16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

FACIAL EXPRESSION RECOGNITION USING ANGLE-RELATED INFORMATION FROM FACIAL MESHES

Nicholas Vretos[1], *Vassilios Solachidis*[1], *Petr Somol*[2] *and Ioannis Pitas*[1]

[1]Computer Science Department, Aristotle University of Thessaloniki
University Campus, 54124, Thessaloniki, Greece
phone: (+30) 2310 996361, fax: (+30) 2310 996304,
email: vretos,vasilis,pitas@aiia.csd.auth.gr
web: www.aiia.csd.auth.gr

[2]Dept. of Pattern Recognition, Inst. of Information Theory and Automation, CAS, Pod vodárenskou věží 4, Prague 8, 18208, Czech Republic
phone: (+420) 266052205, fax: (+420) 284683031,
email: somol@utia.cas.cz
web: ro.utia.cz

ABSTRACT

In this paper we introduce a new method for facial expression recognition. In order to be able to recognize the six main facial expressions [1] we use a grid approach and therefore we establish our new feature space based on the angles that each grid's edge form. This way we undertake several affine transformations such as translation, rotation and scaling which in other approaches are considered very harmful in the overall accuracy of a facial expression recognition algorithm. We will therefore demonstrate how we create this feature space, as well as how we apply a feature selection process within this space. The angular nature of the data impose some considerations which will be clarified in this paper.

1. INTRODUCTION

Recent advances in image and video processing are often oriented on human interpretation in video content. One of the most basic characteristic of a human is the face. Many algorithms have been built in order to detect [2], to track [3] and to recognize [4] human faces. The importance about a face is the fact that it is the main feature the mind is using to distinguish humans and also to communicate feelings. In [1] Ekman has established, under an anthropological investigation, six main facial expressions which are used in order to communicate feelings between humans. Those are until now considered as the most expressive and inter racially interpretable feelings. Most of the efforts done in facial expression recognition are targeted at finding those six or a subset of them. Those expressions, which stem from the early years of homo sapiens, where evolved in the need of communication between humans before spoken language development and are used until now, but in a different context and more or less subconsciously. Those six facial expressions, that [1] Ekman proposed, interpret the feelings of anger, surprise, happiness, disgust, fear and sadness. Often in facial expression recognition, a seventh class is considered which models the neutral face. Throughout this paper this class will be referred to as a special class. considered, we will talk about 6+1 classes.

Although it is an obvious task for humans, to interpret the six feelings from facial expressions, it is not the case for image processing algorithms. During the last decade many attempts have been undertaken to resolve this problem from different points of view. Before we address the diversity of these algorithms, it is a good practice to see where those algorithms converge in a sense. First of all, it is the region that each algorithm considers in order to find facial expressions. That is mainly the eyes, the mouth and the forehead.

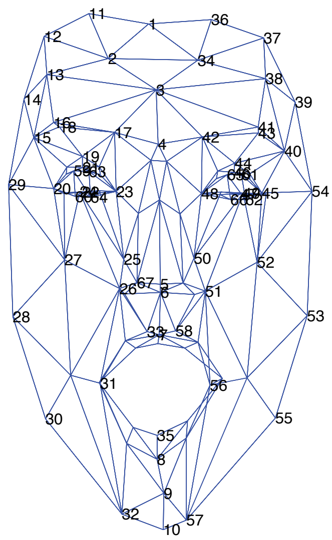

Figure 1: The Candide Face Model extracted from an image of the Kanade DataBase

For most of the algorithms, those areas are considered the ones with the more information concerning facial expressions. Another consideration for facial expression extraction which shows to be common in most of the algorithms is the fact that a facial expression is a dynamic process and not a static one. This dynamic process is considered as a three state process; an *onset* (attack), an *apex* (sustain) and an *offset* (relaxation) as described in [5]. All algorithms seem to follow that rule and even when the problem is not to recognize the whole dynamics of an expression (e.g. intensity, emotion tracking etc.), it is always explicitly marked that the *apex* state of the expression is under consideration.

