University of Szeged
Research Group on Artificial Intelligence

# Kernel-Based Feature Extraction and Speech Technology Applications

Summary of the PhD Thesis

by

**András Kocsor**

Advisor:

**Prof. Dr. János Csirik**

**Szeged**
**2003**

*"The problem of learning is arguably at the very core of the problem of intelligence, both biological and artificial."*

*T. Poggio and C. R. Shelton*

# Introduction

The booklet summarizes the scientific results of the author of the PhD dissertation entitled "Kernel-Based Feature Extraction and Speech Technology Applications". The dissertation concentrates on two key topics in artificial intelligence (AI): machine learning (ML) and its application to speech technology (ST).

Creating intelligent machines is an old dream of mankind. It was realized back in the middle of the last century that the construction of intelligent systems requires automatized learning and decision making [9; 22]. "Learning" in the machine learning sense means the application of the model method. That is, we aim at creating models that correctly simulate human thinking. The best way of doing this is to specify the model by means of a large amount of training patterns; decisions regarding a new pattern are made based on this model. Several fields of science like philosophy (as the science of sciences), physics, mathematics, biology, chemistry and theoretical computer science have all contributed to those tools that AI researchers build their models with. Such building blocks are, for example, short and long-term memory, hierarchical model construction, model hybridization, clustering, data-invariant methods, optimization and approximation. The results of this thesis are also based on the new advances of one such rapidly developing field called kernel methods. This notion had appeared in several fields of mathematics [11; 23] and mathematical physics before it became a key notion in machine learning. The basic idea behind the kernel technique was originally introduced for pattern recognition in [2] and was again employed in the general purpose Support Vector Machine [4; 27], which was followed by other kernel-based methods [14].

As the dissertation consists of two parts, the author's results will also be split into two parts. The *first group* of the results of the dissertation consist of the construction of feature extraction algorithms, applicable to machine learning problems. We discuss four linear feature extraction methods from a unified stand point [20]. Three of them – the Principal Component Analysis (PCA), the Independent Component Analysis (ICA) and the Linear Discriminant Analysis (LDA) – are well-known, while the Springy Discriminant Analysis (SDA) [19; 20] is based on a novel idea. By exploiting both the unified view of the linear methods and the non-linearization methodology that the kernel idea offers, we constructed the kernel counterparts of ICA, LDA and SDA. The resulting methods are known as Kernel-ICA [18], Kernel-LDA [17] and Kernel-SDA [19], respectively. The kernel generalization of PCA (Kernel-PCA) was proposed by Schölkopf et al. [25]. Actually, this work inspired the author to define the unified view for a set of linear methods which is suitable for the kernel-based non-linearization.

The topic of the *second group* of the thesis is the application of the methods of the first part to speech technology. We demonstrated the usefulness of the methods derived in the first part. We did so by performing phoneme classification tests in the framework of speech technology applications, namely the OASIS speech recognizer [15; 16; 20] and the "SpeechMaster" [17–19] speech therapy and reading teaching software package.

# Part I – Kernel-Based Feature Extraction

## The Kernel Idea

*Mercer kernels.* In the following we will assume that $\mathcal{X}$ is a compact set in the $n$-dimensional Euclidean space.

**Definition 1** *A function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a 'Mercer kernel', if and only if it is continuous, symmetric and positive definite.*

The notion of continuity and symmetry is well-known, but positive-definiteness is probably not. Hence we supply a definition of the latter in the following.

**Definition 2** *A function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive definite if for every finite set $\{\mathbf{x}_1, \ldots, \mathbf{x}_k\} \subset \mathcal{X}$ the $k \times k$ matrix $[\kappa(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^{k}$ is positive semi-definite.*

Finding out whether a function is continuous and symmetric is a relatively straightforward task. But checking its positive-definiteness is generally far from trivial.

*Kernel-induced feature spaces.* Now we will examine what kind of feature spaces the Mercel kernels implicitly induce and how these can be exploited in the non-linearization of certain types of algorithms. First we commence with the main theorem [6; 21].

**Theorem 1** *For a Mercer kernel $\kappa$ over $\mathcal{X} \times \mathcal{X}$ there exists a dot product space $\mathcal{F}$ with a map $\phi : \mathcal{X} \to \mathcal{F}$, such that for all $\mathbf{x}, \mathbf{z} \in \mathcal{X}$ $\kappa(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$.*

Usually $\mathcal{F}$ is called the kernel feature space and $\phi$ is the feature map. We have two immediate consequences. When $\phi$ is the identity, the function $\kappa(\mathbf{x}, \mathbf{z}) = \mathbf{x} \cdot \mathbf{z}$ (the simple dot product over the space $\mathcal{X}$) is symmetric, continuous and positive definite, so it constitutes a proper Mercer kernel. Going the other way, when applying a general Mercer kernel we can assume a space $\mathcal{F}$ over which we perform dot product calculations. This space and dot product calculations over it are defined only implicitly via the kernel function itself. The space $\mathcal{F}$ and map $\phi$ may not be explicitly known. We need only define the kernel function, which then ensures an implicit evaluation (see Fig. 1).

