technique for the measurement of facial movement. 12:271–302, 1978. — Cited on pages 33 and 153.

S.H. Fairclough. Fundamentals of physiological computing. *Interacting with computers*, 21 (1):133–145, 2009. — Cited on page 20.

C. Farabet, C. Couprie, L. Najman, Y. LeCun, et al. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–15, 2013. — Cited on pages 33 and 155.

R. Fernandez and R.W. Picard. Modeling drivers speech under stress. *Speech Communication*, 40(1):145–159, 2003. — Cited on pages 30 and 35.

P. Ferreira and P. Azevedo. Protein sequence classification through relevant sequence mining and bayes classifiers. *Progress in Artificial Intelligence*, pages 236–247, 2005. — Cited on page 40.

C.N. Fiechter and S. Rogers. Learning subjective functions with large margins. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 287–294, 2000. — Cited on page 38.

R.R. Fletcher, K. Dobson, M.S. Goodwin, H. Eydgahi, O. Wilder-Smith, D. Fernholz, Y. Kuboyama, E.B. Hedman, M.Z. Poh, and R.W. Picard. icalm: Wearable sensor and network architecture for wirelessly communicating and logging autonomic activity. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):215–223, 2010. — Cited on page 20.

Julien Fleureau, Philippe Guillotel, and Quan Huynh-Thu. Physiological-based affect event detector for entertainment video applications. 2012. — Cited on page 41.

T. Force. Heart rate variability: standards of measurement, physiological interpretation and clinical use. task force of the european society of cardiology and the north american society of pacing and electrophysiology. *Circulation*, 93(5):1043–65, 1996. — Cited on page 84.

Y. Freund and D. Haussler. Unsupervised learning of distributions of binary vectors using two layer networks. Technical report, University of California, 1994. — Cited on page 39.

Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933–969, 2003. — Cited on page 61.

N.H. Frijda. *The emotions: Studies in emotion and social interaction*, volume 44. Cambridge University Press, 1986. — Cited on page 19.

N.H. Frijda. Moods, emotion episodes, and emotions. In *Handbook on Emotions*. Guilford Press, 1993. — Cited on page 21.

J. Fürnkranz and E. Hüllermeier. *Preference learning*. Springer, 2010a. — Cited on pages 23, 46, 60, and 61.

J. Fürnkranz and E. Hüllermeier. Preference learning: An introduction. In *Preference Learning*, pages 1–17. Springer, 2010b. — Cited on page 36.

D. Galin. The structure of awareness: Contemporary applications of william james' forgotten concept of" the fringe". *Journal of Mind and Behavior*, 15:375–375, 1994. — Cited on page 30.

M. Garber-Barron and Mei Si. Using body movement and posture for emotion detection in non-acted scenarios. In *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1 –8, june 2012. doi: 10.1109/FUZZ-IEEE.2012.6250780. — Cited on page 35.

MW Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)–a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15): 2627–2636, 1998. — Cited on page 37.

D. Giakoumis, D. Tzovaras, K. Moustakas, and G. Hassapis. Automatic recognition of boredom in video games using novel biosignal moment-based features. *IEEE Transactions on Affective Computing*, 2(3):119 –133, july-sept. 2011. ISSN 1949-3045. doi: 10.1109/T-AFFC.2011.4. — Cited on page 33.

D. Giakoumis, A. Drosou, P. Cipresso, D. Tzovaras, G. Hassapis, T. Zalla, et al. Using activity-related behavioural features towards more effective automatic. *PLoS ONE*, 7(9), 2012. — Cited on pages 34 and 35.

K. Gilleade, A. Dix, and J. Allanson. Affective videogames and modes of affective gaming: assist me, challenge me, emote me. In *Proceedings of Digital Games Research Association Conference (DiGRA)*, volume 2005, 2005. — Cited on pages 20 and 40.

D.E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley, 1989. — Cited on pages 39 and 63.

J.J. Goldberger, S. Challapalli, R. Tung, M.A. Parker, and A.H. Kadish. Relationship of heart rate variability to parasympathetic effect. *Circulation*, 103(15):1977, 2001. — Cited on pages 83, 84, and 130.

D. Goleman. *Emotional intelligence: Why it can mater more than IQ*. Bantam, 2006. — Cited on page 19.

J. Grafsgaard, K. Boyer, and J. Lester. Predicting facial indicators of confusion with hidden markov models. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 97–106. Springer, 2011. — Cited on pages 32, 35, 42, and 45.

NM Graham, RC Bartholomeusz, N. Taboonpong, and JT La Brooy. Does anxiety reduce the secretion rate of secretory iga in saliva? *The Medical Journal of Australia*, 148(3): 131, 1988. — Cited on page 49.

