Authentic Emotion Detection in Real-Time Video

Yafei Sun[1], Nicu Sebe[2], Michael S. Lew[3], Theo Gevers[2]

[1]School of Computer Science and Engineering, Sichuan University, China
[2]Faculty of Science, University of Amsterdam, The Netherlands
[3]LIACS Media Lab, Leiden University, The Netherlands

Abstract. There is a growing trend toward emotional intelligence in human-computer interaction paradigms. In order to react appropriately to a human, the computer would need to have some perception of the emotional state of the human. We assert that the most informative channel for machine perception of emotions is through facial expressions in video. One current difficulty in evaluating automatic emotion detection is that there are currently no international databases which are based on authentic emotions. The current facial expression databases contain facial expressions which are not naturally linked to the emotional state of the test subject. Our contributions in this work are twofold: First, we create the first authentic facial expression database where the test subjects are showing the natural facial expressions based upon their emotional state. Second, we evaluate the several promising machine learning algorithms for emotion detection which include techniques such as Bayesian Networks, SVMs, and Decision trees.

1 Introduction

In recent years there has been a growing interest in improving all aspects of the interaction between humans and computers. It is argued that to truly achieve effective human-computer intelligent interaction (HCII), there is a need for the computer to be able to interact naturally with the user, similar to the way human-human interaction takes place. Humans interact with each other mainly through speech, but also through body gestures, to emphasize a certain part of the speech and display of emotions. Emotions are displayed by visual, vocal, and other physiological means. There is a growing amount of evidence showing that emotional skills are part of what is called "intelligence" [1, 2]. One of the important way humans display emotions is through facial expressions.

Evaluation of machine learning algorithms generally requires carefully designed ground truth. In facial expression analysis, several test sets exist such as the Cohn-Kanade [3] and JAFFE [4] databases. However, these test sets do not represent the authentic facial expressions for the corresponding emotional state. In these test sets, the subject is asked to mimic the facial expression which may correspond to an emotional state. The subject is not asked to show the natural facial expression corresponding to how he is feeling. Even within these test sets, the authors (i.e. Kanade et al. [3]) have commented that posed facial behavior is mediated by separate motor pathways than spontaneous facial behavior. As far as we are aware, this is the first attempt to create an authentic emotion database. We shall come back to this subject in Section 2.

While the authentic facial expression test set is important for evaluation and comparison, our fundamental goal is to perform real-time emotion classification using automatic machine learning algorithms. Our real-time system uses a model based non-rigid face tracking algorithm to extract motion features that serve as input to a classifier used for recognizing the different facial expressions and is discussed briefly in Section 3. We were also interested in testing different classifiers from the machine learning literature that can be used for facial expression analysis. We present an extensive evaluation of 24 classifiers using our authentic emotion database (Section 4). We have concluding remarks in Section 5.

2 Authentic Expression Analysis

In many applications of human computer interaction, it is important to be able to detect the emotional state of the person in a natural situation. However, as any photographer can attest, getting a real smile can be challenging. Asking someone to smile often does not create the same picture as an authentic smile. The fundamental reason of course is that the subject often does not feel happy so his smile is artificial and in many subtle ways quite different than a genuine smile.

2.1 Authentic Expression Database

Our goal for the authentic expression database was to create ground truth where the facial expressions would correspond to the current emotional state of the subject. We consulted several members of the psychology department who recommended that the test be constrained as follows to minimize bias. First, the subjects could not know that they were being tested for their emotional state. Knowing that one is in a scientific test can invalidate or bias the results by influencing the emotional state. Second, we would need to interview each subject after the test to find out their true emotional state for each expression. Third, we were warned that even having a researcher in the same room with the subject could bias the results.

We decided to create a video kiosk with a hidden camera which would display segments from recent movie trailers. This method had the main advantages that it would naturally attract people to watch it and we could potentially elicit emotions through different genres of video footage - i.e. horror films for shock, comedy for joy, etc. From over 60 people who used the video kiosk, we were able to get the agreement of 28 students within the computer science department for the database. After each subject had seen the video trailers, they were interviewed to find out their emotional state corresponding to the hidden camera video footage. We also secured agreement for the motion data from their video footage to be distributed to the scientific community which is one of the primary goals for this database.