Based on Theorem 1, the essence of the *kernel trick* can be summarized as follows: *If the output of an algorithm is formulated in terms of a Mercer kernel, then alternative algorithms can be constructed by replacing the kernel with a different Mercer kernel.*

## Linear Feature Extraction

In most classification problems it is normal to view the objects to be classified as points in a feature space of proper dimensions. The space has to have a sufficient degree of freedom so that the object classes are sufficiently 'separable'. Making use of superfluous components, however, can confuse classification algorithms. A general practical observation is that it is worth slightly decreasing the dimensionality of the given feature space so that we can still guarantee that the overall structure of the data points remains intact. A simple way of doing this is by means of a linear transformation which linearly maps an initial feature space into a new feature space, usually one with fewer dimensions.

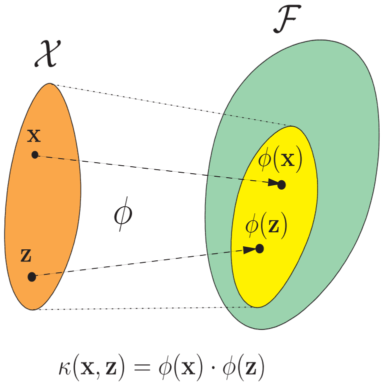


Figure 1: The "kernel-idea". The dot product in the kernel feature space $\mathcal{F}$ is implicitly defined.

*Introduction.* Now without loss of generality we shall assume that, as a realization of multivariate random variables, there are $n$-dimensional real attribute vectors in a compact set $\mathcal{X}$ over $\mathbb{R}^n$ describing objects in a certain domain, and that we have a finite $n \times k$ sample matrix $X = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ containing $k$ random observations. Actually, $\mathcal{X}$ constitutes the initial feature space and $X$ is the input data for the linear feature extraction algorithms which defines a linear mapping $h : \mathcal{X} \to \mathbb{R}^m$ for the extraction of a new feature set. The $m \times n$ $(m \leq n)$ matrix of the linear mapping – which may inherently include a dimension reduction – will be denoted by $V$.

With the linear feature extraction methods we search for an optimal matrix $V$, where the precise definition of optimality can vary from method to method. Although it is possible to define functions that measure the optimality of all the $m$ directions (i.e. the row vectors of $V$) *together*, here we will find each particular direction of the optimal transformations *one-by-one*, employing a $\tau : \mathbb{R}^n \to \mathbb{R}$ objective function for each direction separately. Intuitively, if larger values of $\tau$ indicate better directions and the chosen $m$ directions need to be independent in some ways, then choosing stationary points that have the $m$ largest function values is a reasonable strategy. Obtaining the above stationary points of a general objective function is a difficult global optimization problem. But if $\tau$ is defined by a Rayleigh quotient formulae, i.e.

$$\tau(\mathbf{v}) = \frac{\mathbf{v}^\top B_1 \mathbf{v}}{\mathbf{v}^\top B_2 \mathbf{v}}, \tag{1}$$

where $B_1$ and $B_2$ are symmetric $n \times n$ matrices, $B_2$ is positive definite – finding the solution is relatively quick and straightforward when formulated as a simple eigenvalue problem.

**Proposition 1** *The stationary points of $\tau(\mathbf{v})$ are precisely the eigenvectors of matrix $B_2^{-1} B_1$, and the corresponding eigenvalues are the values $\tau(\mathbf{v})$ takes at these points.*

Actually, the Rayleigh quotient-based approach offers a unified view of the linear transformation methods discussed in this section. These are Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA) and Springy Discriminant Analysis (SDA). Because two of the four methods to be discussed belong to the supervised family,[1] we should expect to handle the class labels. To do this let us assume as well that we have $r$ classes and an indicator function $\mathcal{L} : \{1, \ldots, k\} \to \{1, \ldots, r\}$, where $\mathcal{L}(i)$ gives the class label of the sample $\mathbf{x}_i$. Let $k_j$ further denote the number of vectors associated with label $j$ in the sample data.

[1] The two types of feature vector space transformations (supervised or unsupervised) can be distinguished by whether they utilize an indicator function containing the class information or not.

