BABEŞ-BOLYAI UNIVERSITY, CLUJ-NAPOCA, ROMANIA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES, ROUEN, FRANCE,
LABORATOIRE D 'INFORMATIQUE, DE TRAITEMENT DE L'INFORMATION ET DES SYSTÈMES

Dynamic Machine Learning For Supervised and Unsupervised Classification

PhD Thesis Abstract

PhD Student:
Adela-Maria SÎRBU (married RUS)

PhD Directors:
Professor Dr. **Gabriela CZIBULA** – *Babeş-Bolyai University, Romania*
Professor Dr. **Abdelaziz BENSRHAIR** – *INSA de Rouen, France*

PhD Advisor:
Assoc. Professor Dr. **Alexandrina ROGOZAN** – *INSA de Rouen, France*

June 2016

Aknowledgements

First of all, I would like to express my deepest gratitude to my two PhD directors: Prof. Dr. Gabriela Czibula and Prof. Dr. Abdelaziz Bensrhair, for their excellent guidance, patience, scientific and financial support during my PhD studies. Without their help I would have not been able to complete this work.

I would like to give my special thanks to my PhD a Assoc. Prof. Dr. Alexandrina Rogozan who has also guided me through my research, encouraged me and who made possible the collaboration with INSA.

Second, I would like express my gratitude to the jury that accepted to review my thesis: Prof. Dr. Horia F. Pop (Babeş-Bolyai University), Prof. Dr. Daniela Zaharie (West University of Timisoara) and Prof. Dr. Fawzi Nashashibi (INRIA, France).

Being part of two universities, I had the opportunity to meet wonderful persons who helped me in a professional or personal way. I am very grateful to Assoc. Prof. Dr. Laura Dioşan for her continuous guidance and for being always there for me, since my second year of university. I also want to thank my colleagues from Babeş-Bolyai University: Iuliana, Zsuszanna and Gabriel and from INSA: Alina, Vannee, Bassem, Fabian and Rawia for their help and friendship. It was great to feel part of such a team.

Last, but not least I would like to thank my husband Florin, my families (Sîrbu and Rus) and my friends for believing in me, for their long patience and their unconditional support during these difficult years.

Contents

List of publications

Publications in ISI Web of Knowledge

Publications in ISI Science Citation Index Expanded

1. Gabriela Czibula, Istvan-Gergely Czibula, **Adela Sîrbu** and Gabriel Mircea A novel approach to adaptive relational association rule mining. *Applied Soft Computing* published by *Elsevier*, volume 36, pp. 519–533, 2015

2. Bassem Besbes, Alexandrina Rogozan, **Adela-Maria Rus (Sîrbu)**, Abdelaziz Bensrhair and Alberto Broggi. Pedestrian Detection in Far-Infrared Daytime Images Using a Hierarchical Codebook of SURF. *Sensors*, volume 15, no. 4, pp. 8570-8594, 2015, doi:10.3390/s150408570

Publications in ISI Conference Proceedings Citation Index

3. **Adela-Maria Sîrbu**, Gabriela Czibula and Maria-Iuliana Bocicor. Dynamic Clustering of Gene Expression Data Using a Fuzzy Approach. *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, SYNASC 2014, Timisoara, Romania, September 22-25, pp. 220-227, 2014

4. **Adela-Maria Sîrbu**, Alexandrina Rogozan, Laura Dioşan and Abdelaziz Bensrhair. Pedestrian Recognition by Using a Kernel-Based Multi-modality Approach. *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, SYNASC 2014, Timisoara, Romania, September 22-25, pp. 258-263, 2014

Papers published in international journals and proceedings of international conferences

5. **Adela Sîrbu** and Maria-Iuliana Bocicor. A dynamic approach for hierarchical clustering of gene expression data. *Proceedings of IEEE International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, Romania, pp. 3-6,2013.

