MACHINE LEARNING METHODS IN BIOMARKER ANALYSIS AND ITS APPLICATION ON MOBILE DEVICES

GÉPI TANULÁSON ALAPULÓ ELJÁRÁSOK BIOMARKEREK ELEMZÉSÉHEZ ÉS ALKALMAZÁSUK MOBIL ESZKÖZÖKÖN

Ph.D. Thesis Booklet

Luca Szegletes

Advisor:
Bertalan Forstner, Ph.D.

BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DEPARTMENT OF AUTOMATION AND APPLIED INFORMATICS

Budapest
2016.
Luca Szegletes

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

1117 Budapest, Magyar Tudósok körútja 2. QB-207.

e-mail: luca@aut.bme.hu
tel: +36(1)4631668

Advisor:
Bertalan Forstner, Ph.D.
1 Introduction

Finding and analyzing biomarkers is becoming a widely investigated topic among researchers studying cognitive functions and dysfunctions. This field is rich in interdisciplinary approaches, whereby different viewpoints contribute to understanding the relationship between the brain, humans and anything machine-like.

Classification methods are rapidly becoming a popular tool for discriminating one experimental group from another, using fMRI data. However, the performance of these methods is often poor when the data used includes fMRI data collected at more than one site. This is, in part, due to the differences in scanner hardware and data acquisition protocols across sites.

Here, I address this problem by applying spectral methods and optimization techniques to the data as a preprocessing step. The presented approaches resulted in an increase in cross-site classification and multi-site classification performance for discriminating ADHD children from their neurotypical peers based on their intrinsic connectivity patterns, using resting-state fMRI data acquired at three different sites.

The projection of the data (and classifier information) in multi-site studies is invaluable when one site has significantly less or unbalanced data than the other, this is a reoccurring problem in several sub groups like autistic girls. The main aim is to improve significantly the classification especially on these sites with less data available by pooling additional samples from other sites.

It is hypothesized that the groups would be better distinguished using the presented framework, compared to conventional approach, the experimental results demonstrate the efficacy of the presented methods in accurately discriminating one experimental group from another, using multi-site fMRI data.

In order to analyze further cognitive dysfunctions, a self-rewarding, biofeedback framework was designed in the third part of my dissertation. Multi-subject mental effort classification faces similar problems as described before in multi-site imaging studies, whereby the algorithm must categorize physiological features measured from multiple subjects who undergo the same mental task. Due to inter-subject differences, a system trained from a subject may not be directly applicable to others, and a significant amount of time is required to re-calibrate the classifier to a new subject.

Gamification of learning material has received much interest from researchers in the past years. The third part of the dissertation attempts to further improve such learning experience by applying socio-cognitive gamification to educational games. Dynamic difficulty adjustment (DDA) is a well-known tool in optimizing gaming experience. It is a process to control the parameters in a video game automatically based on user experience in real-time. This method can be extended by using a biofeedback-approach, where certain aspects of the player’s ability is estimated based on physiological measurements (e.g. eye tracking, ECG, EEG). Here, I outline the design of a biofeedback-based framework that supports dynamic difficulty adjustment in educational games. It has a universal architecture, so the concept can be employed to engage users in non-game contexts as well. The framework accepts input from the games, from the physiological sensors and from the so-called supervisor unit. This special unit empowers a new social aspect by enabling another user to observe or intervene during the interaction. To explain the game-user interaction itself in educational games I also propose a hybrid model. One key element of this framework is the multi-source classification methods presented in the previous theses.

Based on these, the objectives of my research are the following:
• Present spectral algorithms to solve the multi-site classification problem primary in imaging studies.

• Propose a representation model of multi-site classification problem with algorithms based on mathematical optimization.

• Create a self-rewarding, biofeedback framework, where the same algorithms can be applied in estimating mental effort with multiple sensors. Investigate how this framework can optimize learning efficiency.

2 Methodological summary

The method of my research was determined by the listed objectives. The basis of my dissertation consists of several areas of the related literature, like Principal Components Analysis, Independent Component Analysis, Linear Discriminant Analysis and Affine Invariant Classifiers. Mostly, I used probability theoretical and optimization methods to model and solve the multi-site classification problem.