The diversities of the facial expressions extraction algorithms stem from many different variations. A very useful survey can be found in [5]. In that article, the authors present an overview of the state of art and categorize facial expression extraction algorithms into two main categories. The image- (or feature-) based and the model-based with the difference being in the type of input. Both approaches have some advantages and disadvantages. In the image based algorithms for instance, one can work faster as no need of any major preprocessing is needed (debluring, noise reduction, and others are considered not a major problem). On the other hand, the model based approaches use a metadata of the face

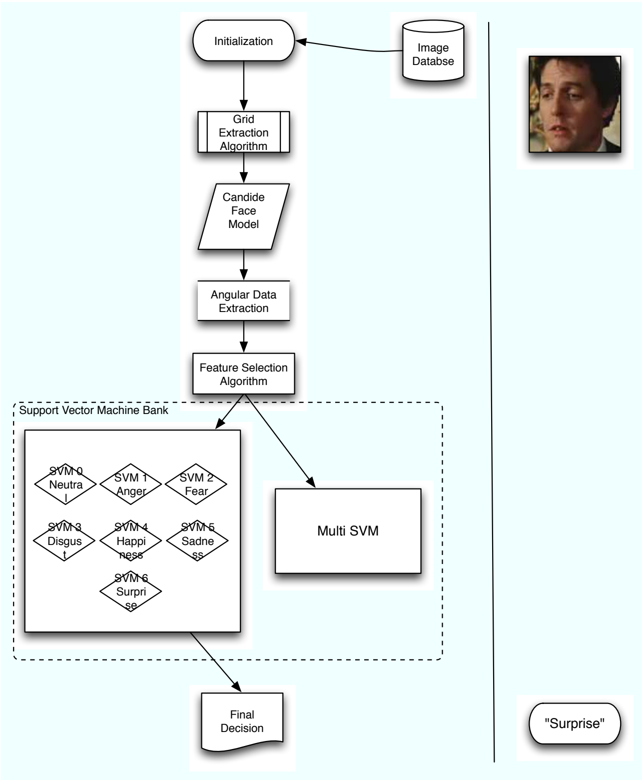

Figure 2: Framework of the facial expression extraction algorithm

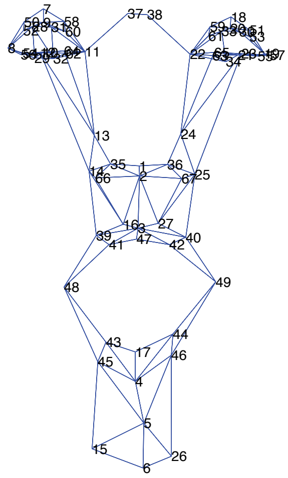

Figure 3: The 67 vertexes that are used to evaluate facial expressions

under consideration (e.g., a grid, a 2-D or 3-D model) which implies a preprocessing step for extracting these metadata. Although they are in most cases slower, model based approaches are popular between researchers because of the accuracy of the input data and also the capability of parallel research in face model extraction, what helps to improve speed and accuracy. Many differences can be found between those two approaches that mentioning all of them would not be wise. What is worth taking under consideration is that model based techniques are more reliable, but slower (because of the preprocessing) and thus not adequate for real time tasks. On the other hand, image based techniques are faster, but considerably less accurate.

In this paper, we introduce a model based technique where novelty stems from the feature space under consideration. We use a grid model and then we establish a feature space based on the angles formed between the grid's edges. Until now and to the best of our knowledge no such effort has been undertaken. Most techniques are based on vertex displacements like in [6] or vertex euclidian distances like in [7]. Both techniques are not robust against basic affine transformations like translation, rotation and scaling and need a normalization step before feature extraction step. In contrary, the proposed method while using angular data is robust against all kind of the aforementioned transformations. In Figure 2 the steps of the facial expression extraction algorithm are illustrated.

The paper is organized as follows: in Section 2 a presentation of the feature space is drawn and some consideration about the directional nature of it is discussed. In Section 3 we describe the classifiers used in our experiments, built to recognize the 6 different feellings. In Section 4 we show results on the KANADE [8] database and we establish the recognition ratio for the classifier. Finally, in Section 5 we present conclusion and discuss future work that can be done in this domain of facial expression recognition.