In this kind of experiment, we can only capture the expressions corresponding to the naturally occurring emotions. This means that our range of emotions for the database was constrained to the ones genuinely felt by the subjects. For this database, the emotions found were either (1) Neutral; (2) Joy; (3) Surprise, or (4) Disgust. From having created the database, some items of note based purely on our experiences: (1) It is very difficult to get a wide range of emotions for all of the subjects. Having all of the subjects

experience genuine sadness for example is difficult. (2) The facial expressions corresponding to the internal emotions is often misleading. Some of the subjects appeared to be sad when they were actually happy. (3) Students are usually open to having the data extracted from the video used for test sets. The older faculty members were generally not agreeable to being part of the database.

2.2 Posed versus Authentic Expressions

In selecting facial stimuli, the issue of whether to use posed or spontaneous expressions has been hotly debated. Experimentalists and most emotion theorists argue that spontaneous expressions are the only ”true” expressions of facial emotion and therefore such stimuli are the only ones of merit.

When recording authentic facial expressions several aspects should be considered. Not all people express emotion equally well; many individuals have idiosyncratic methods of expressing emotion as a result of personal, familial, or culturally learned display rules. Situations in which authentic facial expression are usually recorded (e.g., laboratory) are often unusual and artificial. If the subject is aware of being photographed or filmed, facial expressions may not be spontaneous anymore. Even if the subject is unaware of being filmed, the laboratory situation may not encourage natural or usual emotion response. In interacting with scientists or other authorities, subjects will attempt to act in appropriate ways so that emotion expression may be masked or controlled. Additionally, there are only a few universal emotions and only some of these can be ethically stimulated in the laboratory.

On the other hand, posed expressions may be regarded as an alternative, provided that certain safeguards are followed. Increased knowledge about the face, based in large part on observation of spontaneous, naturally occurring facial expressions, has made possible a number of methods of measuring the face. These measurement techniques can be used to ascertain whether or not emotional facial behavior has occurred and what emotion is shown in a given instance. Such facial scoring provides a kind of stimulus criterion validity that is important in this area. Additionally, posers can be instructed, not to act or pose a specific emotion, but rather to move certain muscles so as to effect the desired emotional expression. In this way, experimental control may be exerted on the stimuli and the relationship between the elements of the facial expression and the responses of observers may be analyzed and used as a guide in item selection.

From the above discussion, it is clear that the authentic facial expression analysis should be performed whenever is possible. Posed expression may be used as an alternative only in restricted cases and they can be mostly used for benchmarking the authentic expressions.

3 Facial Expression Recognition

Extensive studies of human facial expressions performed by Ekman [5, 6] gave evidence to support universality in facial expressions. According to these studies, the “universal facial expressions” are those representing happiness, sadness, anger, fear, surprise, and disgust. To code facial expressions, Ekman and Friesen [6] developed the Facial Action Coding System (FACS) in which the movements on the face are described by a set of action units (AUs) which have some related muscular basis. Ekman’s work inspired

many researchers to analyze facial expressions by means of image and video processing. By tracking facial features and measuring the amount of facial movement, they attempt to categorize different facial expressions. Recent work on facial expression analysis and recognition [7–11] has used these "basic expressions" or a subset of them. The two recent surveys in the area [12, 13] provide an in depth review of many of the research done in recent years. All the methods developed are similar in that they first extract some features from the images or video, then these features are used as inputs into a classification system, and the outcome is one of the preselected emotion categories. They differ mainly in the features extracted and in the classifiers used to distinguish between the different emotions.

Our real time facial expression recognition system (described in Section 3.1) is composed of a face tracking algorithm which outputs a vector of motion features of certain regions of the face. The features are used as inputs to one of the classifiers described in Section 3.2.

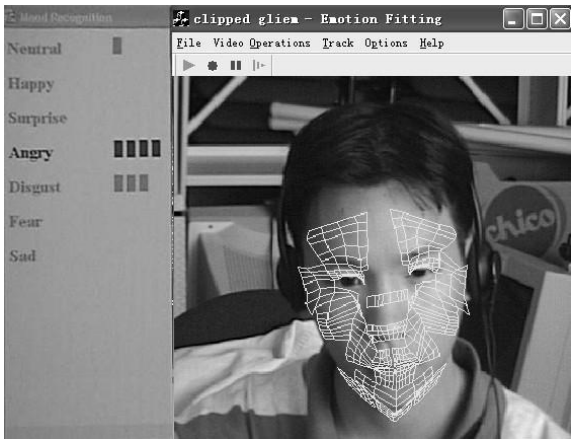
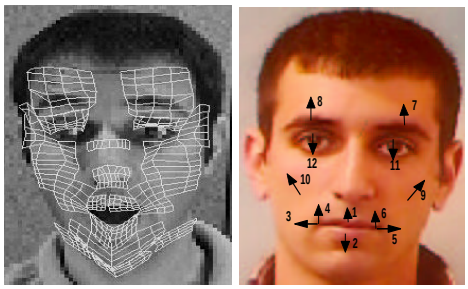