Firstly, two spectral-based approaches were used to eliminate site heterogeneity, where I assume that the distributions are close to each other in a sense: (i) one of the methods is based on an extension of Principal Component Analysis to transform and match the data between the sources, (ii) the other applies Linear Discriminant Analysis in order to project the data as a preprocessing step. During constructing this mathematical framework, my objective was to provide an easy aggregation of datasets from multiple imaging sites.

Second, I constructed a more specific representation, where the inter-site differences are modeled with an affine transformation. With this, I portray the multi-site classification problem as an optimization task, where the aim is to find the best transformation operator between the datasets applying different cost functions.

Lastly, I have built a self-rewarding, biofeedback system, where I applied multiple physiological sensors. I designed a general architecture of game-user interaction based on reinforcement learning and used a basic control theory approach to model learning performance through computer games. Secondly, mental states are modeled in a game tree according to the changes in difficulty and performance. I verified my model with multiple user studies, where widely acknowledged Human-Computer Interaction principles were incorporated in a laboratory environment (Bells test, N-back task, Raven test and an Interest Map).

During my research, the suitability for real-world applications was an important objective. The proposed methods were tested in simulated environments written mainly in MatLAB and the framework was developed on Android operation system.

3 New theoretical results

My results are summarized in three theses, presented briefly in the following.

Thesis I.: Spectral Methods for Multi-Source Classification Problem

Publications related to this thesis are [1][3][5][7]
In my first thesis I propose algorithms based on spectral methods to solve the multi-source problem. The first approach uses Principal Component Analysis to transform and match the data between the sources (subthesis I.1.). The second algorithm uses Linear Discriminant Analysis to project the data to a common subspace for the classification (subthesis I.2.). Both methods are assuming that the source distributions are close to each other in a sense. I demonstrate on simulated and imaging data that both algorithms solve the multiple source heterogeneity problem.

The measurements are represented as an observation from a random vector.

**Definition 3.1 (Random vector).** \((X, Y)\) is a random vector of observations:

\[
(X, Y) = (X^1, X^2, \ldots, X^P, Y)
\]

where \((X^1, X^2, \ldots, X^P) \in \mathbb{R}, Y \in \{0, 1\}\) and \(P\) denotes the number of features.

The subjects are either children with typical development (TD) or with ADHD (denoted with label 0 and 1).

**Definition 3.2 (Data matrix).** \((X, Y)\) is the data matrix containing the observations of the random vector in each class \((X)\) and the corresponding labels \((Y)\):

\[
(X, Y) = \begin{bmatrix}
    x^1_1 & x^2_1 & \ldots & x^P_1 & y_1 \\
    x^1_2 & x^2_2 & \ldots & x^P_2 & y_2 \\
    \vdots & \vdots & \ddots & \vdots & \vdots \\
    x^1_n & x^2_n & \ldots & x^P_n & y_n
\end{bmatrix}
\]

where \(n\) is the number of samples.

**Definition 3.3 (Source).** A source represents the data space that the variables are spanning, for example these are the sites in fMRI studies.

**Definition 3.4 (Data matrices from source 1 and source 2).** \((X, Y)\) and \((\tilde{X}, \tilde{Y})\) stand for the data matrices containing the random observations from different sources respectively.

Here, the slightest a priori information is assumed regarding the relationship between \((X, Y)\) and \((\tilde{X}, \tilde{Y})\) such that there are sampled from distributions close to each other in one sense.

Two different classification problems are defined based on sampling the training and testing set. Firstly, the classifier is trained on \((X, Y)\) from source 1, and \((\tilde{X}, \tilde{Y})\) is chosen as test set from source 2.

**Definition 3.5 (Cross-source Classification).** The cross-source classification problem is defined on source 1 and source 2 as follows:

Building a classifier \(f : \mathbb{R}^P \rightarrow \{0, 1\}\) based on a set of samples from \(X\) and corresponding labels \(Y\).

The task is to use this classifier to test \((\tilde{X}, \tilde{Y})\) from source 2.