*The general concept of all four methods is summarized in the following:*

a) PCA concentrates on those independent directions with the largest variances [7; 13]. Normally in PCA the objective function $\tau$ for selecting new directions is defined by

$$\tau(\mathbf{v}) = \frac{\mathbf{v}^\top C \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}, \tag{2}$$

where

$$C = E\{(\mathbf{x} - E\{\mathbf{x}\})(\mathbf{x} - E\{\mathbf{x}\})^\top\} \tag{3}$$

is the sample covariance matrix. Here $E$ denotes the mean value and Eq. (2) defines $\tau(\mathbf{v})$ as the variance of the centralized sample vectors $\mathbf{x}_1 - E\{\mathbf{x}\}, \ldots, \mathbf{x}_k - E\{\mathbf{x}\}$ projected onto vector $\mathbf{v}/||\mathbf{v}||$.

b) ICA, besides keeping the directions independent, chooses directions along which the 'non-Gaussianity' is large [5]. The aim here is to linearly transform the input data into uncorrelated components, along which the distribution of the sample set is the least Gaussian (i.e. non-Gaussian). The reason for this is that along these directions the data is supposedly easier to classify. For optimal selection of the independent directions, several objective functions were defined using approximately equivalent approaches. We follow the way proposed by Hyvärinen [12]. In his FastICA algorithm for the selection of a new direction $\mathbf{v}$ the following $\tau$ objective function is used:

$$\tau_G(\mathbf{v}) = \left(E\{G(\mathbf{v} \cdot \mathbf{x})\} - E\{G(\nu)\}\right)^2, \tag{4}$$

where $G : \mathbb{R} \to \mathbb{R}$ is an appropriate non-quadratic function, $E$ again denotes the expectation value, $\nu$ is a standardized Gaussian variable and $\mathbf{v} \cdot \mathbf{x}$ is the dot product of the direction $\mathbf{v}$ and sample $\mathbf{x}$. Virtually, FastICA is an approximate Newton iteration procedure for the local optimization of the function $\tau_G(\mathbf{v})$. Before running the optimization procedure, however, the raw input data $X$ must first be preprocessed – by centering and whitening it. Since the standard Principal Component Analysis transforms the covariance matrix into a diagonal form [13], this can be done using a modified version of PCA, which transforms the covariance matrix to a unit matrix. After centering and whitening for every normalized $\mathbf{v}$ the mean of the projected sample is set to zero, and its variance is set to one. Moreover, for any matrix $W$ the covariance matrix of the linearly transformed, preprocessed points will remain a unit matrix if and only if $W$ is orthogonal. Finally, it is sufficient to look for a new orthogonal base $W$ for the preprocessed data, where the values of the non-Gaussianity measure $\tau_G$ for the base vectors are large. Note that since the data remains whitened after an orthogonal transformation, ICA can be considered an extension of PCA.

c) LDA prefers those directions along which the class centers are far away and the average variance of the classes is small [9; 10]. In the case of LDA the objective function $\tau : \mathbb{R}^n \to \mathbb{R}$ for selecting new directions depends not only on the sample data $X$, but also on the indicator function $\mathcal{L}$ owing to the supervised nature of this method. Let us define

$$\tau(\mathbf{v}) = \frac{\mathbf{v}^\top B \mathbf{v}}{\mathbf{v}^\top W \mathbf{v}}, \tag{5}$$

where $B$ is the *Between-class Scatter Matrix*, while $W$ is the *Within-class Scatter Matrix*. Here the *Between-class Scatter Matrix* $B$ shows the scatter of the class mean vectors $\mathbf{m}_j$ around the overall mean vector $\mathbf{m}$:

$$\begin{aligned} B &= \textstyle\sum_{j=1}^{r} \frac{k_j}{k}(\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^\top \\ \mathbf{m} &= \textstyle\frac{1}{k}\sum_{i=1}^{k} \mathbf{x}_i \\ \mathbf{m}_j &= \textstyle\frac{1}{k_j}\sum_{\mathcal{L}(i)=j} \mathbf{x}_i \end{aligned} \tag{6}$$

The *Within-class Scatter Matrix* $W$ represents the weighted average scatter of the covariance matrices $C_j$ of the sample vectors having label $j$:

$$\begin{aligned} W &= \textstyle\sum_{j=1}^{r} \frac{k_j}{k} C_j \\ C_j &= \textstyle\frac{1}{k_j}\sum_{\mathcal{L}(i)=j}(\mathbf{x}_i - \mathbf{m}_j)(\mathbf{x}_i - \mathbf{m}_j)^\top. \end{aligned} \tag{7}$$

d) SDA creates attractive forces between the samples belonging to the same class and repulsive forces between samples of different classes via springs & antisprings [20]. Then it chooses those directions along which the potential energy of the system is maximal. Let $\tau(\mathbf{v})$, the potential of the spring model along the direction $\mathbf{v}$, be defined by

$$\tau(\mathbf{v}) = \frac{\mathbf{v}^\top D\mathbf{v}}{\mathbf{v}^\top\mathbf{v}}, \tag{8}$$

where

$$D = \sum_{i,j=1}^{k} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top [M]_{ij} \tag{9}$$

and

$$[M]_{ij} = \begin{cases} -1, & \text{if } \mathcal{L}(i) = \mathcal{L}(j) \\ 1, & \text{otherwise} \end{cases} \qquad i,j = 1,\ldots,k. \tag{10}$$

Naturally, the elements of matrix $M$ can be initialized with values different from $\pm 1$ as well. It can be considered as a kind of force constant and can be set to a different value for any pair of data points.