D. Grangier and S. Bengio. Inferring document similarity from hyperlinks. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM)*, pages 359–360. ACM, 2005. — Cited on page 38.

J. Gratch and S. Marsella. Tears and fears: Modeling emotions and emotional behaviors in synthetic agents. In *Proceedings of international conference on Autonomous agents (AA)*, pages 278–285. ACM, 2001. — Cited on page 50.

Oliver Grewe, Frederik Nagel, Reinhard Kopiez, and Eckart Altenmüller. Emotions over time: synchronicity and development of subjective, physiological, and facial affective reactions to music. *Emotion*, 7(4):774, 2007. — Cited on page 41.

H. Gunes and M. Piccardi. Bi-modal emotion recognition from expressive face and body gestures. *Journal of Network and Computer Applications*, 30(4):1334–1345, 2007. — Cited on page 35.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002. — Cited on page 36.

P. Hamel, S. Lemieux, Y. Bengio, and D. Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *Proceedings of International Conference on Music Information Retrieval (ICMIR)*, 2011. — Cited on pages 37 and 152.

J.B. Hampshire, A.H. Waibel, et al. A novel objective function for improved phoneme recognition using time-delay neural networks. *IEEE Transactions on Neural Networks*, 1 (2):216–228, 1990. — Cited on page 37.

J. Han and M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006. ISBN 1558609016. — Cited on page 39.

G.R. Harik, F.G. Lobo, and D.E. Goldberg. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):287–297, 1999. — Cited on page 64.

E.F. Harrington. Online ranking/collaborative filtering using the perceptron algorithm. In *Proceedings of International Conference on Machine Learning (ICML)*, volume 20, page 250, 2003. — Cited on page 38.

R.L. Hazlett. Measuring emotional valence during interactive experiences: boys at video game play. In *Proceedings of SIGCHI conference on Human Factors in computing systems (CHI)*, pages 1023–1026. ACM, 2006. — Cited on pages 34 and 41.

L. He, M. Lech, N. Maddage, and N. Allen. Stress and emotion recognition using log-gabor filter analysis of speech spectrograms. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–6, 2009. — Cited on page 34.

Jennifer A Healey and Rosalind W Picard. Detecting stress during real-world driving tasks using physiological sensors. *Intelligent Transportation Systems, IEEE Transactions on*, 6(2):156–166, 2005. — Cited on pages 30 and 31.

A. Heraz and C. Frasson. Predicting the three major dimensions of the learners emotions from brainwaves. *World Academy of Science, Eng. and Technology*, 25:323–329, 2007. — Cited on page 35.

R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, volume 1, pages 97 –102, 1999. doi: 10.1049/cp:19991091. — Cited on pages 61, 70, and 71.

R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in large margin classifiers*, volume 88, pages 115–132. MIT Press, 2000. — Cited on page 69.

Javier Hernandez, Rob R. Morris, and Rosalind W. Picard. Call center stress recognition with person-specific models. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, volume 6974, pages 125–134. Springer, 2011. ISBN 978-3-642-24599-2. — Cited on pages 31 and 35.

G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. — Cited on pages 34 and 54.

G.E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. — Cited on page 39.

Geoffrey E. Hinton and R. S. Zemel. Autoencoders, minimum description length, and helmholtz free energy. In *Advances in neural information processing systems (NIPS)*, 1994. — Cited on page 54.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the International ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 50–57. ACM, 1999. — Cited on page 38.

K. Höök. Affective loop experiences: designing for interactional embodiment. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3585–3595, 2009. — Cited on page 20.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. — Cited on page 37.

E. Hudlicka. Affective game engines: motivation and requirements. In *Proceedings of International Conference on Foundations of Digital Games (FDG)*, pages 299–306. ACM, 2009. — Cited on pages 20 and 40.

M. Hussain, O. AlZoubi, R. Calvo, and S. DMello. Affect detection from multichannel physiology during learning sessions with autotutor. In *Proceedings of International Conference in Artificial Intelligence in Education (AIED)*, pages 131–138. Springer, 2011. — Cited on page 42.

W. James. *The principles of psychology*. Henry Holt and Co, 1890. — Cited on page 30.

J.H. Janssen, E.L. van den Broek, and J.H.D.M. Westerink. Personalized affective music player. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–6. IEEE, 2009. — Cited on pages 30 and 41.

D. Jiang, Y. Cui, X. Zhang, P. Fan, I. Ganzalez, and H. Sahli. Audio visual emotion recognition based on triple-stream dynamic bayesian network models. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction*, pages 609–618. Springer, 2011. — Cited on page 45.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of European Conference on Machine learning (ECML)*, pages 137–142. Springer, 1998. — Cited on page 36.