In the second group of approaches, the aim is to improve the classification rate with pooling extra data from a different source. First, \(X\) is divided to a training set and a test set, here denoted by \(X_{tr}, X_{test}\) respectively. The classifier is trained on mixed samples from \((X_{tr}, Y_{tr})\) and from the pooled \((\tilde{X}, \tilde{Y})\). Finally, the classifier is tested on \(X_{test}\).
Definition 3.6 (Multi-source Classification). The multi-source classification problem is defined on source 1 and source 2 as follows:

Building a classifier \( f : \mathbb{R}^P \to \{0, 1\} \) based on a set of samples from \( \mathbf{X}_{tr}, \tilde{\mathbf{X}} \) and corresponding labels \( \mathbf{Y}_{tr}, \tilde{\mathbf{Y}} \).

Use this classifier to test on \( \mathbf{X}_{test} \).

Subthesis I.1.: Matching Principal Component Analysis

I proposed an algorithm that uses Principal Components Analysis and the Hungarian matching algorithm to decrease source differences, when the distributions are close to each other and assuming linear separability. I have shown with experimental results that the proposed algorithm allows higher precision in classification, than the baseline method.

In the first approach, the well-known Principal Component Analysis (PCA) algorithm is extended in order to solve the classification problems.

Figure 1: Cross-source Classification Analysis Flowchart with Matching PCA. In the first step, the algorithm applies a PCA on each dataset from the sites, separately. The selected classifier is trained on one dataset and the others are used as test data. Secondly, the optimum number of principal components is identified for each dataset (selecting an equal number of principal components), in order to represent the data in a reduced dimensional space. Thirdly, between the principal components from each site a feature matching algorithm is performed to select the optimal pairs of vectors. According to this matched pairs, the order of the principal components is changed in the test data. The training and test data were transformed by projecting the original data on to the matched PCA component space. Finally, a classifier is trained using transformed training data and the same classifier used on the test dataset.

The Matching PCA has three main steps in cross-source classification, which are summarized in Fig. 3.

Definition 3.7 (Average). Define \( \mathbf{m} \) as the average of \( \mathbf{X} \):

\[
\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_i
\]

(3.3)

where \( n \) is the number of data points and \( \mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iP}) \) denotes the \( i \)th row of \( \mathbf{X} \).
Definition 3.8 (Empirical covariance matrix). Define $C$ as a $P \times P$ covariance matrix calculated from the data $X$:

$$C_{(i,j)} = \frac{1}{n} \sum_{k=1}^{n} x_i^k x_j^k - \left( \frac{1}{n} \sum_{k=1}^{n} x_i^k \right) \left( \frac{1}{n} \sum_{k=1}^{n} x_j^k \right)$$

(3.4)

where $n$ is the number of variables.

Before applying any transformation to the data, it is normalized to zero mean and unit variance.

Definition 3.9 (Eigenvector matrix). $W$ corresponds to the matrix containing the eigenvectors (the principal components, its $i$th vector is the eigenvector of the empirical covariance matrix $C$) as column vectors sorted by the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K$:

$$W = (w_1, w_2, \ldots, w_K)$$

(3.5)

where $K$ is the number of selected eigenvectors of the features $P$ ($K \leq P$). From each source, the same number $K$ of principal component are selected.

PCA is based on a projection that linearly transforms each input vector $x_i$ (where $i = 1, 2, \ldots, n$ and $n$ corresponds to the number of samples) into a new vector $x_i'$:

$$x_i' = W^T x_i$$

(3.6)

Definition 3.10 (Distance of eigenvectors). A correlation based distance is defined between eigenvectors $w_i, w_j$ as:

$$d(w_i, w_j) = 1 - corr(w_i, w_j) = 1 - \frac{(w_i - \overline{w})(w_j - \overline{w})^T}{\sqrt{(w_i - \overline{w})(w_i - \overline{w})^T} \sqrt{(w_j - \overline{w})(w_j - \overline{w})^T}}$$

(3.7)

where $d(w_i, w_j)$ denotes the distance between $w_i$ and $w_j$, and $\overline{w}$ is the average of the coordinates of $w_i$.