## Non-linear Feature Extraction with Kernels

The approach of feature extraction could be either linear or non-linear, but it seems the kernel-idea is, in some sense, breaking down the barrier between the two types. As we have already seen if some linear method uses only the pairwise dot product of its input vectors during its computations then, just by altering the dot product operation in a proper way, we can create a non-linear version of it. The effect of the replacement of the operation is that the original linear method will implicitly be performed in a space of more (possibly even infinite) dimensions, and thus with a higher degree of freedom.

In the following this notion is used to derive non-linear counterparts of PCA, ICA, LDA and SDA. To this end let the dot product be defined by a Mercer kernel $\kappa$, which induces non-linearly a dot product space denoted by $\mathcal{F}$ via a feature map $\phi$ (see Theorem 1). First, let us examine the Rayleigh quotient of Eq. (1) in this space. Formally,

$$\tau(\boldsymbol{v}) = \frac{\boldsymbol{v}^\top \mathcal{B}_1 \boldsymbol{v}}{\boldsymbol{v}^\top \mathcal{B}_2 \boldsymbol{v}}, \quad \boldsymbol{v} \in \mathcal{F} \tag{11}$$

where $\mathcal{B}_1$ and $\mathcal{B}_2$ are symmetric matrices of size $\dim(\mathcal{F}) \times \dim(\mathcal{F})$, and $\mathcal{B}_2$ is positive definite. Unfortunately, this form is not restrictive enough so that we could express $\tau(\boldsymbol{v})$ as a function of the kernel $\kappa$.

Let us now observe that in the case of all the linear transformations the matrices in the nominator of the Rayleigh quotient are always a unique function of the sample matrix $X$. They all have the common form

$$X\Theta X^\top = \sum_{i=1}^{j} [\Theta]_{ij} \mathbf{x}_i \mathbf{x}_j^\top, \quad [\Theta]_{ij} \in \mathbb{R} \tag{12}$$

(cf. Eqs. (2), (5) and (8)), where $\Theta$ was a symmetric real matrix specific to the method. The matrix in the denominator is the identity matrix for PCA, ICA, SDA and, in the case of LDA, it is again of the form of Eq. (12) (see Eq. (7)). Based on this observation, the linear methods of the previous section take the following special Rayleigh quotient form

$$\frac{\mathbf{v}^\top X\Theta_1 X^\top \mathbf{v}}{\mathbf{v}^\top (X\Theta_2 X^\top + \delta I)\mathbf{v}}, \tag{13}$$

where $\mathbf{v} \in \mathcal{X}$, $X$ is the matrix containing the samples, $\Theta_1, \Theta_2$ are method-dependent real symmetric matrices of size $n \times n$, and $\delta \in \mathbb{R}_+$. In the case of LDA $\delta = 0$, and for PCA, ICA and SDA $\delta = 1$ and $\Theta_2$ is the zero matrix.

Now we can formalize the special Rayleigh quotient that corresponds to Eq. (13) in the kernel feature space. To do this we simply have to substitute $F = (\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_k))$ for $X = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$ and $\boldsymbol{v} \in \mathcal{F}$ for $\mathbf{v} \in \mathbb{R}^n$:

$$\tau(\boldsymbol{v}) = \frac{\boldsymbol{v}^\top F\Theta_1 F^\top \boldsymbol{v}}{\boldsymbol{v}^\top (F\Theta_2 F^\top + \delta I)\boldsymbol{v}}. \tag{14}$$

Now let us have a look at the stationary points of $\tau(\boldsymbol{v})$.

**Proposition 2** *$\boldsymbol{v} \in span(\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n))$ holds for all stationary points of $\tau(\boldsymbol{v})$.*

That is, we can assume that $\boldsymbol{v} = \alpha_1\phi(\mathbf{x}_1) + \cdots + \alpha_n\phi(\mathbf{x}_n) = F\boldsymbol{\alpha}$. With this we arrive at the following form of the Rayleigh quotient defined in Eq. (14), now depending on vector $\boldsymbol{\alpha}$:

$$\tau(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^\top F^\top F\Theta_1 F^\top F\boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top F^\top (F\Theta_2 F^\top + \delta I)F\boldsymbol{\alpha}}. \tag{15}$$

And because $F^\top F$ is the same as the kernel matrix $K = [\kappa(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^k$, we obtain the formula

$$\tau(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^\top K\Theta_1 K\boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top (K\Theta_2 K + \delta K)\boldsymbol{\alpha}}, \tag{16}$$

where, according to Proposition 1, the eigenvectors of the following matrix

$$(K\Theta_2 K + \delta K)^{-1} K\Theta_1 K \tag{17}$$

are the stationary points. If the row vectors of matrix $\mathcal{A}$ is defined as the eigenvectors corresponding to the dominant eigenvalues, then feature extraction for any new sample $\mathbf{z} \in \mathcal{X}$ can be performed using $\mathcal{A}F^\top\phi(\mathbf{z})$. Since $\phi$ is not directly known this expression cannot be readily applied. Now, let us notice that $F^\top\phi(\mathbf{z}) = (\kappa(\mathbf{x}_1, \mathbf{z}), \ldots, \kappa(\mathbf{x}_k, \mathbf{z}))^\top$, which offers a way for an implicit evaluation.