2. THE FEATURE SPACE

In this section we describe the feature space to be used. We first demonstrate the grid which is extracted from the face and explain the way how this grid is simplified by removing vertexes and edges. Therefore we show how the angles are calculated from the grid and finally we discuss the potential of this feature space in the domain of facial expression recognition. A discussion of the feature selection process follows.

2.1 The Face Grid

In our framework the first step is to extract a grid from the face under consideration. In order to do so, we use the Candide face model [9] and after a manual initialization of 7 vertexes in the *onset* state of the face we use the Kanade-Lucas-Tomasi (KLT) algorithm in order to track node displacements and therefore to take the *apex* state of the facial expression. The metadata produced in this process are then passed to the second processing phase where some vertexes are excluded as non-informative. The selection of vertexes is based on the FAUs as defined in [10]. We consider only the vertexes which are playing an essential role in defining the aforementioned $6+1$ facial expressions. In Figure 1 the results for a specific expression are shown. As it can be seen in Figure 3, the subset used is practically the mouth region and the eyes. Those 67 vertexes are chosen in order to further evaluate facial expressions in a face.

The grid assigns at each vertex two spacial coordinates. In the context of the grid application to the face, several problems arise. Each face has its own scale and size. It can be concluded that after the grid application, the grid will be deformed in different referential systems with totally different coordinates for each of its node. Even for the same face, we can have two very distinct face models either because the grid application has been done using different algorithms, either because the face image is scaled. Those problems are typically tackled with several normalization steps like in [6]. In this paper, we will show that this normalization process is not needed because of the feature space we use.

2.2 Angles extraction

Before we explain in detail the angle calculation process, it will be helpful to provide some background on graph theory. In a non-oriented graph the degree of a vertex d is the number of the adjacent edges towards this vertex. The adjacency matrix of a non-oriented graph is a matrix defined as follows:

$$A(i,j) = \begin{cases} 1 & e_{ij} \in E \\ 0 & e_{ij} \notin E \end{cases} \quad (1)$$

where e_{ij} is the edge connecting vertex i with vertex j. From construction we can see that this matrix is a diagonal symmetric one.

Let A be the adjacency matrix of a graph then the degree of a vertex is defined as:

$$d_i = \sum_{j=1}^{n} A(i,j) \quad (2)$$

where n is the number of vertexes in that graph. The number of angles, that can be formed in a graph are equal to the sum of all the combinations of the edges adjacent to a vertex. That is a vertex v_i which is adjacent to d_i edges can form $C_2^{d_i}$ angles. Therefore, the total number of angles formed from a graph is:

$$D = \sum_{i=0}^{N} C_2^{d_i} \quad (3)$$

where D is the total number of the angles extracted from the graph.

In order to calculate those D angles we first calculate the vectors formed by the grid's edges. That is: suppose that an edge e_{ij} is joining two vertexes v_i and v_j with spacial coordinates v_{i1}, v_{i2} and v_{j1}, v_{j2} respectively. The vector $\mathbf{x}_p$ is formed from the coordinates of the two vertexes as :

$$\mathbf{x}_p = \begin{bmatrix} |v_{i1} - v_{j1}| \\ |v_{i2} - v_{j2}| \end{bmatrix} \quad (4)$$

Therefore, the angle between two vectors $\mathbf{x}_p$ and $\mathbf{x}_q$ equals the difference between the angle they form with the axe xx' as shown in Figure 4. We do not use the inner product of the two vectors, due to the fact that in our grid there may be angles which are greater than π. Because of the cos function in the inner product formula this would result in wrong angle calculation. On the other hand, our approach will result in angles between 0 to $2 \cdot \pi$ by taking the modulo $2 \cdot \pi$ of the difference between the two aforementioned angles.

As mentioned before the sub-grid has 67 vertexes and those vertexes form a graph with 157 edges. The number of angles that can be found is 642 in total. This is, as it will be explained later in this section, the dimension of each feature vector in the feature space. From a practical point of view, many of these angles can be calculated throughout a trigonometrical process using relations that occur from the triangles they form. That is not all the 642 angles are necessary to be calculated from the process mentioned before, relations between those angles (e.g. they form the same triangle, or they are supplementary, vertical etc.) can provide information so as to avoid recalculation.