T. Joachims. Making large-scale svm learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods-Support Vector Learning*. MIT-press, 1999. — Cited on page 114.

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD international conference on Knowledge Discovery in Data Mining (KDD)*, pages 133–142. ACM, 2002. — Cited on pages 37, 61, and 71.

T. Joachims. Training linear svms in linear time. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD)*, pages 217–226, 2006. — Cited on page 36.

T. Johnstone and K.R. Scherer. Vocal communication of emotion. In *Handbook of emotions*, volume 2, pages 220–235. Guilford Press New York, 2000. — Cited on page 32.

P.N. Juslin and K.R. Scherer. *Vocal expression of affect*. Oxford University Press, Oxford, UK, 2005. — Cited on page 32.

J. Kagan, J.S. Reznick, and N. Snidman. The physiology and psychology of behavioral inhibition in children. *Child development*, pages 1459–1473, 1987. — Cited on page 150.

D. Kahneman. Objective happiness. In *Well-being: The foundations of hedonic psychology*. Russell Sage Foundation, 1999. — Cited on page 151.

R. Kaliouby and P. Robinson. Real-time inference of complex mental states from facial expressions and head gestures. In *Real-time vision for human-computer interaction*, pages 181–200. Springer, 2005. — Cited on pages 35 and 45.

R. Kaliouby, R. Picard, and S. Baron-Cohen. Affective computing and autism. *Annals of the New York Academy of Sciences*, 1093(1):228–248, 2006. — Cited on pages 20 and 42.

T. Kamishima, H. Kazawa, and S. Akaho. A survey and empirical comparison of object ranking methods. In *Preference Learning*. Springer. — Cited on page 72.

T. Kamishima, H. Kazawa, and S. Akaho. Supervised ordering-an empirical survey. In *Proceedings of International Conference on Data Mining (ICDM)*, pages 4–pp. IEEE, 2005. — Cited on pages 37 and 61.

T. Kannetis and A. Potamianos. Towards adapting fantasy, curiosity and challenge in multimodal dialogue systems for preschoolers. In *Proceedings of international conference on Multimodal interfaces (ICMI)*, pages 39–46. ACM, 2009. — Cited on page 32.

A. Kapoor, W. Burleson, and R.W. Picard. Automatic prediction of frustration. *International Journal of Human-Computer Studies*, 65(8):724–736, 2007. — Cited on pages 30, 31, 32, 33, and 35.

Kostas Karpouzis, George Caridakis, Roddy Cowie, and Ellen Douglas-Cowie. Induction, recording and recognition of natural emotions from facial expressions and speech prosody. *Journal on Multimodal User Interfaces*, pages 1–12, 2013. — Cited on page 30.

Stéphanie Khalfa, Peretz Isabelle, Blondin Jean-Pierre, and Robert Manon. Event-related skin conductance responses to musical emotions in humans. *Neuroscience letters*, 328(2): 145–149, 2002. — Cited on page 41.

K.H. Kim, S.W. Bang, and S.R. Kim. Emotion recognition system using short-term monitoring of physiological signals. *Medical and biological engineering and computing*, 42(3): 419–427, 2004. — Cited on pages 34 and 35.

R. Kiros and C. Szepesvari. Deep representations and codes for image auto-annotation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 917–925, 2012. — Cited on page 39.

J.M. Kivikangas, I. Ekman, G. Chanel, S. Järvelä, M. Salminen, B. Cowley, P. Henttonen, and N. Ravaja. Review on psychophysiological methods in game research. In *Procedings of Nordic Digital Games Research Association Conference (Nordic DiGRA)*, 2010. — Cited on page 29.

A. Kleinsmith and N. Bianchi-Berthouze. Affective body expression perception and recognition: A survey. *IEEE Transactions on Affective Computing*, PP(99):1, 2012. ISSN 1949-3045. doi: 10.1109/T-AFFC.2012.16. — Cited on pages 29, 30, and 33.

H. Kobayashi and F. Hara. Dynamic recognition of basic facial expressions by discrete-time recurrent neural network. In *Proceedings of International Joint Conference on Neural Networks*, volume 1, pages 155 – 158 vol.1, oct. 1993. doi: 10.1109/IJCNN.1993.713882. — Cited on pages 35 and 45.

S. Koelstra, C. Muhl, M. Soleymani, J.S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. Deap: A database for emotion analysis; using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, 2012. — Cited on pages 30, 31, 41, 73, and 85.

Ron Kohavi, Roger Longbotham, Dan Sommerfield, and RandalM. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18:140–181, 2009. ISSN 1384-5810. doi: 10.1007/s10618-008-0114-1. URL `http://dx.doi.org/10.1007/s10618-008-0114-1`. — Cited on page 31.