Fig. 1. A snap shot of our realtime facial expression recognition system. On the right side is a wireframe model overlayed on a face being tracked. On the left side the correct expression, Angry, is detected (the bars show the relative probability of Angry compared to the other expressions).

3.1 Our Real-Time System

A snap shot of our real-time system with the face tracking and the recognition result is shown in Figure 1. The face tracking we use is based on a system developed by Tao and Huang [14] called the Piecewise Bézier Volume Deformation (PBVD) tracker. This face tracker uses a model-based approach where an explicit 3D wireframe model of the face is constructed. In the first frame of the image sequence, landmark facial features such as the eye corners and mouth corners are selected interactively. A generic face model is then warped to fit the selected facial features. The face model consists of 16 surface patches embedded in Bézier volumes. The surface patches defined this way are guaranteed to be continuous and smooth. The shape of the mesh can be changed by changing the locations of the control points in the Bézier volume.

Once the model is constructed and fitted, head motion and local deformations of the facial features such as the eyebrows, eyelids, and mouth can be tracked. First the 2D image motions are measured using template matching between frames at different resolutions. Image templates from the previous frame and from the very first frame are both used for more robust tracking. The measured 2D image motions are modeled as projections of the true 3D motions onto the image plane. From the 2D motions of many points on the mesh, the 3D motion can be estimated by solving an overdetermined system of equations of the projective motions in the least squared sense. Figure 2(a) shows an example from one frame of the wireframe model overlayed on a face being tracked.



(a) (b)

Fig. 2. (a) The wireframe model, (b) the facial motion measurements

The recovered motions are represented in terms of magnitudes of some predefined motion of various facial features. Each feature motion corresponds to a simple deformation on the face, defined in terms of the Bézier volume control parameters. We refer to these motions vectors as Motion-Units (MU's). Note that they are similar but not equivalent to Ekman's AU's and are numeric in nature, representing not only the activation of a facial region, but also the direction and intensity of the motion. The MU's used in the face tracker are shown in Figure 2(b). The MU's are used as the basic features for the classifiers described in the next section.

3.2 Classifiers

Several classifiers from the machine learning literature were considered in our system and are listed below. We give a brief description for each of the classifiers and ask the reader to get more details from the original references. We also investigated the use of voting algorithms to improve the classification results.

Generative Bayesian Networks classifiers. Bayesian networks can represent joint distributions we usethem to compute the posterior probability of a set of *labels* given the observable *features*, and then we classify the features with the most probable label.

A Bayesian network is composed of a directed acyclic graph in which every node is associated with a variable X_i and with a conditional distribution $p(X_i|\Pi_i)$, where

Π_i denotes the parents of X_i in the graph. The directed acyclic graph is the *structure*, and the distributions $p(X_i|\Pi_i)$ represent the *parameters* of the network. A Bayesian network classifier is a *generative* classifier when the class variable is an ancestor (e.g., parent) of some or all features. We consider three examples of generative Bayesian Networks: (1) Naive-Bayes classifier [15] (**NB**) makes the assumption that all features are conditionally independent given the class label. Although this assumption is typically violated in practice, NB have been used successfully in many classification problems. Better results may be achieved by discretizing the continuous input features yielding the **NBd** classifier. (2) The Tree-Augmented Naive-Bayes classifier [16] (**TAN**) attempts to find a structure that captures the dependencies among the input features. In the structure of the TAN classifier, the class variable is the parent of all the features and each feature has at most one other feature as a parent, such that the resultant graph of the features forms a tree. (3) The Stochastic Structure Search classifier [17] (**SSS**) goes beyond the simplifying assumptions of NB and TAN and searches for the correct Bayesian network structure focusing on classification. The idea is to use a strategy that can efficiently search through the whole space of possible structures and to extract the ones that give the best classification results.