Definition 3.11 (Distance matrix). Let $W, \tilde{W}$ be the matrices containing the eigenvectors $w_i, \tilde{w}_i$ of $C, \tilde{C}$ and define the distance between these matrices $W, \tilde{W}$ as:

$$D(W, \tilde{W}) = \begin{bmatrix}
    d(w_1, \tilde{w}_1) & d(w_1, \tilde{w}_2) & \cdots & d(w_1, \tilde{w}_K) \\
    \vdots & \vdots & \ddots & \vdots \\
    d(w_K, \tilde{w}_1) & d(w_K, \tilde{w}_2) & \cdots & d(w_K, \tilde{w}_K) \\
    \vdots & \vdots & \ddots & \vdots \\
    d(w_K, \tilde{w}_1) & d(w_K, \tilde{w}_2) & \cdots & d(w_K, \tilde{w}_K)
\end{bmatrix}$$

(3.8)

where $D(W, \tilde{W})$ is a matrix and its elements are the pairwise distances of eigenvectors $d(w_i, \tilde{w}_j)$.

Secondly, based on a distance matrix, the basis of the subspace spanned by the orthogonal principal components (eigenvectors) in each test dataset, are matched to the components of the training data, using the Hungarian algorithm.

7
**Definition 3.12** (Minimum set of $D(W, \tilde{W})$). $D(W, \tilde{W})$ denotes the distance of $W, \tilde{W}$ matrices constructed by the eigenvectors. Its minimum set is based on the pairwise distances, a set is minimal. We search for a permutation $\pi$ of the eigenvectors in $\tilde{W}$ to minimize the sum of the pairwise distances as:

$$\arg\min_{\pi} \sum_{i=1}^{K} D(W, \tilde{W})_{i,\pi(i)}$$  \hspace{1cm} (3.9)

Third, the testing data is projected into this new space, which is defined by its re-ordered principal components.

**Definition 3.13** (Permuted eigenvector matrix). $W_\pi$ corresponds to the matrix containing the eigenvectors applying the permutation $\pi$:

$$W_\pi = (w_{\pi(1)}, w_{\pi(2)}, \ldots, w_{\pi(K)})$$  \hspace{1cm} (3.10)

where $K$ is the number of selected eigenvectors of the features $P$ ($K \leq P$).

In the cross-source method, the matching PCA is applied to train the classifier on the transformed $W^\top X$ from **source 1** and test on the transformed $\tilde{W}_\pi^\top \tilde{X}$ from **source 2**.

**Algorithm 1** Matching PCA in Cross-Source Classification

1: Input: $X$ and $\tilde{X}$ data matrices  
2: Apply Principal Component Analysis on $X$, $\tilde{X}$  
3: Compute $W$, $\tilde{W}$, eigenvector matrices from the covariance matrices $C$, $\tilde{C}$  
4: Compute $D(W, \tilde{W})$ distance matrix representing pairwise distances between the principal components  
5: Apply the Hungarian Algorithm to solve the assignment problem and find the minimal distance set between $W$, $\tilde{W}$  
6: Apply a permutation to the principal components in order to match them one by one according to the minimal set: $W$, $\tilde{W}_\pi$  
7: Train classifier on $W^\top X$  
8: Test on $\tilde{W}_\pi^\top \tilde{X}$

In the multi-source method, the same matching PCA is used, but the classifier is trained on data mixing the training samples from $W^\top X$ from **source 1** with $\tilde{W}_\pi^\top \tilde{X}$ from the pooled **source 2**. Lastly, the classifier is tested on the testing samples from **source 1** $W^\top X$.

Experimental results show that the classification performance improved significantly and is suitable to solve the multi-site imaging problem.

**Subthesis I.2.: Transformation with Linear Discriminant Analysis**

*I have designed a method based on linear discriminant analysis to transform the data into a subspace to improve the classification performance assuming that the distributions are close to each other.*

Figure 3 shows the main steps of this approach.

As in the Matching PCA, the aim of applying LDA on the sources is to find a projection matrix as follows
In the first step, the algorithm applies a Linear Discriminant Analysis on each dataset from the sites separately. The eigenvectors are identified for each dataset in order to represent the data in a reduced dimensional space. The training and test data are transformed by projecting the original data onto the LDA component space.

\[
X'_i = W^T X_i, \quad (3.11)
\]

In the first step, the $P$-dimensional mean vectors are computed of the classes in $X, Y$ and $\tilde{X}, \tilde{Y}$ respectively.

**Definition 3.14 (Group data matrices).** Let $X_0, X_1$ be the divided data matrices of $X$ according to the class labels $Y$ ($X_0$ consists those rows of $X$ that corresponds to the labels 0).