6. **Adela-Maria Rus (Sîrbu)**, Alexandrina Rogozan, Laura Dioşan and Abdelaziz Bensrhair. Pedestrian recognition using a dynamic modality fusion approach. *Proceedings of IEEE International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, Romania, September 3-5, pp. 393 - 400, 2015

7. **Adela-Maria Rus (Sîrbu)**, Alexandrina Rogozan, Laura Dioşan and Abdelaziz Bensrhair. Pedestrian recognition by using a dynamic modality selection approach. *Proceedings of IEEE 18th International Conference on Intelligent Transportation Systems*, Las Palmas, Spain, September 15-18, pp. 1862 - 1867, 2015

8. Maria-Iuliana Bocicor, **Adela-Maria Sîrbu** and Gabriela Czibula. Dynamic core based clustering of gene expression data. *International Journal of Innovative Computing, Information and Control*, volume 10, no. 3, pp. 1051-1069, 2014

9. **Adela Sîrbu**. A study on dynamic clustering of gene expression data. *Studia Universitatis "Babes-Bolyai", Informatica*, Romania, Vol. LIX, Number 1, pp. 16-27, 2014.

10. Anca Andreica, Laura Dioşan, Radu Găceanu and **Adela Sîrbu**. Pedestrian recognition by using kernel descriptors. *Studia Universitatis "Babes-Bolyai", Informatica*, Romania, Vol. LVII, Number 2, pp. 77-89, 2013.

Introduction

This PhD thesis is the result of my research in the field of applying dynamic machine learning models for solving supervised and unsupervised classification problems. This research was started in 2012, under the supervision of my PhD directors: Prof. Dr. Gabriela Czibula and Prof. Dr. Abdelaziz Bensrhair and my PhD advisor Assoc. Prof. Alexandrina Rogozan.

Machine Learning (ML) is a subfield of artificial intelligence, focused on constructing computer programs that automatically improve with experience [43]. The research direction we are focusing on in the thesis is applying dynamic machine learning models to solve *supervised* and *unsupervised* classification problems. The particular problems that we have decided to approach in the thesis are pedestrian recognition (a supervised classification problem) and clustering of gene expression data (an unsupervised classification problem). The approached problems are representative for the two main types of classification and are very challenging, having a great importance in real life.

The problem that we address within the dynamic unsupervised classification research direction is the dynamic clustering of gene expression data. Modern microarray technology is nowadays used to experimentally detect the levels of expressions of thousand of genes, across different conditions and over time. Once the gene expression data has been gathered, the next step is to analyze it and extract useful biological information, which can be achieved through clustering. In the case of gene expression data sets, each gene is represented by its expression values (features), at distinct points in time, under the monitored conditions.

In our proposed models, the term *dynamic* indicates that the data set is not static, but it is subject to change. Still, as opposed to the incremental approaches from the literature, where the data set is enriched with new genes (instances) during the clustering process, our approaches tackle the cases when new features (expression levels for new points in time) are added to the genes already existing in the data set. To our best knowledge, there are no approaches in the literature that deal with the problem of dynamic clustering of gene expression data, defined as above.

The problem that we approach within the dynamic supervised classification field is the development of dynamic pedestrian recognition systems. Pedestrian safety is a critical issue with global impact, as traffic accidents are one of the major causes of death and injuries around the world. In order to protect the vulnerable road users (pedestrians, cyclists) advanced driver assistance systems (ADAS) were developed. They assist the driver in making decisions, provide signals in potentially dangerous driving situations and execute counteractive measures. However, in order for the ADAS to work, it is highly necessary to develop efficient recognition systems.

In order to develop dynamic pedestrian recognition systems that are able to adapt to varying environmental conditions, we integrate information from multi-modality images such as intensity, depth and flow into dynamic models. It is known that the first need of a pedestrian recognition system is finding a robust feature set extracted from images, that allows cleanly discriminate the human form, even under difficult conditions. Taking this into account, we also aim to study a new technique for extracting features from images, by using kernel descriptors (KDs) [12], which obtained good results for visual recognition, but to our best knowledge, have not been used in pedestrian detection so far.

The thesis is organized in four chapters as follows.

In **Chapter 1** we present the background for the unsupervised classification problem approached in the thesis, the problem of dynamic clustering of gene expression data. We begin with an overview on the most important clustering methods, with focus on *k-means*, *fuzzy c-means* and *hierarchical clustering* algorithms, that we are going to further exploit in our proposed approaches. We continue with a special type of clustering, the dynamic clustering, along with a literature review in this direction. Finally, we address the problem of dynamic clustering of gene expression data, together with a short review of the existing approaches from the literature.