A.N. Kolmogorov. *On the Representation of Continuous Functions of Several Variables in the Form of Super Positions of Continuous Functions of One Variable and Additive Functions*. SLA Translations Center, 1963. — Cited on page 62.

S. Kramer, G. Widmer, B. Pfahringer, and M. Groeve. Prediction of ordinal classes using regression trees. *Fundamenta Informaticae*, 47(1-2):1–13, 2001. — Cited on page 61.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. — Cited on page 155.

WJ Krzanowski. The performance of fisher's linear discriminant function under non-optimal conditions. *Technometrics*, 19(2):191–200, 1977. — Cited on pages 34 and 59.

P.J. Lang, M.M. Bradley, and B.N. Cuthbert. International affective picture system (iaps): Technical manual and affective ratings, 1999. — Cited on page 30.

Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks*, volume 3361. Cambridge, MA: MIT Press, 1995. — Cited on pages 33 and 51.

Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. — Cited on page 37.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. — Cited on page 37.

C.M. Lee and S.S. Narayanan. Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, 13(2):293–303, 2005. — Cited on pages 34 and 35.

N. Lesh, M.J. Zaki, and M. Ogihara. Mining features for sequence classificatio. In *Proceedings of ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD)*, pages 342–346. ACM, 1999. ISBN 1581131437. — Cited on page 40.

R.W. Levenson. Emotion elicitation with neurological patients. In *The handbook of emotion elicitation and assessment*, pages 158–168. Oxford University Press New York, NY, 2007. — Cited on pages 30, 41, and 47.

L. Lidberg, S.E. Levander, and D. Schalling. Habituation of the digital vasoconstrictive orienting response. *Journal of Experimental Psychology; Journal of Experimental Psychology*, 102(4):700, 1974. — Cited on page 126.

R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932. — Cited on pages 31 and 38.

C. Lisetti, F. Nasoz, C. LeRouge, O. Ozyer, and K. Alvarez. Developing multimodal intelligent affective interfaces for tele-home health care. *International Journal of Human-Computer Studies*, 59(1):245–255, 2003. — Cited on page 42.

C. Liu, K. Conn, N. Sarkar, and W. Stone. Physiology-based affect recognition for computer-assisted intervention of children with autism spectrum disorder. *International journal of human-computer studies*, 66(9):662–677, 2008. — Cited on page 42.

P.N. Lopes, P. Salovey, S. Côté, M. Beers, and R.E. Petty. Emotion regulation abilities and the quality of social interaction. *Emotion*, 5(1):113, 2005. — Cited on page 19.

L.O. Lundqvist, F. Carlsson, P. Hilmersson, and P.N. Juslin. Emotional responses to music: experience, expression, and physiology. *Psychology of Music*, 37(1):61–90, 2009. — Cited on page 41.

R.L. Mandryk and M.S. Atkins. A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies. *International Journal of Human-Computer Studies*, 65(4):329–347, 2007. ISSN 1071-5819. — Cited on pages 35, 40, and 80.

R.L. Mandryk, K.M. Inkpen, and T.W. Calvert. Using psychophysiological techniques to measure user experience with entertainment technologies. *Behaviour & Information Technology*, 25(2):141–158, 2006. — Cited on pages 34, 40, and 50.

H. P. Martínez and G. N. Yannakakis. Genetic search feature selection for affective modeling: a case study on reported preferences. In *Proceedings of international workshop on Affective interaction in natural environments (AFFINE)*, pages 15–20. ACM, 2010. — Cited on pages 41 and 77.

H. P. Martínez and G. N. Yannakakis. Analysing the relevance of experience partitions to the prediction of players self-reports of affect. In *Proceedings of workshop on Emotion and Games (EMOGames)*, pages 538–546. Springer, 2011a. — Cited on pages 77 and 151.

H. P. Martínez and G. N. Yannakakis. Mining multimodal sequential patterns: a case study on affect detection. In *Proceedings of International Conference on Multimodal Interfaces (ICMI)*, pages 3–10. ACM, 2011b. — Cited on pages 32 and 77.

H. P. Martínez, A. Jhala, and G. N. Yannakakis. Analyzing the impact of camera viewpoint on player psychophysiology. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 394–399, Amsterdam, The Netherlands, September 2009. IEEE. — Cited on page 77.

H. P. Martínez, M. Garbarino, and G. N. Yannakakis. Generic physiological features as predictors of player experience. In *Proceedings of Affective Computing and Intelligent Interaction (ACII)*, pages 267–276. Springer, 2011. — Cited on pages 77, 127, and 150.

H. P. Martínez, Y Bengio, and G. N. Yannakakis. Learning deep physiological models of affect. *Computational Intelligence Magazine, IEEE*, 9(1):20–33, 2013. — Cited on page 30.