*Having obtained a uniform framework, we can proceed with the 'kernelized' methods, one after the other. The general concept of the methods is summarized in the following:*

a) Kernel-PCA [25] concentrates on those non-linear directions along which the variance of the data set is large. The $\tau$ function of Kernel-PCA is of the form (cf. [16; 20])

$$\tau(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^\top \frac{1}{k} K(I - \hat{I})K\boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}}, \tag{18}$$

where $[K]_{ij} = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel matrix and

$$\hat{I} = \frac{1}{k}\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}. \tag{19}$$

b) Kernel-ICA searches non-linearly for directions which are independent, and along which the distribution of the data significantly differs from a Gaussian one [18]. As we saw earlier, the ICA algorithm of Hyvärinen consists of two main parts: centering & whitening, and the subsequent approximate Newton algorithm. In these we extended non-linearly only the first phase, since after it we obtain uncorrelated data in the kernel feature space $\mathcal{F}$ in a non-linear way. However, although it could be done,[2] the second, iterative part of FastICA will not be non-linearized here.

c) Kernel-LDA performs non-linear feature extraction with the aim of class separation [17]. The classes are pushed apart while the data points belonging to the same class are pulled together. The $\tau(\boldsymbol{\alpha})$ function of Kernel-LDA has the form:

$$\frac{\boldsymbol{\alpha}^\top K(R - \hat{I})K\boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top K(I - R)K\boldsymbol{\alpha}}, \tag{20}$$

where $K$ is the kernel matrix, $\hat{I}$ is defined in Eq. (19) and

$$[R]_{ij} = \begin{cases} \frac{1}{k_t} & \text{if } t = \mathcal{L}(i) = \mathcal{L}(j) \\ 0 & otherwise. \end{cases} \tag{21}$$

d) Kernel-SDA non-linearly maps the initial feature space, and in the space obtained it seeks to separate classes just as Kernel-LDA does, but by means of defining attractive and repulsive forces (see Fig. 2) [19]. The Rayleigh quotient for Kernel-SDA has the form:

$$\tau(\boldsymbol{\alpha}) = 2\frac{\boldsymbol{\alpha}^\top K(\tilde{M} - M)K^\top\boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top K\boldsymbol{\alpha}}, \tag{22}$$

where $K$ is the kernel matrix and $\tilde{M}$ is a diagonal matrix with the sum of each row of the force constant matrix $M$ in the diagonal.

[2] Obviously, Eq. (4) could be very easily non-linearized using kernels, as the formula contains only one dot product. We chose to disregard this step because it did not fit into our Rayleigh quotient-based non-linearization approach.