Chapter 2 is original and presents our work related to the problem of dynamic clustering of gene expression data. We begin by defining the problem that we approach and its importance in real life, then we introduce three dynamic clustering algorithms that can handle new collected gene expression levels, by starting from a previous obtained partition, without the need to re-run the algorithm from scratch. In the same context of dynamic data, we also propose an algorithm for adaptive relational association rule mining of gene expression. We experimentally evaluate our approaches on a gene expression dataset, analyze and compare our results with the ones obtained by other approaches from the literature. The obtained results emphasize the effectiveness of using our dynamic models.

In **Chapter 3** we present the background for the supervised classification problem approached in the thesis, the problem of dynamic pedestrian recognition. We begin by presenting the most important components in a pedestrian recognition system: the feature extraction and the classification components. Therefore, we give an overview of the most commonly used features, then we briefly present the most popular classifiers used in pedestrian detection. Finally, we provide a short literature review in the field, with emphasis on the fusion of image modalities, for which we are going to further introduce a dynamic approach.

Chapter 4 is original and addresses the problem of pedestrian recognition in single and multi-modality images. We begin with a comparison on several state-of-the-art features in far infra-red (FIR) spectrum, we continue with a literature review on kernel descriptors, then we present our studies on pedestrian recognition using these features for image representation.

In the first study we investigate how two learning algorithms, Support Vector Machines (SVM) and Genetic Programming (GP), are able to perform pedestrian recognition using kernel descriptors, extracted with three types of kernels: Exponential, Gaussian and Laplacian, while in the second one we study how kernel descriptors perform in single vs. multi-modality pedestrian recognition. We propose two dynamic models for pedestrian recognition, that are able to select the most discriminative modalities for each image in particular and further use them the classification process. Experimental evaluations on a pedestrian dataset confirm the performance of our dynamic models.

The original contributions from the thesis are presented in Chapters 2 and 4 and are the following:

- Three dynamic clustering algorithms, which can handle newly collected gene expression levels by starting from a previously obtained partition, without the need to re-run the algorithms from the beginning

  - A dynamic core based clustering algorithm, based on *k-means* clustering algorithm (Subsection 2.2.3.1) [14]; an heuristic to determine the optimal number of clusters in a gene expression data set (Subsection 2.2.2) [14], experimental evaluations of the dynamic core based clustering algorithm on a real life gene expression data set, analysis of the results and comparisons with results obtained by other dynamic approaches in the literature (Subsections 2.3.3 and 2.4.2) [14].

  - A dynamic algorithm for hierarchical clustering of gene expression data, based on *hierarchical agglomerative* clustering algorithm (Subsection 2.2.3.2) [59]; experimental evaluations of the algorithm for hierarchical clustering on a real life gene expression data set, analysis of the results and comparisons with results obtained by our other dynamic approaches (Subsections 2.3.3 and 2.4.2) [59].

  - A dynamic algorithm for fuzzy clustering of gene expression data, based on *fuzzy c-means* clustering algorithm (Subsection 2.2.3.3) [60]; experimental evaluations of the algorithm for fuzzy clustering on a real life gene expression data set, analysis of the results and comparisons with results obtained by our other dynamic approaches (Subsections 2.3.3 and 2.4.2) [60]

- A dynamic algorithm for relational association rule mining of gene expression data, which can handle the newly arrived features by adapting previously obtained rules, without the need of re-running the mining algorithm from scratch (Subsection 2.4.3) [18]; experimental evaluations of the algorithm for relational association rule mining on a real life gene expression data set and analysis of the results (Subsection 2.4.3.1) [18]

- The usage of kernel descriptors for pedestrian recognition (Subsections 4.2.1 and 4.2.2) [3, 61]

  - A comparison on how two machine learning algorithms: SVM and GP are able to learn based on KD features extracted by using three kernels: Exponential, Gaussian and Laplacian (Subsection 4.2.1) [3]

  - A comparison on how KDs perform on single vs. multi-modality images, with parameters optimized independently on each modality: intensity, depth and flow (Subsection 4.2.2) [61]

- Two dynamic machine learning based algorithms, capable to dynamically determine the most suitable modalities to classify an image

  - A dynamic modality selection algorithm which retains one suitable modality among intensity, depth and flow (Subsection 4.3.3) [52]; experimental evaluations of the algorithm on a pedestrian data set, analysis of the results and comparisons to other approaches from the literature (Subsection 4.3.3.1) [52]

  - A dynamic modality fusion algorithm which fuses the modalities considered suitable among intensity, depth and flow (Subsection 4.3.3) [62]; experimental evaluations of the algorithm on a pedestrian data set, analysis of the results and comparisons to other approaches from the literature (Subsection 4.3.4.1) [62]

Chapter 1

Unsupervised classification. Background.