H.P. Martínez, K. Hullett, and G.N. Yannakakis. Extending neuro-evolutionary preference learning through player modeling. In *Proceedings of the International Conference on Computational Intelligence and Games (CIG)*, pages 313–320. IEEE, 2010. — Cited on pages 73, 77, and 144.

J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks and Machine Learning*, pages 52–59. Springer, 2011. — Cited on page 54.

M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5):555–559, 2003. — Cited on page 33.

W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biology*, 5(4):115–133, 1943. — Cited on page 36.

D. McDuff, R. el Kaliouby, and R. Picard. Crowdsourced data collection of facial responses. In *Proceedings of international conference on multimodal interfaces (ICMI)*, pages 11–18. ACM, 2011. — Cited on page 41.

D. McDuff, R. El Kaliouby, and R. Picard. Crowdsourcing facial responses to online videos. *IEEE Transactions on Affective Computing*, 3(4):456–468, 2012. — Cited on pages 20 and 31.

S. McQuiggan, S. Lee, and J. Lester. Early prediction of student frustration. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction*, pages 698–709. Springer, 2007. — Cited on pages 20, 32, 35, 40, and 42.

S.W. Mcquiggan, B.W. Mott, and J.C. Lester. Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Modeling and User-Adapted Interaction*, 18(1): 81–123, 2008. — Cited on pages 35 and 41.

A. Mehrabian. Framework for a comprehensive description and measurement of emotional states. *Genetic, social, and general psychology monographs*, 1995. — Cited on page 31.

M. Minsky and P. Seymour. *Perceptrons: An Introduction to Computational Geometry*. MIT press, 1969. — Cited on page 36.

P.W. Mirowski, Y. LeCun, D. Madhavan, and R. Kuzniecky. Comparing svm and convolutional networks for epileptic seizure prediction from intracranial eeg. In *IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, pages 244–249. IEEE, 2008. — Cited on page 37.

T.M. Mitchell et al. *Machine learning*. McGraw-Hill New York:, 1997. — Cited on page 36.

Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 2012. — Cited on page 31.

G.B. Moody. Spectral analysis of heart rate without resampling. In *Proceedings of Computers in Cardiology*, pages 715–718. IEEE, 1993. — Cited on page 84.

D.E. Moriarty and R. Miikkulainen. Forming neural networks through efficient and adaptive coevolution. *Evolutionary Computation*, 5(4):373–399, 1997. — Cited on pages 39 and 63.

J.D. Morris. Observations: Sam: The self-assessment manikinan efficient cross-cultural measurement of emotional response. *Journal of advertising research*, 35(6):63–68, 1995. — Cited on pages 31 and 86.

L. Nacke and C.A. Lindley. Flow and immersion in first-person shooters: measuring the player's gameplay experience. In *Proceedings of Conference on Future Play: Research, Play, Share*, pages 81–88. ACM, 2008. — Cited on pages 34 and 40.

V. Nair and G. Hinton. 3d object recognition with deep belief nets. In *Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 1339–1347, 2009. — Cited on page 39.

F. Nasoz, K. Alvarez, C.L. Lisetti, and N. Finkelstein. Emotion recognition from physiological signals using wireless sensors for presence technologies. *Cognition, Technology & Work*, 6(1):4–14, 2004. — Cited on pages 35 and 41.

D. Natapov, S.J. Castellucci, and I.S. MacKenzie. Iso 9241-9 evaluation of video game controllers. In *Proceedings of Graphics Interface*, pages 223–230. Canadian Information Processing Society, 2009. — Cited on page 20.

C. Nebauer. Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks*, 9(4):685–696, 1998. — Cited on page 37.

Mihalis A Nicolaou, Hatice Gunes, and Maja Pantic. Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space. *Affective Computing, IEEE Transactions on*, 2(2):92–105, 2011. — Cited on page 35.

J. Nielsen, T. Clemmensen, and C. Yssing. Getting access to what goes on in people's heads?: reflections on the think-aloud technique. In *Proceedings of Nordic conference on Human-computer interaction (Nordic CHI)*, pages 101–110. ACM, 2002. — Cited on pages 31 and 47.

J.B. Nielsen, B.S. Jensen, and J. Larsen. On sparse multi-task gaussian process priors for music preference learning. In *Proceedings of Workshop on Choice Models and Preference Learning (CMPL)*, 2011. — Cited on page 37.

E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Proceedings of IEEE Workshop Neural Networks for Signal Processing (NNSP)*, pages 276–285. IEEE, 1997. — Cited on page 36.

Tapio Pahikkala, Evgeni Tsivtsivadze, Antti Airola, Jouni Järvinen, and Jorma Boberg. An efficient algorithm for learning to rank from preference graphs. *Machine Learning*, 75(1): 129–165, 2009. — Cited on pages 38 and 151.