**Definition 3.15 (Group averages).** A $m_0, m_1$ defines group averages:

\[
m_0 = \frac{1}{n_0} \sum_{i=1}^{n_0} x_i, \quad m_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} \tilde{x}_i, \quad (3.12)
\]

where $n_0$ is the number of samples in class $X_0$ and $n_1$ is number of samples in class $X_1$.

LDA aims to maximize the between class distance and minimize the within class distance in the dimensionality-reduced space solving generalized eigenvalue problem $S_w, S_b$ for the scatter matrices (between-class and within-class scatter matrix). For a binary classification problem where there is only two classes, LDA provides just one discriminant direction. In order to achieve additional discriminant directions, the first discriminant direction is taken out of data and it is projected onto the largest subspace orthogonal to the first discriminant to find the next direction. The linear transformation is applied to each row of the data matrix.

\[
X' = W^T X \quad (3.13)
\]

\[
\tilde{X}' = \tilde{W} \tilde{X} \quad (3.14)
\]
In the cross-source method, the classifier is trained on $W^\top X$ from source 1 and it is tested on $\tilde{W}^\top \tilde{X}$ from source 2.

**Algorithm 2** Cross-Source Classification with LDA as a Transformation Method

Input: $(X, Y)$ and $(\tilde{X}, \tilde{Y})$ data matrices

1. Compute group average $m_0, m_1$ for $X$
2. Compute group average $\tilde{m}_0, \tilde{m}_1$ for $\tilde{X}$
3. Compute between-class and within-class scatter matrices $S_w$ and $S_b$ for $X$
4. Compute between-class and within-class scatter matrices $\tilde{S}_w$ and $\tilde{S}_b$ for $\tilde{X}$
5. Compute the eigenvector matrix $W$
6. Compute the eigenvector matrix $\tilde{W}$
7. Train classifier on $W^\top X$
8. Test classifier $\tilde{W}^\top \tilde{X}$

In the multi-source method LDA is applied similarly with the aim to project the data to a new, common space but the classification procedure is different.

Experimental results show that the classification performance improved significantly and is suitable to solve the multi-site imaging problem.

**Thesis II.: Optimization Methods for Multi-Source Classification Problem**

Publications related to this thesis are [4][5][7][10].

My second thesis proposes optimization based approaches to solve the cross-source and multi-source classification problems with modelling source differences as an affine transformation between the sources $(X, Y)$ and $(\tilde{X}, \tilde{Y})$. I designed algorithms based on optimization techniques to validate the model assuming that the data matrices are each other affine deformations. I have proposed and proved that the affine transformation could be computed from the average, the empirical covariance matrices and the empirical covariance vectors (subthesis II.1.). I have also proposed an algorithm based on a similar approach, where the parameters are computed by matching the moments in sources. In the second group of methods, the affine transformation is computed with the maximization of the classification accuracy (subthesis II.2.). I also proposed an approach using a hierarchical classifier to eliminate the effects of site heterogeneity.

**Definition 3.16** (Affine transformation). An affine transform is defined as:

$$X = A\tilde{X} + b \quad (3.15)$$

$A$ corresponds to a linear transformation matrix and $b$ to the translation vector. The linear transformation matrix is multiplied by each row of the data matrix, as the translation vector is added to each of them.

For easier processing, in some methods the affine transformation is represented in homogeneous coordinates with only a $(P + 1)$ dimensional matrix $T$ instead of a $P$ dimensional linear transformation matrix $A$ and translation vector $b$. 
Definition 3.17 (Affine transformation matrix in homogeneous coordinates T). Affine transformation matrix $T$ is defined as

$$T = \begin{bmatrix} A & b \\ 0 & \ldots & 0 & 1 \end{bmatrix} \quad (3.16)$$

where $T$ is $(P+1) \times (P+1)$ matrix.

In this thesis, inter-source differences are modeled with an affine transformation between the distributions:

$$(X,Y) \sim (A\tilde{X} + b, \tilde{Y}) \quad (3.17)$$

Subthesis II.1.: Projecting the Statistical Properties

I have proposed approaches based on matching the statistical properties of the data matrices in order to estimate the parameters of the affine transformation. I proved that if the relationship between the sources are defined by an affine transformation, the linear transformation matrix $A$ and the translation vector $b$ are computed from the average, empirical covariance matrix and empirical covariance vector. In addition, the statistical moments could be matched to each other in order to estimate the parameters of the affine transformation.