The Decision Tree Inducers. The purpose of the decision tree inducers is to create from a given data set an efficient description of a classifier by means of a decision tree. The decision tree represents a data structure which efficiently organizes descriptors. The purpose of the tree is to store an ordered series of descriptors. As one travels through the tree he is asked questions and the answers determine which further questions will be asked. At the end of the path is a classification. When viewed as a black box the decision tree represents a function of parameters (or descriptors) leading to a certain value of the classifier. We consider the following decision tree algorithms and use their $\mathcal{MLC}$++ implementation [18]: (1) **ID3** is a very basic decision tree algorithm with no pruning based on [19]. (2) **C4.5** is an extension of ID3 that accounts for unavailable values, continuous attribute value ranges, and pruning of decision trees [20]. (3) **MC4** is similar to C4.5 [20] with the exception that unknowns are regarded as a separate value. The algorithm grows the decision tree following the standard methodology of choosing the best attribute according to the evaluation criterion. After the tree is grown, a pruning phase replaces subtrees with leaves using the same pruning algorithm that C4.5 uses. (4) **OC1** is the Oblique decision tree algorithm by Murthy et al [21]. It combines deterministic hill-climbing with two forms of randomization to find a good oblique split (in the form of a hyperplane) at each node of a decision tree.

Other inducers. (1) Support vector machines [22] **SVM** were developed based on the Structural Risk Minimization principle from statistical learning theory. They are one of the most popular classifiers and can be applied to regression, classification, and density estimation problems. (2) **kNN** is the instance-based learning algorithm (nearest-neighbor) by Aha [23]. This is a good, robust algorithm, but slow when there are many attributes. (3) **PEBLS** is the Parallel Exemplar-Based Learning System by Cost and Salzberg [24]. This is a nearest-neighbor learning system designed for applications where the instances have symbolic feature values. (4) **CN2** is the direct rule induc-

tion algorithm by Clark and Niblett [25]. This algorithm inductively learns a set of propositional if...then... rules from a set of training examples. To do this, it performs a general-to-specific beam search through rule-space for the "best" rule, removes training examples covered by that rule, then repeats until no more "good" rules can be found. (5) **Winnow** is the multiplicative algorithm described in [26]. (6) **Perceptron** is the simple algorithm described in [27]. Both Perceptron and Winnow are classifiers that build linear discriminators and they are only capable of handling continuous attributes with no-unknowns and two-class problem. For our multi-class problem we implemented several classifiers, each classifying one class against the rest of the classes and in the end we averaged the results.

Voting algorithms. Methods for voting classification, such as Bagging and Boosting (AdaBoost) have been shown to be very successful in improving the accuracy of certain classifiers for artificial and real-world datasets [28]. A voting algorithm takes an inducer and a training set as input and runs the inducer multiple times by changing the distribution of training set instances. The generated classifiers are then combined to create a final classifier that is used to classify the test set.

The **bagging** algorithm (**B**ootstrap **agg**regat**ing**) by Breiman [29] votes classifiers generated by different bootstrap samples (replicates). A bootstrap sample is generated by uniformly sampling m instances from the training set with replacement. T bootstrap samples $B_1, B_2, \ldots, B_T$ are generated and a classifier C_i is built from each bootstrap sample B_i. A final classifier C^* is built from $C_1, C_2, \ldots, C_T$ whose output is the class predicted most often by its sub-classifiers, with ties broken arbitrarily. Bagging works best on unstable inducers (e.g., decision trees), that is, inducers that suffer from high variance because of small perturbations in the data. However, bagging may slightly degrade performance of stable algorithms (e.g. kNN) because effectively smaller training sets are used for training each classifier.

Like bagging **AdaBoost** (**Ada**ptive **Boost**ing) algorithm [30] generates a set of classifiers and votes them. The AdaBoost however, generates classifiers sequentially, while bagging can generate them in parallel. AdaBoost also changes the weights of the training instances provided as input to each inducer based on classifiers that were previously built. The goal is to force the inducer to minimize the expected error over different input distributions. Given an integer T specifying the number of trials, T weighted training sets $S_1, S_2, \ldots, S_T$ are generated in sequence and T classifiers $C_1, C_2, \ldots, C_T$ are built. A final classifier C^* is formed using a weighted voting scheme: the weight of each classifier depends upon its performance on the training set used to build it.