PC Pandey and SV Barai. Multilayer perceptron in damage detection of bridge structures. *Computers & Structures*, 54(4):597–608, 1995. — Cited on page 37.

M. Pantic and L.J.M. Rothkrantz. Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE*, 91(9):1370–1390, 2003. ISSN 0018-9219. — Cited on pages 24 and 29.

M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *IEEE International Conference on Multimedia and Expo*, page 5. IEEE, 2005. — Cited on page 56.

C. Pedersen, J. Togelius, and G.N. Yannakakis. Modeling player experience in super mario bros. In *Proceedings of Symposium on Computational Intelligence and Games (CIG)*, pages 132 –139, 2009. — Cited on page 37.

C. Pedersen, J. Togelius, and G.N. Yannakakis. Modeling player experience for content creation. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(1):54–67, 2010. — Cited on pages 32, 34, 41, and 59.

C.H. Pedersen, R. Khaled, and G.N. Yannakakis. Ethical considerations in designing adaptive persuasive games. In *Proceedings of Persuasive Technology*, page 13, 2012. — Cited on page 42.

B. Perron. A cognitive psychological approach to gameplay emotions. In *Proceedings of Digital Games Research Association Conference (DiGRA)*, 2005. — Cited on page 20.

R.W. Picard. Affective computing. Technical report, MIT Media Laboratory Perceptual Computing, 1995. — Cited on page 19.

R.W. Picard. *Affective computing*. The MIT press, 1997. ISBN 0262661152. — Cited on page 21.

R.W. Picard. Future affective technology for autism and emotion communication. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3575–3584, 2009. — Cited on page 42.

R.W. Picard, E. Vyzas, and J. Healey. Toward machine emotional intelligence: Analysis of affective physiological state. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1175–1191, 2001. — Cited on pages 30, 33, 34, and 84.

RW Picard, Seymour Papert, Walter Bender, Bruce Blumberg, Cynthia Breazeal, David Cavallo, Tod Machover, Mitchel Resnick, Deb Roy, and Carol Strohecker. Affective learninga manifesto. *BT Technology Journal*, 22(4):253–269, 2004. — Cited on page 41.

S.M. Pincus. Approximate entropy as a measure of system complexity. In *Proceedings of the National Academy of Sciences*, volume 88, pages 2297–2301. National Acad Sciences, 1991. — Cited on page 33.

L. Qu, N. Wang, and W. Johnson. Using learner focus of attention to detect learner motivation factors. In *Proceedings of International Conference on User Modeling (UM)*, pages 149–149. Springer, 2005. — Cited on page 42.

F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *Proceedings of ACM SIGKDD international conference on Knowledge Discovery in Data Mining (KDD)*, pages 239–248. ACM, 2005. — Cited on page 37.

P. Rani, N. Sarkar, and C. Liu. Maintaining optimal challenge in computer games through real-time physiological feedback. In *Proceedings of the 11th International Conference on Human Computer Interaction*, pages 184–192, 2005. — Cited on pages 34 and 40.

N. Ravaja, T. Saari, J. Laarni, K. Kallinen, M. Salminen, J. Holopainen, and A. Jarvinen. The psychophysiology of video gaming: Phasic emotional responses to game events. In *Proceedings of Digital Games Research Association Conference (DiGRA)*, 2005. — Cited on pages 32, 34, 41, and 139.

N. Ravaja, T. Saari, M. Salminen, J. Laarni, and K. Kallinen. Phasic emotional reactions to video game events: A psychophysiological investigation. *Media Psychology*, 8(4):343–367, 2006. — Cited on pages 119 and 139.

G. Rebolledo-Mendez, I. Dunwell, E. Martínez-Mirón, M. Vargas-Cerdán, S. De Freitas, F. Liarokapis, and A. García-Gaona. Assessing neuroskys usability to detect attention levels in an assessment exercise. *Human-Computer Interaction. New Trends*, pages 149–158, 2009. — Cited on page 32.

S. Rifai, X. Muller, X. Glorot, G. Mesnil, Y. Bengio, and P. Vincent. Learning invariant features through local space contraction. Technical report, Universit de Montral - DIRO - LISA, 2011. — Cited on page 39.

S. Rifai, Y. Bengio, A. Courville, P. Vincent, and M. Mirza. Disentangling factors of variation for facial expression recognition. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012. — Cited on pages 33, 34, 153, and 155.

S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. *NIST Special Publication SP*, pages 109–109, 1995. — Cited on page 38.

M.D. Robinson and G.L. Clore. Belief and feeling: Evidence for an accessibility model of emotional self-report. *Psychological Bulletin*, 128(6):934, 2002. — Cited on pages 30, 31, 47, and 150.