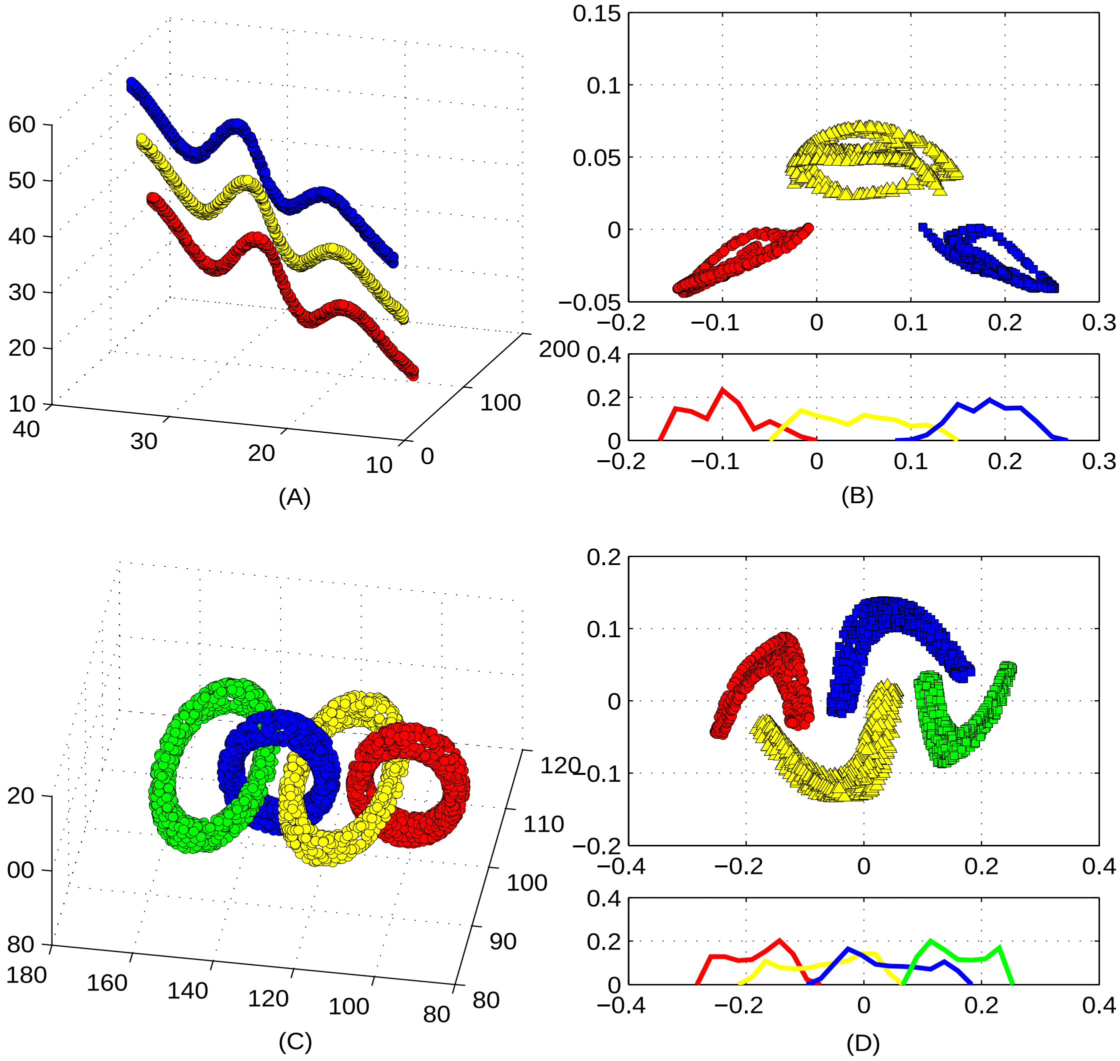


Figure 2: The effect of Kernel-SDA on 3D data sets. Figs. (A) and (C) depict 3-dimensional data sets. Their distributions after Kernel-SDA in 2 dimensions are shown in Figs. (B) and (D), respectively.

## *Thesis I/1*

*Eight feature extraction algorithms, 4 linear (PCA, ICA, LDA, SDA) and 4 non-linear (Kernel-PCA, Kernel-ICA, Kernel-LDA, Kernel-SDA) are discussed by the author – in a uniform framework – via the optimization of Rayleigh quotient formulas [15–20]. The 4 non-linear methods are derived by non-linearizing the corresponding linear algorithms applying the so-called kernel non-linearization technique.*

## *Thesis I/2*

*The author constructed a novel linear method called SDA [20], which fits in nicely with 3 linear methods (PCA, ICA, LDA) well-known from the literature.*

## *Thesis I/3*

*Making use of the kernel idea the author non-linearized the ICA, LDA and SDA linear algorithms. This resulted in the non-linear methods Kernel-ICA [18], Kernel-LDA [17] and Kernel-SDA [19].*

# Part II – Speech Technology Applications

## Speech Recognition

Automatic speech recognition is a special pattern classification problem which aims to mimic the perception and processing of speech in humans. For this reason it clearly belongs to the fields of machine learning and artificial intelligence. For historical reasons, however, it is mostly ranked as a sub-field of electrical engineering, with its own unique technologies, conferences and journals. In the last two decades the dominant method for speech recognition has been the hidden Markov modeling approach. In the meantime, the theory of machine learning has developed considerably and now has a wide variety of learning and classification algorithms for pattern recognition problems [3; 8; 10]. The primary goal of this chapter is to study the applicability of some of these methods to phoneme classification.

When choosing the directions of our speech recognition research, we decided to focus on Hungarian with the hope that we could address some special issues concerning the processing of our national language, and also that we could make use of our previous experience with NLP for Hungarian. Furthermore, we were looking for a flexible framework that allows experimentation with different preprocessing techniques, feature extraction methods and machine learning algorithms. These expectations led us to the stochastic segmental approach which, in a certain sense, can be viewed as an extension of hidden Markov modeling. Our recognition system, OASIS [26], was designed to be as modular as possible, so we can easily conduct experiments by combining different techniques for the several subtasks of recognition.