4 Facial Expression Recognition Experiments

In our experiments we use the authentic database described in Section 2. For this database we have a small number of frames for each expression which makes insufficient data to perform person dependent tests. We measure the classification error of each frame, where each frame in the video sequence was manually labeled to one of the expressions (including neutral). This manual labeling can introduce some 'noise' in our classification because the boundary between Neutral and the expression of a sequence is

Classifiers	Classification Error	Classifiers	Classification Error
NB	$8.46 \pm 0.93\%$	MC4	$8.45 \pm 0.94\%$
NB bagging	$8.35 \pm 0.92\%$	MC4 bagging	$7.35 \pm 0.76\%$
NB boosting	$8.25 \pm 0.97\%$	MC4 boosting	$5.84 \pm 0.78\%$
NBd	$8.46 \pm 0.93\%$	OC1	$9.05 \pm 1.10\%$
NBd bagging	$9.26 \pm 1.15\%$	SVM	$13.23 \pm 0.93\%$
NBd boosting	$8.65 \pm 1.03\%$	kNN	$4.43 \pm 0.97\%$
TAN	$6.46 \pm 0.34\%$	kNN bagging	$4.53 \pm 0.97\%$
SSS	$5.89 \pm 0.67\%$	kNN boosting	$4.43 \pm 0.97\%$
ID3	$9.76 \pm 1.00\%$	PEBLS	$6.05 \pm 1.09\%$
ID3 bagging	$7.45 \pm 0.66\%$	CN2	$9.26 \pm 0.82\%$
ID3 boosting	$6.96 \pm 1.00\%$	Winnow	$12.07 \pm 1.87\%$
C4.5	$8.45 \pm 0.91\%$	Perceptron	$7.75 \pm 1.41\%$

Table 1. Classification errors for facial expression recognition together with their 95% confidence intervals.

not necessarily optimal, and frames near this boundary might cause confusion between the expression and the Neutral. A different labeling scheme is to label only some of the frames that are around the peak of the expression leaving many frames in between unlabeled. We did not take this approach because a real-time classification system would not have this information available to it.

When performing the error estimation we used n-fold cross-validation (n=10 in our experiments) in which the dataset was randomly split into n mutually exclusive subsets (the folds) of approximately equal size. The inducer is trained and tested n times; each time tested on a fold and trained on the dataset minus the fold. The cross-validation estimate of error is the average of the estimated errors from the n folds. To show the statistical significance of our results we also present the 95% confidence intervals for the classification errors.

We show the results for all the classifiers in Table 1. Surprisingly, the best classification results are obtained with the kNN classifier (k=3 in our experiments). This classifier is a distance-based classifiers and does not assume any model. It seems that facial expression recognition is not a simple classification problem and all the models tried (e.g., NB, TAN, or SSS) were not able to entirely capture the complex decision boundary that separates the different expressions. This argumentation may also explain the surprisingly poor behavior of the SVM.

kNN may give the best classification results but it has its own disadvantages: it is computationally slow and needs to keep all the instances in the memory. The main advantage of the model-based classifiers is their ability to incorporate unlabel data [17]. This is very important since labeling data for emotion recognition is very expensive and requires expertise, time, and training of subjects. However, collecting unlabel data is not as difficult. Therefore, it is beneficial to be able to use classifiers that are learnt with a combination of some labeled data and a large amount of unlabeled data. Another important aspect is that the voting algorithms improve the classification results of the decision trees algorithms but do not significantly improve the results of the more stable algorithms such as NB and kNN.

We were also interested to investigate how the classification error behaves when more and more training instances are available. The corresponding learning curves are presented in Figure 3. As expected kNN improves significantly as more data are used for training.

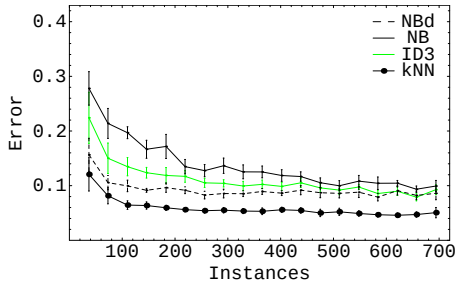


Fig. 3. The learning curve for different classifiers. The vertical bars represent the 95% confidence intervals.