J. Robison, S. McQuiggan, and J. Lester. Evaluating the consequences of affective feedback in intelligent tutoring systems. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–6. IEEE, 2009. — Cited on pages 41 and 42.

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958. — Cited on page 36.

M. Ross. Relation of implicit theories to the construction of personal histories. *Psychological review*, 96(2):341, 1989. — Cited on page 150.

D.E. Rumelhart. *Backpropagation: theory, architectures, and applications*. Lawrence Erlbaum, 1995. — Cited on page 63.

D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. — Cited on page 36.

J.A. Russell. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161, 1980. — Cited on pages 31, 46, and 80.

P.M. Sanderson and C. Fisher. Exploratory sequential data analysis: Foundations. *Human–Computer Interaction*, 9(3-4):251–317, 1994. — Cited on pages 30 and 47.

Kristina Schaaff and Tanja Schultz. Towards emotion recognition from electroencephalographic signals. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–6. IEEE, 2009. — Cited on page 30.

K.R. Scherer. Psychological models of emotion. In *The neuropsychology of emotion*. Oxford University Press, 2000. — Cited on page 21.

K.R. Scherer. Vocal communication of emotion: A review of research paradigms. *Speech communication*, 40(1):227–256, 2003. — Cited on page 19.

K.R. Scherer. What are emotions? and how can they be measured? *Social science information*, 44(4):695–729, 2005. — Cited on page 31.

B. Schölkopf and A.J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2001. — Cited on page 36.

B. Schuller, R.J. Villar, G. Rigoll, and M. Lang. Meta-classifiers in acoustic and linguistic feature fusion-based affect recognition. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 325–328, 2005. — Cited on page 34.

M. Schwartz, H. P. Martínez, G. N. Yannakakis, and A. Jhala. Iinvestigating the interplay between camera viewpoints, game information, and challenge. In *Proceedings of Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*. AAAI, October 2009. — Cited on page 77.

Norbert Schwarz. Emotion, cognition, and decision making. *Cognition & Emotion*, 14(4): 433–440, 2000. — Cited on page 41.

N. Shaker, G.N. Yannakakis, and J. Togelius. Towards automatic personalized content generation for platform games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE). AAAI Press*, 2010. — Cited on page 41.

P.K. Shivaswamy and T. Joachims. Online learning with preference feedback. *Arxiv preprint arXiv:1111.0712*, 2011. — Cited on page 61.

Fernando Silveira, Brian Eriksson, Anmol Sheth, and Adam Sheppard. Predicting audience responses to movie content from electro-dermal activity signals. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 707–716. ACM, 2013. — Cited on page 41.

M. Soleymani, G. Chanel, J.J.M. Kierkels, and T. Pun. Affective characterization of movie scenes based on multimedia content analysis and user's physiological emotional responses. In *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, pages 228–235, 2008. doi: 10.1109/ISM.2008.14. — Cited on page 41.

M. Soleymani, M. Pantic, and T. Pun. Multimodal emotion recognition in response to videos. *IEEE Transactions on Affective Computing*, 3(2):211–223, 2012. — Cited on page 35.

R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of international conference on Extending database technology: Advances in database technology (EDBT)*, pages 1–17. Springer, 1996. — Cited on pages 46 and 57.

K.O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002. — Cited on pages 39 and 64.

Stanley Smith Stevens et al. On the theory of scales of measurement, 1946. — Cited on page 48.

M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata. Pedestrian detection with convolutional neural networks. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 224–229. IEEE, 2005. — Cited on page 37.

G. Tesauro. Connectionist learning of expert preferences by comparison training. In *Advances in neural information processing systems (NIPS)*, pages 99–106. Morgan Kaufmann Publishers Inc., 1989. — Cited on page 37.

T. Tijs, D. Brokken, and W. Ijsselsteijn. Dynamic game balancing by recognizing affect. In *Proceedings of International Conference on Fun and Games*, pages 88–93. Springer, 2008. — Cited on page 40.

S. Tognetti, M. Garbarino, A.T. Bonanno, M. Matteucci, and A. Bonarini. Enjoyment recognition from physiological data in a car racing game. In *Proceedings of the international workshop on Affective interaction in natural environments (AFFINE)*, pages 3–8. ACM, 2010a. — Cited on pages 32, 34, 35, and 61.

S. Tognetti, M. Garbarino, A. Bonarini, and M. Matteucci. Modeling enjoyment preference from physiological responses in a car racing game. In *Proceedings of IEEE Conference on Computational Intelligence and Games (CIG)*, pages 321–328. IEEE, 2010b. — Cited on pages 34, 40, and 61.