In this chapter we are presenting the background knowledge related to the unsupervised classification problem approached in the thesis, the problem of *dynamic clustering of gene expression data.* Thus, in Section 1.1 we give a short overview on clustering, with emphasis on *k-means*, *fuzzy c-means* and *hierarchical clustering* algorithms, which stand at the basis of our proposed approaches introduced in Chapter 2. In Section 1.2 we present a special type of clustering, the dynamic clustering, together with some existing dynamic approaches from the literature. The problem of dynamic clustering of gene expression data is presented in Section 1.3 followed by several dynamic approaches existing in the literature for clustering gene expression.

1.1 Clustering

Clustering implies partitioning a particular data set in groups, whose components are similar to each other [35]. According to Jain and Dubes [34], clustering is a type of unsupervised classification applied on a finite set of instances (objects), between which the relationship is represented by a proximity matrix. Kendal and Stuart [36] suggest the term of *clustering* for techniques that group variables, and *classification* for techniques that group individuals. However, *clustering* is considered to be the most important *unsupervised learning* problem in the machine learning domain.

1.1.1 K-means clustering

The *k-means* algorithm takes as input parameter the number of clusters k and performs a partitioning of a set of n objects into k clusters, with the goal of achieving a high intracluster similarity and a low intercluster similarity. The similarity between clusters is computed against the mean value of the objects in the cluster, referred as cluster centroid [31].

The algorithm begins by randomly selecting k objects, as cluster centroids. Each of the remaining objects are assigned to the nearest cluster considering the distance between the object and the centroid (mean) of the cluster, then a new mean for each cluster is computed. The process repeats until convergence.

1.1.2 Hierarchical clustering

The hierarchical clustering method performs a grouping of objects into a tree of clusters. Depending on how hierarchical decomposition is done, bottom-up by merging or top-down by splitting, hierarchical clustering algorithms can be divided into *agglomerative* or *divisive.*

The agglomerative clustering method, starts with a partition in which each object is placed in its own cluster. Then, the pair of closest clusters is merged into a single cluster, creating a partition and decreasing by one the number of clusters. This step is repeated until all objects belong to a single cluster. The divisive hierarchical clustering method performs the same steps, but in reversed order: starts with a partition where all objects are in a single cluster, and ends when all objects are placed in their own cluster.

In order to decide which clusters to merge, the distance between clusters is computed. There are several measures used in the literature for the distance between two clusters: minimum distance, maximum distance, mean distance and average distance.

1.1.3 Fuzzy C-means clustering

Fuzzy c-means clustering (FCM) [1], [34], also known as Fuzzy ISODATA, is a clustering method which is different from hard k-means clustering. *FCM* uses the idea of fuzzy partitioning, where a data instance (object) can belong to all clusters with different membership degrees (varying between 0 and 1).

FCM uses fuzzy sets in the clustering process, associating to each object a degree of membership to each cluster.

Denoting by k the desired number of clusters, a matrix U is used, where U_{ij} $(i \in \{1,,2...,k\}, j \in \{1,2...,n\}$ expresses the membership degree of object j to cluster i, such that

$$\sum_{i=1}^{k} U_{ij} = 1, \forall j \in \{1,2,...,n\} \tag{1.1}$$

Through an iterative process, FCM updates the cluster centroids and the membership degrees, in order to move the cluster means to the correct place within the data set. The convergence of *FCM* to the optimal solution is not ensured, because of the random initialization of the initial centroids (the initial values for matrix U). The algorithm reports the final values for the matrix U. Considering the final membership degrees given by the matrix U, an object O_j is usually assigned to the cluster $i = argmax_{l=1,k} U_{ij}$. Then, the following formula is used:

$$K_i = \{j \mid j \in \{1,...,n\}, U_{ij} > U_{rj}, \forall r \in \{1,...,n\}, r \neq j\},$$

in order to obtain the clusters in data after applying *FCM*. The number of clusters formed by *FCM* is less or equal to k, because empty clusters could be obtained.