5 Summary and Discussion

In this work we presented our efforts in creating an authentic facial expression database based on spontaneous emotions. We created a video kiosk with a hidden camera which displayed segments of movies and was filming several subjects that showed spontaneous emotions. One of our main contribution in this work was to create a database in which the facial expressions correspond to the true emotional state of the subjects. As far as we are aware this is the first attempt to create such a database and our intention is to make it available to the scientific community.

Furthermore, we tested and compared a wide range of classifiers from the machine learning community including Bayesian Networks, decision trees, SVM, kNN, etc. We also considered the use of voting classification schemes such as bagging and boosing to improve the classification results of the classifiers. We demonstrated the classifiers for facial expression recognition using our authentic database. Finally, we integrated the classifiers and a face tracking system to build a real time facial expression recognition system.

References

1. Salovey, P., Mayer, J.: Emotional intelligence. Imagination, Cognition, and Personality **9** (1990) 185–211
2. Goleman, D.: Emotional Intelligence. Bantam Books, New York (1995)
3. Kanade, T., Cohn, J., Tian, Y.: Comprehensive database for facial expression analysis. In: Int. Conference on Automatic Face and Gesture Recognition. (2000) 46–53
4. Lyons, M., Akamatsu, A., Kamachi, M., Gyoba, J.: Coding facial expressions with Gabor wavelets. In: IEEE International Conference on Automatic Face and Gesture Recognition. (1998) 200–205

5. Ekman, P.: Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique. Psychological Bulletin **115** (1994) 268–287
6. Ekman, P., Friesen, W.: Facial Action Coding System: Investigator's Guide. Consulting Psychologists Press (1978)
7. Bourel, F., Chibelushi, C., Low, A.: Robust facial expression recognition using a state-based model of spatially-localised facial dynamic. In: Int. Conference on Automatic Face and Gesture Recognition. (2002) 113–118
8. Zhang, Y., Ji, Q.: Facial expression understanding in image sequences using dynamic and active visual information fusion. In: ICCV. (2003) 113–118
9. Donato, G., Bartlett, M., Hager, J., Ekman, P., Sejnowski, T.: Classifying facial actions. IEEE Trans. on Pattern Analysis and Machine Intelligence **21** (1999) 974–989
10. Bartlett, M., Littlewort, G.and Fasel, I., Movellan, J.: Real time face detection and expression recognition: Development and application to human-computer interaction. In: CVPR Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction. (2003)
11. Oliver, N., Pentland, A., Bérard, F.: LAFTER: A real-time face and lips tracker with facial expression recognition. Pattern Recognition **33** (2000) 1369–1382
12. Pantic, M., Rothkrantz, L.: Automatic analysis of facial expressions: The state of the art. IEEE Trans. on PAMI **22** (2000) 1424–1445
13. Fasel, B., Luettin, J.: Automatic facial expression analysis: A survey. Pattern Recognition **36** (2003) 259–275
14. Tao, H., Huang, T.: Connected vibrations: A modal analysis approach to non-rigid motion tracking. In: CVPR. (1998) 735–740
15. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley and Sons (1973)
16. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning **29** (1997) 131–163
17. Cohen, I., Sebe, N., Cozman, F., Huang, T.: Semi-supervised learning for facial expression recognition. In: ACM Workshop on Multimedia Information Retrieval. (2003) 17–22
18. Kohavi, R., Sommerfield, D., Dougherty, J.: Data mining using $\mathcal{MLC}$++: A machine learning library in C++. International Journal on Artificial Intelligence Tools **6** (1997) 537–566
19. Quinlan, J.: Induction of decision trees. Machine Learning **1** (1986) 81–106
20. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufman (1993)
21. Murthy, S., Kasif, S., Salzberg, S.: A system for the induction of oblique decision trees. Journal of Artificial Intelligence Research **2** (1994) 1–33
22. Vapnik, V.: The Nature of Statistical Learning Theory. Springer (1995)
23. Aha, D.: Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. International Journal of Man-Machine Studies **36** (1992) 267–287
24. Cost, S., Salzberg, S.: A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning **10** (1993) 57–78
25. Clark, P., Niblett, T.: The CN2 induction algorithm. Machine Learning **3** (1989) 261–283
26. Littlestone, N.: Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning **10** (1993) 57–78
27. Hertz, J., Krogh, A., Palmer, R.: Introduction to the Theory of Neural Computation. Addison Wesley (1991)
28. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms:Bagging, boosting, and variants. Machine Learning **36** (1999) 105–142
29. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
30. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning. (1996) 148–156