Definition 3.18 (Average of the labels $Y$). Define $m_Y$ as the empirical average of $Y$:

$$m_Y = \frac{1}{n} \sum_{i=1}^{n} y_i \quad (3.18)$$

where $n$ refers to the number of variables.

Definition 3.19 (Empirical covariance vector). Define $c$ as a covariance vector calculated from the data matrix $X$ and label vector $Y$:

$$c = \frac{1}{n} \sum_{k=1}^{n} \left( x_k - \bar{m} \right) \left( y_k - m_Y \right) \quad (3.19)$$

where $n$ is the number of variables.

This method works with selected statistical properties and its main steps are illustrated on Figure 3).

$A$ is estimated by solving a least square optimization problem with quadratic restriction:

$$\min_{A} \| A\tilde{c} - c \|$$

$$s.t. \quad C = A\tilde{C}A^\top \quad (3.20)$$

And the translation vector $b$ is determined as

$$b = \tilde{m} - Am \quad (3.21)$$

Similarly, the algorithm can be applied to the multi-source problem.

In the second approach following the previous approach, the affine transformation is estimated from geometric operations of the statistical moments assuming that the statistical moments are similar in $X$ and $\tilde{X}$.

There are only a few points needed in order to exactly compute the parameters of the affine transformation according to the number of dimensions.
Figure 3: **Cross-source Classification Analysis Flowchart with Projecting Statistical Properties**. In the first step, the algorithm computes the expected value, covariance vector and covariance matrix for the training and the testing data separately. In the next step, the linear transformation matrix $\mathbf{A}$ is calculated as a least square optimization problem with a quadratic constraint regarding the covariance vectors and matrices. The translation vector $\mathbf{b}$ is computed from difference between the expected values of the training and testing data. Finally, a classifier is trained using transformed training data and the same classifier was used on every training dataset.

**Algorithm 3** Cross-classification with the empirical covariance matrix, empirical covariance vector and average

1. Input: $(\mathbf{X}, \mathbf{Y})$ and $(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$ data matrices with corresponding labels
2. Compute average $(\bar{\mathbf{m}}, \tilde{\mathbf{m}})$, empirical covariance matrix $(\mathbf{C}, \tilde{\mathbf{C}})$ and empirical covariance vector $(\tilde{\mathbf{c}}, \bar{\mathbf{c}})$ of $(\mathbf{X}, \mathbf{Y})$ and $(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}})$.
3. Compute $\mathbf{A}$ by minimizing $\min_{\mathbf{A}} \| \mathbf{A} \tilde{\mathbf{c}} - \mathbf{c} \|$ subject to $\mathbf{C} = \mathbf{A} \tilde{\mathbf{C}} \mathbf{A}^\top$
4. Compute $\mathbf{b} = \tilde{\mathbf{m}} - \mathbf{A} \mathbf{m}$
5. Train classifier on $\mathbf{X}$
6. Test on $\mathcal{F}(\tilde{\mathbf{C}}) = \mathbf{A} \tilde{\mathbf{X}} + \mathbf{b}$

**Definition 3.20** (Degree of freedom). The $df$ degree of freedom denotes the independent variables that are free to vary in the aforementioned affine transform

$$df = P^2 + P$$

if $P$ is the dimensionality of $\mathbf{X}$.

The $k$th moments of the data are selected as independent variables in order to estimate the affine transformation matrix.

**Definition 3.21** (Moment of order $k$).

$$m_k = \frac{\left( x_1^k + x_2^k + \cdots + x_n^k \right)}{n}$$

where $n$ refers to the number of variables. Exponentiation of a vector $\mathbf{x}$ refers to raising each of the coordinates to the power of $k$. 

12
**Definition 3.22** (Moment matrix of data matrix $X$ with the order of $k$).

$$M_k(X) = [m_1, m_2, \ldots, m_k]$$ \hspace{1cm} (3.24)

where $m_k$ are the $k$th moment vectors of $X$.