1.2 Dynamic clustering

A clustering algorithm can be regarded as *dynamic* from several perspectives: operates on dynamic data sets, or/and adapts in some way the clustering process (e.g. adaptation from external feedback, dynamic thresholds). In the following we are going to briefly review them, together with some existing approaches from the literature, with focus on the first type of dynamism, which will be tackled in this thesis.

Firstly, a clustering algorithm is considered *dynamic* if is able to handle data sets that are subject to change. The term *dynamic* represents a generalization of the terms *adaptive*, *incremental* and *temporal* used in the literature to describe this particular type of clustering. Moreover, the dynamism of the data sets can be regarded from two perspectives: when new objects (instances) are added to the data set, or when new attributes are added to the existing objects. There are several works in the literature which approach the problem of dynamic clustering when new instances are sequentially added to the data set.

In [72] is proposed an adaptive clustering algorithm based on DBSCAN, which is able to to reduce the number of distance calculations by using the triangle inequality. Their method first stores the distances between a representative object and objects in n-dimensional space, then further use them to avoid distance calculations in (n+m)-dimensional space. Experimental evaluations confirm the efficiency of their approach.

In [39] is proposed an adaptive clustering algorithm for network clustering (SACA), which adaptively forms clusters based on an accurate clustering measure called SCM, taking into account the connectivity of the nodes. In order to join or leave their clusters, the nodes must fulfill the condition of improving the SCM value of the whole network. Experiments prove the efficiency and accuracy of the method, even for large topologies.

In [38] is introduced a batch dynamic incremental c-means clustering algorithm based on rough fuzzy set, BD-RFCM, in which new available data is not added one by one to the data set, but in form of cluster. The affiliation between a cluster from new incremental data and a cluster from original data is measured by the inclusion degree, which is similar to the membership degree of fuzzy set. Experiments on a synthetic data set confirm the effectiveness and correctness of the algorithm.

Most of the existing clustering methods, such as the *k-means* algorithm [33] or *hierarchical agglomerative clustering* algorithm (HACA) [31], start with a known set of objects, characterized by a set of attributes (features). All attributes are simultaneously used when computing objects' similarity. However, there are various applications where the attribute set characterizing the objects evolves, thus, re-clustering is required. An option in this situation would be to apply the clustering algorithm again from the beginning on the feature-extended objects, but would not be efficient. In order to overcome this problem, Şerban and Câmpan introduce in [55] and [56] two adaptive clustering algorithms that are able to identify a new partition of the set of objects, when the features set increases. The methods start from the set of clusters that was obtained by applying *k-means*, respectively *HACA* before the feature set was extended. This partitioning is adapted considering the newly added features. The authors show that the result is reached more efficiently than running *k-means*, respectively *HACA* from the beginning on the extended data set.

Secondly, a clustering algorithm is considered dynamic if performs an adaptation of the clustering. In [64] is presented a fast incremental clustering algorithm by dynamically changing the radius threshold value, in which the original data set is scanned only once according to the memory capacity. Experimental evaluations performed on mushroom dataset demonstrate the effectiveness of the method.

In [6] is introduced an adaptive clustering model which uses external feedback to improve cluster quality. Execution time is speeded up by using past experience in an adaptive environment, in which the reward values of successive clusterings are learned through Q-learning. Experimental evaluations show that the use of adaptive clustering brings important improvements of instance-based learning methods, like k-nearest neighbor classifiers.

1.3 Clustering of gene expression data

Nowadays, microarray technology and the more modern next-generation sequencing technology allow the collection of huge amounts of genetic data in the form of gene expression. These new technologies offer the possibility to measure the expression levels, or the activity of thousands of genes belonging to cells from different organisms, as the cells are exposed to different environmental conditions. The amount, time or location of expression of a certain gene are important characteristics to be determined, as they have a great impact on the well functioning of cells or of an organism, on a broader scale. One of the most popular procedures used to analyse the gene expression data is clustering [63].

A great number of algorithms have been proposed for clustering gene expression data sets that are not subject to change. Among these, we mention approaches based on the k-means [7] or the fuzzy k-means algorithms [5], on artificial neural networks [75] or methods using self organizing maps in conjunction with hierarchical clustering [32], with k-means clustering [74] or with particle swarm optimisation [73].

Concerning clustering of dynamic gene expression data sets, to our knowledge, there are no approaches in the literature that deal with the dynamic clustering problem in the context when new expression levels are added to the existing genes. Although, as mentioned, no techniques exist that cluster gene expression data containing instances with an increasing number of features, there are a series of incremental clustering methods that are designed to work for data sets in which the number of instances may increase over time. We will present in the following some of these incremental approaches, as well as other studies that use dynamic or incremental clustering methods.