Jason Tsai, Emma Bowring, Stacy Marsella, Wendy Wood, and Milind Tambe. A study of emotional contagion with virtual characters. In *Intelligent Virtual Agents*, pages 81–88. Springer, 2012. — Cited on page 31.

WM van den Hoogen, WA IJsselsteijn, and YAW de Kort. Exploring behavioral expressions of player experience in digital games. In *Proceedings of the Workshop on Facial and Bodily Expression for Control and Adaptation of Games (ECAG)*, pages 11–19, 2008. — Cited on page 32.

D. Ververidis and C. Kotropoulos. Automatic speech classification to five emotional states based on gender information. In *Procedings of European Signal Processing Conference (EUSPICO)*, pages 341–344, 2004. — Cited on pages 33 and 34.

P. Vincent, H. Larochelle, Y. Bengio, and P.A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the international conference on Machine learning (ICML)*, pages 1096–1103. ACM, 2008. — Cited on pages 39, 51, and 54.

P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010. — Cited on page 39.

M. Viswanathan. Measurement of individual differences in preference for numerical information. *Journal of Applied Psychology*, 78(5):741, 1993. — Cited on pages 23, 32, and 48.

T. Vogt and E. André. Comparing feature sets for acted and spontaneous speech in view of automatic emotion recognition. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pages 474–477. IEEE, 2005. — Cited on page 32.

E. Vyzas and R.W. Picard. Affective pattern classification. In *AAAI Fall Symposia technical report: Emotional and Intelligent: The Tangled Knot of Cognition*, volume 176182, 1998. — Cited on page 34.

J. Wagner, J. Kim, and E. André. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, pages 940–943. IEEE, 2005. — Cited on pages 34 and 35.

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(3):328–339, 1989. — Cited on page 37.

D. Watson. *Mood and temperament*. Guilford Press, 2000. — Cited on pages 31 and 47.

D. Watson and L.A. Clark. The panas-x: Manual for the positive and negative affect schedule-expanded form, 1999. — Cited on page 31.

P. Werbos. *Beyond regression: new fools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, 1974. — Cited on page 36.

B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, 1990. — Cited on page 154.

C.K.I. Williams and C.E. Rasmussen. Gaussian processes for regression. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 1996. — Cited on page 38.

S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1):37–52, 1987. — Cited on pages 34, 51, and 59.

Naiyu Wu, Huiping Jiang, and Guosheng Yang. Emotion recognition based on physiological signals. In *Advances in Brain Inspired Cognitive Systems*, volume 7366, pages 311–320. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-31560-2. — Cited on page 29.

G. Yannakakis and J. Hallam. Rating vs. preference: a comparative study of self-reporting. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 437–446. Springer, 2011. — Cited on pages 32, 35, 38, 48, 81, and 151.

G. N. Yannakakis and J. Hallam. Entertainment modeling through physiology in physical play. *International Journal of Human-Computer Studies*, 66(10):741–755, 2008. — Cited on pages 35, 81, 84, 125, and 127.

G. N. Yannakakis and A. Paiva. Emotion in games. In *Handbook on Affective Computing*. 2013. (to appear). — Cited on page 20.

G. N. Yannakakis, J. Hallam, and H. H. Lund. Entertainment capture through heart rate activity in physical interactive playgrounds. *User Modeling and User-Adapted Interaction*, 18(1):207–243, 2008. — Cited on pages 33, 35, and 81.

G.N. Yannakakis. Game adaptivity impact on affective physical interaction. In *Proceedings of International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–6. IEEE, 2009. — Cited on page 41.

G.N. Yannakakis and J. Hallam. Game and player feature selection for entertainment capture. In *Proceedings of IEEE Symposium on Computational Intelligence and Games (CIG)*, 2007. — Cited on pages 34 and 59.

G.N. Yannakakis and J. Togelius. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161, 2011. — Cited on pages 20 and 40.

G.N. Yannakakis, M. Maragoudakis, and J. Hallam. Preference learning for cognitive modeling: a case study on entertainment preferences. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 39(6):1165–1175, 2009. — Cited on pages 38 and 61.

G.N. Yannakakis, H.P. Martínez, and A. Jhala. Towards affective camera control in games. *User Modeling and User-Adapted Interaction*, 20(4):313–340, 2010. — Cited on pages 32, 38, 59, 73, 77, 83, 119, and 125.

J.H. Zar. Spearman rank correlation. In *Encyclopedia of Biostatistics*. Wiley Online Library, 1998. — Cited on page 68.

J. Zeman and J. Garber. Display rules for anger, sadness, and pain: It depends on who is watching. *Child development*, 67(3):957–973, 1996. — Cited on page 50.

Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009. — Cited on pages 29 and 32.