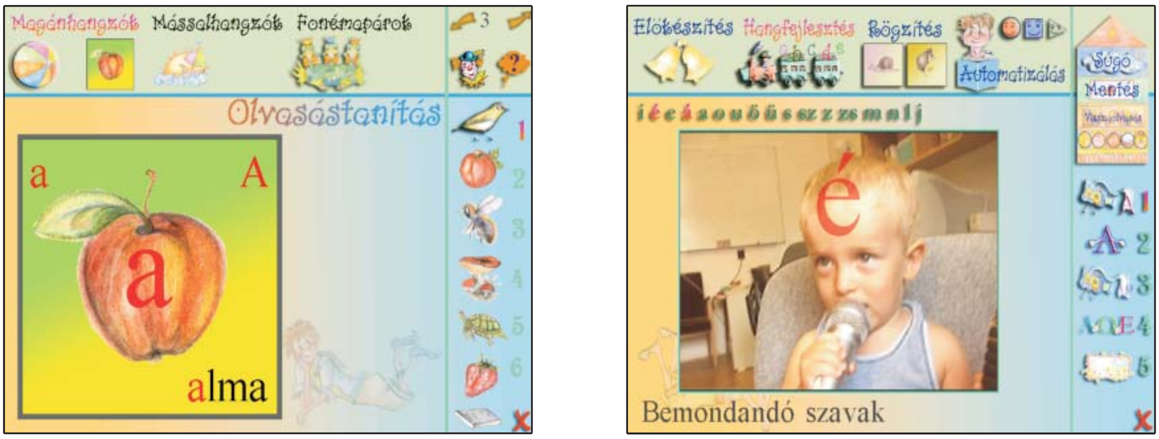
In the thesis we designed and performed a segmental phoneme classification tests within the framework of the OASIS speech recognizer [15; 16; 20]. The goal of this test was to examine how the feature extraction algorithms - when combined with classification algorithms (such as Timbl, OC1, C4.5, GMM, ANN) - influence the classification accuracy.

Examining the results of the many tests performed, we may state that, in the hope of better classification, it is worth applying feature extraction algorithms prior to learning.

## Phonological Awareness Teaching

An important clue to the process of learning to read in alphabet-based languages is the ability to separate and identify consecutive sounds that make words and to associate these sounds with its corresponding written form [1; 24]. To learn to read in a fruitful way young learners must, of course, also be aware of the phonemes and be able to manipulate them. Many children with learning disabilities have problems in their ability to process phonological information. Furthermore, phonological awareness teaching has also great importance for the speech and hearing handicapped, along with evolving the corresponding articulatory strategies of tongue movement.

The "SpeechMaster" software (Fig. 3) developed by our team seeks to apply speech recognition technology to speech therapy and the teaching of reading. Both applications require a real-time response from the system in the form of an easily comprehensible visual feedback [17]. With the simplest display setting feedback is given by means of flickering letters, their identity and brightness being adjusted to the speech recognizer's output. In speech therapy we try to supplement the missing



(A)

(B)

Figure 3: Screenshots of the "SpeechMaster" phonological awareness teaching system. (A-B) The teaching reading part and the speech therapy part, respectively.

auditive feedback of the hearing impaired (Fig. 3B), while in teaching reading it is necessary to reinforce the correct association between the phoneme-grapheme pairs (Fig. 3A). With the aid of a computer children can practice without the need for the continuous presence of the teacher. This is very important because the therapy of the hearing impaired requires a long and tedious fixation phase. Furthermore, experience shows that most children prefer computer exercises to conventional drills.

In the tests performed in this thesis within the "SpeechMaste" software package we again studied how the combination of feature extraction algorithms with classifiers (ANN, PPL, GMM, SVM) affects classification [17–19]. We found that non-linear transformations in general lead to a better classification than the non-linear ones, and thus are a promising new direction for research. We also found that the supervised transformations are usually better than the unsupervised ones. These transformations greatly improved our phonological awareness teaching system by offering a robust and reliable real-time phoneme classification.

## Thesis II/1

*The author designed and, with the help of his colleagues, executed several segmental phoneme classification tests within the framework of the OASIS speech recognizer [15; 16; 20]. The goal of these tests was to study how the feature extraction methods affect classification performance. Besides the design of the tests, the implementation and running of the feature extraction algorithms was exclusively his own work.*

## Thesis II/2

*To improve the real-time phoneme classification accuracy of the "SpeechMaster" speech therapy, teaching reading and reading therapy software package, the author designed and conducted several further classification tests along with his colleagues[3] [17–19]. The subtasks of these tests were shared in exactly the same way as specified in Thesis II/1.*

[3] Although during the last couple of years the author has been the project and research manager of both the OASIS and "SpeechMaster" projects, he does not consider the two systems be results of this dissertation.

| $\mathcal{N}o.$ | PCA | ICA | LDA | SDA | Kernel-PCA | Kernel-ICA | Kernel-LDA | Kernel-SDA | framework |
|---|---|---|---|---|---|---|---|---|---|
| [15] | ● | ● | ● |  |  |  |  |  | OASIS |
| [16] | ● |  |  |  | ● |  |  |  | OASIS |
| [17] |  |  | ● |  |  |  | ● |  | SpeechMaster |
| [18] |  | ● |  |  |  | ● |  |  | SpeechMaster |
| [19] |  |  |  |  |  |  |  | ● | SpeechMaster |
| [20] | ● | ● | ● | ● | ● | ● | ● | ● | OASIS |

Table 1: The relation between the thesis topics and the corresponding publications.

# Conclusions

The kernel idea and the linear and non-linear feature extraction methods presented in the dissertation demonstrate that perhaps the gap between linear and non-linear models is not that big at all. More precisely, a subset of the non-linear models may be linear but in another space.

The results of the speech technology applications demonstrated that, in order to increase classification performance, it is worth decomposing the classification problem into a feature extraction and a learning step. Albeit that both the feature extraction and the learning algorithms aim to separate the classes, performing it in two steps usually proves more efficient.