Sarmah and Bhattacharyya [53] present a density based approach technique for clustering gene expression data, which can also be applied for incremental data. Their algorithm, GenClus [53], obtains a hierarchical cluster solution and has as an advantage the fact that it does not require the number of clusters as input. InGenClus, the incremental version of GenClus, uses the result offered by GenClus and is able to update it when new genes are added to the data set, therefore decreasing the computational time. Both algorithms are evaluated using real-life data sets and the reported results prove a good performance.

In [22] the authors propose an incremental clustering algorithm for gene expression data - incDGC, based on a clustering algorithm they had previously introduced. The main idea is that when a new gene is introduced into the data set, the current clustering should only be affected in the neighborhood of this gene. This algorithm does not require as input the number of clusters and it helps avoiding performing the clustering each time the data set is updated. The algorithm was tested on three data sets and it proved to outperform other clustering algorithms, such as k-means or hierarchical clustering.

Lu *et al.* [42] introduce an Incremental Genetic K-means Algorithm (IGKA), which computes an objective value that the authors define and clusters centroids incrementally under the condition that the mutation probability from the genetic algorithm part is small, which leads to high costs of centroid calculation.

Bar-Joseph *et al.* present in [8] a clustering algorithm for time series gene expression data that performs the clustering on continuous curve representations of the data, obtained using statistical spline estimation.

An and Doerge [2] introduce a novel dynamic clustering approach that deals with the time dependent nature of the genes so that the genes in the same cluster may have different starting and ending points or different time durations.

Chapter 2

New approaches for dynamic clustering of gene expression data

In this chapter, we address the problem of dynamic clustering of gene expression data and we propose three dynamic clustering algorithms, which can handle newly collected data, by starting from a previously obtained partition, without the need to re-run the algorithm from the beginning. In the same context of dynamic data, an algorithm for adaptive relational association rule mining of gene expression is also proposed [18]. The models introduced in this chapter are original, and were introduced in [14, 58, 59, 60, 18].

2.1 Problem statement and relevance

The emergence of microarray technology, that allows measuring the levels of expression of thousands of genes has led to an exponential increase in the amount of gene expression data. Still, all this data would be useless unless relevant biological information was extracted from it, therefore thorough exploratory analysis are usually required and performed. One of the most widely used techniques for this analysis and, most frequently, the first step, is clustering.

In what concerns gene clustering, the goal is twofold: firstly, by dividing the huge amount of gene expression data into clusters, this data becomes easier to process and analyse; secondly, but not less important, it is assumed that genes having similar expression patterns over time (during the experiments) are likely to have similar biological functions and therefore clustering can also be considered an initial stage in the process of determining gene functions.

Gene expression data is usually collected with the goal of investigating the progress of different biological processes, as they evolve under different conditions and over time. Since biological processes are dynamic and time varying, they are best described by time series gene expression data [2]. A time series data set consists of a collection of data obtained from a particular type of biological experiment: samples of cells or tissues are extracted from the same individual at different time points, during the evolution of the biological process. Thus, for each of the targeted genes, the level of expression is measured at several distinct points in time. The data set will then be composed of thousands of genes (instances), each gene being characterized by a set of attributes (features): its expression levels (which can be quantified as real numbers) at different points in time.

However, there are some processes worth studying that may take days, or even months (e.g. diseases that progress over time), as well as experiments that are conducted over longer durations of time. For such cases, researchers must either wait until the experiment finishes and the expression levels of the genes are available at all time points, or analyse the data gradually, as the experiment progresses. Clustering of gene expression data might be performed during the evolution of the experiment, at intermediate time steps, when genes would be characterized by only a subset of the entire set of features (time points). The main disadvantage is that as the experiments advance and new expression levels of the targeted genes for new points in time are available, the clustering process must once again start from scratch, which requires considerable time (especially as the number of the genes in such data sets is extremely high), in the end leading to a slower and more inefficient processing of the data.

To overcome this drawback, we propose a *dynamic clustering algorithm*, based on the idea of incremental clustering introduced in [55]. Given a previously obtained partition of a data set and new features for all the genes in this data set, the dynamic clustering algorithm is able to re-cluster the set of instances, without the need to start from the beginning, but by using the existing partition. This way, the clustering of genes at intermediate time points during the experiment can be more efficiently exploited and the final result could be achieved in smaller amounts of time.