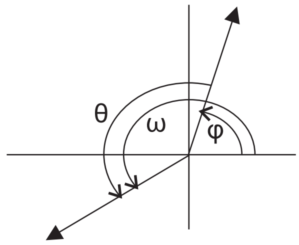

Figure 4: Angles formed between vectros

2.3 Feature Selection

Regarding the 642 features it can be expected that many of them strongly depend on each other and that the amount of information carried by various features and its impact on classification accuracy may vary considerably. Therefore we perform feature selection (FS) to optimize the performance of our classification system.

To reduce the number of features we have resorted to the Wrapper approach [11]. In this approach the feature selection process directly optimizes the performance of a chosen classifier.

In our case we deal with a dataset with difficult dimensionality vs. sample size ratio, therefore we follow the recent common apprach of testing more than one feature selection algorithm of various optimization strenghts. This enables us to identify the most suitable approach with respect to generalization performance for our case.

The methods considered for our purpose are: 1) Individually Best ranking (IB), 2) Sequential Forward Selection (SFS) [12], and 3) Sequential Forward Floating Selection (SFFS) [13]. The methods represent a hierarchy, with IB being the weakest regarding optimization performance (and ignoring completely inter-feature relations) but being also the least prone to over-fitting; while SFFS represents the most powerful but potentially the most prone to over-fitting and consequently to degraded resulting performance on independent data. Other methods exist, especially on the high-optimization-power side, but these are not recommended for use with our type of data (high dimensionality, low number of samples).

In all cases we choose the resulting number of features so as maximize the classification accuracy. In case of IB we evaluate each feature individually, then rank the features descending according to their individual values and then we evaluate each subset of first d features for each $d \in \langle 1, 642 \rangle$. In case of both SFS and SFFS we let the algorithm run until the full dimensionality is achieved. Both algorithms store all intermediate results throughout the course of search, therefore we can eventually select simply the subset with highest respective criterion value.

3. CLASSIFICATION

short discussion about the classification schemes that where used in order to make a clustering of the six main facial expressions. We mainly vector machine framework (SVMs). We tested our method with three different classifiers: Support Vector Machine (SVM) in two settings and for comparison purposes also the 3-Nearest Neighbor classifier. Both for use with SVM and 3NN we scaled all data values to $\langle 0, 1 \rangle$.

3.1 Support Vector Machines

We used Support Vector Machine (SVM) in two settings - one with Linear kernel and another with Radial Basis Function (RBF). For SVM training and testing we used the widely popular LibSVM library [14]. SVM performance may be strongly affected by parameter setting. We set the SVM parameters once only for each kernel using simple grid-search on a small range of values based on a random subset of the data.

3.2 Nearest Neighbor

Our 3-Nearest Neighbour classifier uses the common form of voting: for each sample to be classified the 3 nearest neighbours are identified. The sample is then assigned to the class from which the most neighbours come. Ties are resolved randomly on a first-to-come first-to-serve basis.

4. EXPERIMENTATION AND RESULTS

Our experiments were conducted to evaluate particularly the achiavable performance on independent data, as well as to evaluate the impact of feature selection.

4.1 The experimentation set

We have used the Chon-Kanade [8] database in order to evaluate our method. Firstly, we have applied the grid onto the faces of the database and therefore tracked the face as the facial expressions were evolving with the tracker in [15]. In our experiment we used only the *apex* phase of the expression. Overall we established 440 grid models: 35 for anger 35 for disgust 55 for fear 90 for happiness 65 for saddness 70 for surprise and finally 90 for neutral. We have conducted experiments for the 6 facial expressions.

your input here!!! For the experiments that you have done. I believe that it is better that you write this section as you run the experiments.

4.1.1 Convex Data Form

The feature space described in Section 2 allows ambiguous description of an angle between two given vectors - depending on the form of the mesh the angle between the same two vectors can be appear as convex or concave. We have found out that the classification performance improves if the angular data is normalized to what we call a *convex* form. This is ensured by applying the following transformation on all feature values:

$$\text{if } \theta > \pi, \text{let } \theta = 2 \cdot \pi - \theta$$

In most classifier – FS method combinations this *convex* form of the data allows notable improvement of classification accuracy.