Finally, Table 1 summarizes which publication covers which method of the thesis and which software environment was used in tests carried out.

# References

[1] M. J. Adams, Beginning to read: Thinking and learning about print, Cambridge, MA: MIT Press, 1990.

[2] M. A. Aizerman, E. M. Braverman, L. I. Rozonoer, "Theoretical foundation of the potential function method in pattern recognition learning," Automat. Remote Cont., Vol. 25, pp. 821-837, 1964.

[3] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford Univerisity Press Inc., New York, 1996.

[4] B. E. Boser, I. M. Guyon, V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in Proc. of the Fifth Annual ACM Conference on Computational Learning Theory, D. Haussler (eds.), ACM Press, Pittsburg, pp. 144-152, 1992.

[5] P. Comon, "Independent component analysis, A new concept?" Signal Processing, Vol. 36, pp. 287-314, 1994.

[6] F. Cucker, S. Smale, "On the mathematical foundations of learning," Bull. Am. Math. Soc., Vol. 39, pp. 1-49, 2002.

[7] K. I. Diamantaras, S. Y. Kung, Principal Component Neural Networks: Theory and Applications, John Wiley, New York, 1996.

[8] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, John Wiley & Sons, New York, 2001.

[9] R. A. Fisher, "The use of multiple measurements in taxonomic problems," Annals of Eugenics, Vol. 7, pp. 179-188, 1936.

[10] K. Fukunaga, Statistical Pattern Recognition, Academic Press, New York, 1989.

[11] D. J. Hand, Kernel discriminant analysis, Research Studies Press, New York, 1982.

[12] A. Hyvärinen, E. Oja, "A fast fixed-point algorithm for independent component analysis," Neural Computation, Vol. 9, No. 7, pp. 1483-1492, 1997.

[13] I. J. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.

[14] Kernel Machines Web site, http://kernel-machines.org.

[15] A. Kocsor, L. Tóth, A. Kuba Jr., K. Kovács, M. Jelasity, T. Gyimóthy, J. Csirik, "A Comparative Study of Several Feature Space Transformation and Learning Methods for Phoneme Classification", International Journal of Speech Technology, Vol. 3, No. 3/4, pp. 263-276, 2000.

[16] A. Kocsor, A. Kuba Jr., L. Tóth, "Phoneme Classification Using Kernel Principal Component Analysis", Periodica Polytechnica, Vol. 44, No. 1, pp. 77-90, 2000.

[17] A. Kocsor, L. Tóth, D. Paczolay, "A Nonlinearized Discriminant Analysis and its Application to Speech Impediment Therapy", in: V. Matousek, P. Mautner, R. Moucek, K. Tauser (Eds.): Proceedings of Text, Speech and Dialogue: 4th International Conference, TSD 2001, LNAI 2166, pp. 249-257, Springer Verlag, 2001.

[18] A. Kocsor, J. Csirik, "Fast Independent Component Analysis in Kernel Feature Spaces", in: L. Pacholski and P. Ruzicka (Eds.): Proceedings of SOFSEM 2001: Theory and Practice of Informatics: 28th Conference on Current Trends in Theory and Practice of Informatics, LNCS 2234, pp. 271-281, Springer Verlag, 2001.

[19] A. Kocsor, K. Kovács, "Kernel Springy Discriminant Analysis and Its Application to a Phonological Awareness Teaching System", in: P. Sojka, I. Kopecek, K. Pala (Eds.): Proceedings of Text, Speech and Dialogue: 5th International Conference, TSD 2002, LNAI 2448, pp. 325-328, Springer Verlag, 2002.

[20] A. Kocsor, L. Tóth, "Application of Kernel-Based Feature Space Transformations and Learning Methods to Phoneme Classification", accepted for Applied Intelligence.

[21] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," Philos. Trans. Roy. Soc. London, A, Vol. 209, pp. 415-446, 1909.

[22] J. V. Neumann, O. Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, 1947.

[23] E. Parzen, "On estimation of probability density function and mode", Annals of Mathematical Statistics, Vol. 33, pp. 1065-1076, 1962.

[24] D. J. Sawyer, B. J. Fox, Phonological Awareness in Reading: The Evolution of Current Perspectives (Springer Series in Language and Communication, Vol 28), Springer-Verlag, New York, 1991.

[25] B. Schölkopf, A. J. Smola, K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Computation, Vol. 10, pp. 1299-1319, 1998.

[26] L. Tóth, A. Kocsor, K. Kovács, "A Discriminative Segmental Speech Model and its Application to Hungarian Number Recognition," in: P. Sojka, I kopecek, K. Pala (Eds.), Proceedings of Text, Speech and Dialogue: 3th International Conference, TSD 2000, LNAI 1902, pp. 307-313, Springer Verlag, 2000.

[27] V. N. Vapnik, Statistical Learning Theory, John Wiley & Sons Inc., 1998.