2.2 Methodology

2.2.1 Theoretical considerations

In the following we introduce our approaches for dynamic clustering of gene expression data. The starting point of our proposals are the incremental clustering ideas previously introduced in [55, 56] that are extended in the following to handle the problem of dynamic clustering of gene expression data.

We first introduce some theoretical considerations, common for all methods that we propose.

Let $X = \{G_1, G_2, \ldots, G_n\}$ be the set of genes to be classified. Each gene is measured m times and is therefore described by an m-dimensional vector $G_i = (G_{i1}, \ldots, G_{im}), G_{ik} \in \Re, 1 \leq i \leq n, 1 \leq k \leq m$. An element G_{ik} from the vector characterizing the gene G_i represents the expression level of gene G_i at time point k.

Let $\{K_1, K_2, \ldots, K_p\}$ be the set of clusters identified in X by applying the *k-means*, *HACA* or *fuzzy c-means* algorithms. Each cluster is a set of genes, $K_j = \{G_1^j, G_2^j, \ldots, G_{n_j}^j\}$, $1 \leq j \leq p$. The mean (centroid) of the cluster K_j is denoted by f_j, where $f_j = \left(\frac{\sum_{k=1}^{n_j} G_{k1}^j}{n_j}, \ldots, \frac{\sum_{k=1}^{n_j} G_{km}^j}{n_j} \right)$.

Two of the most used similarity measures or distances for gene expression data are the *Euclidean distance* and the *Pearson correlation* [37]. The measure we use for discriminating genes is the *Euclidian distance*: $d(G_i, G_j) = d_E(G_i, G_j) = \sqrt{\sum_{l=1}^{m} (G_{il} - G_{jl})^2}$. We have chosen this type of distance for the present study because it takes into account the magnitude of the changes in gene expression [37], therefore preserving more data.

The set of attributes consisting of the m expression levels of the genes coming from m consequent measurements is afterwards extended with s ($s \geq 1$) new attributes, coming from new measurements, numbered as $(m+1), \ldots, (m+s)$. The genes' extended vectors are $G_i' = (G_{i1}, \ldots, G_{im}, G_{i,m+1}, \ldots, G_{i,m+s}), 1 \leq i \leq n$.

Our aim is to analyse how the extended genes are grouped into clusters, starting from the grouping obtained before the attribute set extension. Our goal is to achieve an increased performance in comparison with the process of partitioning from the beginning.

The starting point is that, at the end of the *k-means* or *fuzzy c-means* clustering process, all genes are closer to the mean of their cluster than to any other mean. So, for any cluster j and any gene $G_i^j \in K_j$, the inequality below holds.

$$d(G_i^j, f_j) \leq d(G_i^j, f_r), \forall j, r, \ 1 \leq j, r \leq p, \ r \neq j. \tag{2.1}$$

By $K_j', 1 \leq j \leq p$, is denoted the set composed by the same genes as K_j, after the feature set was extended, and by $f_j', 1 \leq j \leq p$, the centroid of the set K_j'. After the feature set was increased, the sets $K'_j, 1 \leq j \leq p$, will not necessarily represent clusters. The newly added features can change the genes' placement into clusters, obtained such that the similarity within each cluster to be maximized and similarity between clusters to be mininimized. Taking into account that the genes' features have equal weights (and normal data distribution) it is very likely that by adding one or few features to the genes, the old partitioning in clusters is similar to the actual one. Certainly, the clusters after extending the feature set could be obtained by applying the clustering algorithm (*k-means* or *HACA* or *fuzzy c-means*) on the set of extended genes. But, this process can be computationaly costly, that is why we focus on replacing it with one less expensive, which preserves the accuracy of the obtained results. In the following, we continue to refer the sets K'_j as clusters.

2.2.2 Identifying the number of clusters

It is well known that a problem with the *k-means* and *fuzzy c-means* algorithms is that the choice of the initial centroids may lead to the convergence of the squarred error value to a local minimum, instead of a global one. On the other hand, in *HACA*, the process of merging clusters must stop when a certain number of clusters is reached. To identify the optimal number of clusters in the gene data set, as well as the initial centroids for applying *k-means*, or *fuzzy c-means* algorithm, we introduce an heuristic that identifies the representative genes in the clusters (one in each cluster).