4.2 Two-Tier Cross-Validation

The ratio between dataset dimensionality and number of available samples makes reliable classifier training a difficult task. Consequently, both classifier construction and the feature selection process should be performed in a way that prevents sufficiently the risk of over-training, which has been found an important factor in many tasks [16].

According to recent knowledge we resorted to test various feature selection set- ups and used 2-tier cross-validation to

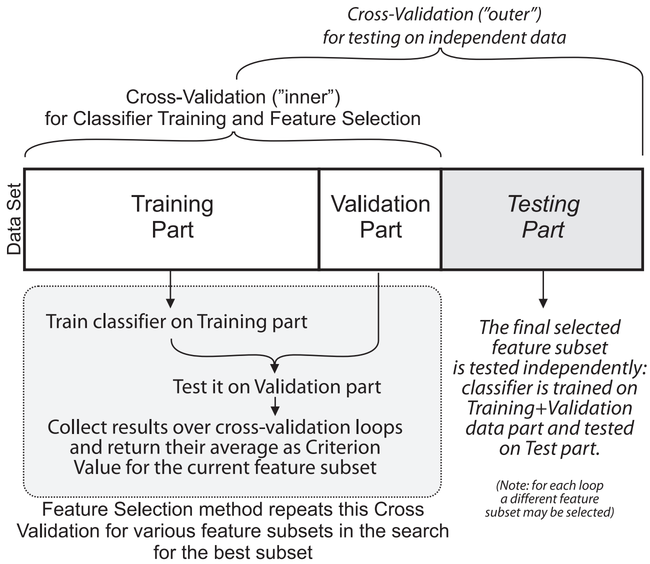

Figure 5: Two-Tier Cross- Validation to evaluate the impact of Feature Selection

evaluate as accurately as possible the effect of feature selection on classifier generalization ability, i.e., performance on unknown data. The process is ilustrated in Figure 5.

We run both the *inner* and *outer* cross-validation loop 3-fold. Finer splitting has been found to yield comparable results in this case at a cost of considerably longer computational time.

4.2.1 Bias Reduction Technique

Inspired by a technique known in Data Mining [17] we employed a technique of bias reduction, which is targeted at reducing the small sample problem especially in the 2-tier cross-validation described above. In the course of feature selection the classifier trained on Training data part is tested not only on Validation part but separately also on the Training part. The difference in obtained accuracies is considered a bias estimate. Before finalizing the feature selection criterion value after the end of one cross-calidation loop, both the Training and Validation parts are used once more at once to train and test the classifier. The averaged bias estimate is then subtracted from the obtained classifer accuracy to finalize the criterion value for the current feature subset.

Note: In this setting more data is utilized for classification performance evaluation (Training+Validation) than in the standard 2-tier cross-validation setting.

4.3 Results

The obtained results are summarized in Table 1. Two classification accuracies are reported for each classifier + FS method pair. First the accuracy as result of the optimization process is reported, i.e., the highest Wrapper criterion yielded by the feature selection method. Note that despite cross-validation, the result is not unbiased with respect to expected performance on unknown data. To obtain unbiased estimate of the classification performance on unknown data the second (or "outer") cross-validation loop is performed.

The best classification accuracy on unknown data (85%) was achieved using SVM with Linear kernel and IB feature