The approach is adopted to the data sets as follows in algorithm 4 for the cross-source classification and a similar approach can be added to the multi-source problem:

**Algorithm 4** Cross-classification with Projecting the moments

1: Input: $(X, Y)$ and $(\tilde{X}, \tilde{Y})$ data matrices with corresponding labels
2: Compute moment matrix $M_k(X)$ containing the $k$th order moments for $X$, where $k$ is close to $P + 1$
3: Solve the least square problem $\min_T \| (M_k(T\tilde{X}), 1) - (M_k(X), 1) \|$
4: Train classifier on $X$
5: Test on $T\tilde{X}$

**Subthesis II.2.: Optimization of classification performance**

I have proposed optimization methods with the aim to maximize the classification accuracy in cross and multi source problems. Here, the affine deformation matrix is estimated through the optimization procedure. Experimental results show that the approach significantly increases the classification rate. I also proposed a hierarchical classification based method derived from the maximization of expected accuracy, where a classifier on each dataset is used to project the data into a new common space.

**Definition 3.23** (Expected Accuracy). Define $\pi$ as the expectation of accuracy given random trials of the classification problem on samples from $X$ using classifier $\sigma(X)$.

$$\pi(\sigma(X)) = E\{ac(\sigma(X))\}$$ \hspace{1cm} (3.25)

where $ac(X)$ refers to the accuracy after one trial.

**Definition 3.24** (Optimal Affine Transformation in Cross-source Classification with Expected Accuracy). Assume the optimal projection is estimated between the sources, where the classifier is trained on $(X, Y)$ and tested on $T\tilde{X}\tilde{X}$ from source 2:

$$\max_{T\tilde{X}, \sigma} E\{ac(\sigma_X(T\tilde{X}))\}$$ \hspace{1cm} (3.26)

Experimental results show that the classification performance improved significantly and is suitable to solve the multi-site imaging problem.

**Definition 3.25** (Projection with a Classifier). Define $\xi(X)$ as a projection of the training data $X$ to a space, where the classification is optimal, this projection is dependent from the classification algorithm.

Substitute $\sigma_X$ classifier function with projection $\xi(X)$.

The projection is naturally the outcome of the classifier algorithm. By using SVM the data is projected to the SVM hyperplane space, each of the data points are converted into their distance from the hyperplane, these are the SVM scores. Then for the actual classification this distance score can be applied as an input from all datasets for the higher level classifier.
Definition 3.26 (Hierarchical Classifier in Cross Source Classification). With a classification algorithm $X$ is projected to $\xi(X)$, similarly $\tilde{X}$ is projected to $\tilde{\xi}(\tilde{X})$. Then a higher level classifier is trained on $\xi(X)$ and tested on $\tilde{\xi}(\tilde{X})$:

$$\max_{\sigma, \xi, \tilde{\xi}} \sigma \xi(X)(\tilde{\xi}(\tilde{X}))$$  \hspace{1cm} (3.27)

Experimental results show that the classification performance improved significantly and is suitable to solve the multi-site imaging problem.

Thesis III.: Game-User Interaction, Physiological markers and Learning

Publications related to this thesis are [2][3][7][8][9][10][11].

My third thesis also focuses on analysing cognitive dysfunctions with biomarkers using partly the algorithms proposed in the previous theses but with a different perspective and with physiological sensors. I present a self-rewarding, biofeedback framework with the aim of measuring and modeling learning in a gaming environment. I propose a hybrid Game-User Interaction model based on research in cognitive psychology and on user studies in order to investigate the learning experience mainly in educational games. Lastly, a software package is presented on Android OS as well as the applications of the results focusing on children with learning difficulties.

First, I model dynamic difficulty adjustment with a general architecture based on reinforcement learning (subthesis III.1.). I validated the difficulty adjustment with a user experience study and represented game-user interaction with a hybrid model based on control and game theory. I defined mental states and modeled them with a game tree and validated them with a user experience study (subthesis III.2.). Lastly, I proposed markers to track learning and measured them with widely acknowledged tests in cognitive psychology. I also validated these markers with a user study (subthesis III.3.).

Subthesis III.1.: Representation of Game User Interaction: a reinforcement learning model

I proposed a general architecture based on reinforcement learning to model game-user interaction and dynamic difficulty adjustment and I offered definitions to model its parameters based on Flow theory, a cognitive psychological approach.

Definition 3.27 (Difficulty). The difficulty $d$ denotes the difficulty level in the game session. This is a complex measure and it is purely defined by the game.