Table 1: Best Achieved Classification Accuracy

Classifier	FS Method	**Inner Cross-Validation** (FS Optimized Criterion) Mean	*St.Dv.*	**Outer Cross-Validation** (Independent Test) Mean	*St.Dv.*	Subset Size Mean	*St.Dv.*
3-NN	IB	84.3%	*1.48*	82.3%	*3.22*	78	*17.1*
	SFS	94.2%	*0.39*	79.4%	*4.78*	59.3	*4.1*
	SFFS	92.3%	*2.18*	76.5%	*3.54*	25.3	*2.1*
	all features	-	-	70.2%	*4.57*	642	-
SVM LIN	IB	85.9%	*0.98*	**85.1%**	*0.36*	57.3	*11.8*
	SFS	93.3%	*1.58*	81.1%	*2.42*	41	*9.2*
	SFFS	**94.6%**	*0.73*	82.5%	*2.53*	26.3	*5.2*
	all features	-	-	74.8%	*2.86*	642	-
SVM RBF	IB	78%	*1.40*	79.1%	*2.50*	281	*53.5*
	SFS	80.8%	*4.58*	73.6%	*1.73*	58.3	*1.7*
	SFFS	91%	*1.34*	76.8%	*4.22*	41.3	*1.7*
	all features	-	-	76.5%	*4.23*	642	-

selection method. The highest accuracy as obtained in the course of Wrapper feature selection was 94.6% using linear SVM + SFFS method.

Feature selection improves the independent results by up to $\approx 10\%$ over the full set of features.

Note: It should be noted that SVM performance depends to a great degree on parameter adjustment. In our experiments we have set the parameters using very simple heuristics. There is undoubtedly space for improvement of the overall performance of our system in this respect.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced a new method for automatic facial expressions recognition using a novel feature space. Accordingly we have used an SVM framework and a feature selection process which shows better results in the facial expressions recognition domain. The angular data feature space has not been used until now in the domain of model based facial expressions recognition at least to the authors' knowledge. Results as described in Section 4 show that the proposed method in many aspects improves over previously published results.

Angular data in model based techniques seems a promising solution towards the affine transformations robustness. It clearly solves issues in rotation, translation and scaling problems due to the nature of these features.

Acknowledgement: This work was supported by the Commision EU(XE) grant No. FP6-507752(MUSCLE Network of Excellence Project), and grants GA MŠk 1M0572, 2C06019, GA CR GA102/08/0593 in the Czech Republic.

REFERENCES

[1] P. Ekman, "Facial expression and emotion", *Personality: Critical Concepts in Psychology*, vol. 48, pp. 384–92, 1998.

[2] E. Hjelmas and B.K. Low, "Face detection: A survey", *Computer Vision and Image Understanding*, vol. 83, no. 3, pp. 236–274, 2001.

[3] G.R. Bradski et al., "Computer vision face tracking for use in a perceptual user interface", *Intel Technology Journal*, vol. 2, no. 2, pp. 12–21, 1998.

[4] W. Zhao, R. Chellappa, PJ Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey", *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[5] M. Pantie and LJM Rothkrantz, "Automatic analysis of facial expressions: the state of the art", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 12, pp. 1424–1445, 2000.

[6] I. Kotsia and I. Pitas, "Facial Expression Recognition in Image Sequences Using Geometric Deformation Features and Support Vector Machines", *Image Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 172–187, 2007.

[7] K. Kahler, J. Haber, and H.P. Seidel, "Geometry-based muscle modeling for facial animation", *Proc. Graphics Interface 2001*, pp. 37–46, 2001.

[8] T. Kanade, J.F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis", *Proceedings of the fourth IEEE International conference on automatic face and gesture recognition (FG00)*, pp. 46–53, 2000.

[9] M. Rydfalk, *Candide: a parameterised face*, Linkoping University, 1978.

[10] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski, "Classifying facial actions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, 1999.

[11] Ron Kohavi and George H. John, "Wrappers for feature subset selection", *Artif. Intell.*, vol. 97, no. 1-2, pp. 273–324, 1997.

[12] Pierre A. Devijver and Josef Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall, Englewood Cliffs, London, UK, 1982.

[13] Pavel Pudil, Jana Novovičová, and Josef Kittler, "Floating search methods in feature selection", *Pattern Recogn. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.

[14] Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at www.csie.ntu.edu.tw/ ~cjlin/libsvm.

[15] C. Tomasi and T. Kanade, "Detection and tracking of point features", *School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91–132*, 1991.

[16] Juha Reunanen, "Overfitting in making comparisons between variable selection methods", *J. Mach. Learn. Res.*, vol. 3, pp. 1371–1382, 2003.

[17] David J. Hand, Heikki Mannila, and Padhraic Smyth, *Principles of Data Mining*, MIT Press, 2001.