Definition 3.28 (Result). The result $r$ represents the actual performance of the player. This is a computed value and is directly measured in the game from the user interactions.

Definition 3.29 (Mental State). Mental states $\Lambda$ are discrete projections of the mental effort presented in the second thesis. Four different mental states are differentiated between in this model: fatigue, frustration, boredom or engagement.

The framework intends not only to create the state of Flow by providing a more engaging experience but to maximize the performance of the educational gameplay.
Subthesis III.2.: User performance and mental state representation

I proposed a hybrid model based on a general state-space model and a game tree to model learning in computer games. The performance over game sessions is represented by differential equation. I validated the model with user studies.

The player has a preliminary performance result denoted by $r_0$, which indicates an initial state, it represents the prior level of the player’s performance. To control the game and the player, the framework calculates the difficulty $d$ of the next level and sends it to the game. As a result, the player solves the session by increasing the performance $r$. During the game session, the player maintains a certain mental state denoted by $\Lambda$ (fatigue, frustration, boredom or engagement), and this is estimated from the physiological signals. The actual performance of the player is defined by the result, and the change in this value represents learning. The performance is modeled by a state space model (Eq. 3.28), where it depends on the preliminary performance and the latest difficulty in the game. This particular model representation was created using cognitive psychology principles.

$$\dot{r} = -\alpha r + \beta d$$  \hspace{1cm} (3.28)

On the other hand, mental state can be estimated by monitoring the history of the game. If the difficulty does not change but the players performance drops, it indicates that the player is getting into the "fatigued" mental state. The upcoming mental state $\Lambda$ is represented by a game tree, where one of the opponents is the player and the other is the game itself (figure 5). In one level the game adaptively changes the difficulty, and the player provides a result in accordance. A gameplay consists of multiple levels, so there are multiple cycles of these two steps.

In a user study, I demonstrated that adapting the difficulty in a well-known game can actually lead to a change in enjoyment, scores and mental states. I also selected physiological markers in order to estimate the mental effort and classify the mental states.
Figure 5: **Mental States.** The possible mental states are modeled as a result of an interaction between the game and the player. This interaction can be represented by a game tree.

**Subthesis III.3.: User performance markers**

*I proposed psychological markers to estimate user performance in an educational gaming environment. These markers are based on cognitive psychological research advances and they are measured accordingly in a separate user study.*

**Definition 3.30 (Ability).** *The ability \( \beta \) denotes the cognitive ability of the player to increase their performance through the game.*

**Definition 3.31 (Attitude).** *The attitude \( \alpha \) represents the attitude towards the game.*

To obtain the ability a group of well-defined cognitive psychological tests were used (non-verbal test of intelligence, working memory tests, etc.). For attitude Gardner’s Multiple Intelligences were used, estimated with a questionnaire created by Gardner, Armstrong and extended by Eva Gyarmathy.

I designed a user study to validate these markers and the experimental results suggest strong correlation between the performance in game and the defined measures.

### 4 Application of the new results

The first two group of methods (thesis I and II) proposed in my dissertation have been evaluated in an environment consisting of mainly MatLAB scripts. I created a general framework for the evaluation of the various aspects.

The proposed algorithms and methods can be applied to different multi-source problems like EEG studies to eliminate inter-subject differences.

The third thesis was implemented as a software package on Android system and in addition the user studies were implemented on Windows to provide a highly accurate laboratory environment for the physiological measures. The Android software package was designed with these demands in mind and has four roles. The first role is the supervisor unit, it enables the teachers (supervisors) to observe and in special cases control user
activity. The second component is the server, where all data is stored. With them supervisors are able to track user profiles (e.g. to detect any changes in activity). The third role is the framework. It communicates with the supervisor and sends the data to the server. It is also responsible for integrating physiological signals with educational gaming on the mobile device. The framework also classifies mental effort levels and creates markers around areas of special interest.

The supervisor and the framework were implemented on Android operating system, while the server resides in a .NET environment. These decisions were made based on platform popularity and the availability to access low-level services. It was also important to run native code on the mobile device in case of calculation-intensive algorithms. The platform selected for the server side ensures the use of best practices regarding the architecture and good scalability.

5 Scientific Publications

International Journals


Conference Proceedings