2.2.3 Our approaches

2.2.3.1 The Core Based Dynamic Clustering of Gene Expression (CBDCGE) Approach

The algorithm which will be described in the following adapts the idea from [55] for the particular problem of dynamic gene expression clustering.

We will use the inequality (2.2) in order to determine within each cluster $K'_j, 1 \leq j \leq p$, those genes that are very likely to remain stable in their cluster, instead of moving to another cluster as a result of the feature set extension. These objects are considered to represent the *core* of their cluster. The idea of our approach is to compute, for each cluster K_j, its core, denoted by $Core_j$.

$$G_{il}^{j} \geq \frac{\sum\limits_{k=1}^{n_j} G_{kl}^{j}}{n_j}. \tag{2.2}$$

Let us denote by $StrongCore_j = \{G_i^{j'} | G_i^{j'} \in K'_j, G_i^{j'}$ satisfies the inequality (2.2) $\forall\, l \in \{m+1, m+2, \dots, m+s\}\}$. We denote by $WeakCore_j$ the set of genes in K'_j satisfying inequality (2.2) for at least the average number of features (computed from all genes belonging to K'_j) for which (2.2) holds.

For an added feature l ($m+1 \leq l \leq m+s$), and a cluster K'_j, the gene which has the maximum value for the feature l among all genes from K'_j verifies inequality (2.2). It is not sure that cluster K'_j contains any gene which satisfies inequality (2.2) for all added features $m+1, \dots, m+s$. But if such genes exist they are closer to the centroid f'_j than to any other centroid f'_r $(1 \leq r \leq p,\ r \neq j)$ and they will form the $StrongCore_j$. In this case, $Core_j$ will be chosen as $StrongCore_j$ and will be the nucleus of cluster j in the CBDCGE algorithm. In the case when $StrongCore_j$ is the empty set, then we will consider as nucleus for cluster j other genes, the most stable ones among all genes in K'_j (the genes that satisfy inequality (2.2) for as many features as possible). These genes will form the set $WeakCore_j$.

The *cluster cores*, selected as described above, will represent the nuclei in the dynamic clustering process. If the clusters remain unchanged, then all genes from $Core_j$ will certainly remain together in the same group. This will not be the case for all core genes, but for most of them.

The first step of this approach consists of applying the *k-means* algorithm on the initial data set, the one in which each gene is characterized by m expression values, at m time points. After the dataset was increased with s new expression levels, CBDCGE algorithm is applied. The dynamic algorithm begins with computing the old clusters' cores. The cores will be considered as the initial clusters in the adaptive process. Then, the CBDCGE performs the same steps as the classical *k-means* method. We have to mention that if at a certain iteration a cluster from the partition becomes empty, it is removed from the partition, and consequently the number of clusters in the partition is decreased.

2.2.3.2 The Dynamic Hierarchical Clustering of Gene Expression (DHCGE) Approach

The *DHCGE* algorithm starts from the partition obtained by *HACA*. Further, our method is based on the idea of identifying cores [56] inside existing clusters. These cores are composed of those genes that are very likely to remain in the same cluster, after the new attributes are added to all instances of the data set.

We denote by $StrongCore_j = \{G_i^{j\prime} | G_i^{j\prime} \in K_j^{\prime}, G_i^{j\prime}$ satisfies the inequality (2.2) $\forall\, l \in \{m+1, m+2, \ldots, m+s\}\}$. We denote by $WeakCore_j$ the set of genes in $K_j^{\prime}$ satisfying inequality (2.2) for at least the average number of features. If, for cluster $K_j^{\prime}$, the set $StrongCore_j$ is not empty, then it will be considered as the core $Core_j$ for cluster $K_j^{\prime}$. Else, for $Core_j$ not to be empty, the most stable genes between all genes in $K_j^{\prime}$ will be taken as seed for cluster j, i.e. the genes from the set $WeakCore_j$.

The *DHCGE* algorithm begins by identifying the cores of the clusters obtained by applying *HACA* on the initial data set. Then, when the feature set is extended, the algorithm begins the clustering process starting from these cores and continuing as the traditional *HACA*. The advantage over applying *HACA* from scratch is that *DHCGE* does not start again from clusters composed of one gene, therefore the clustering process is accelerated. We mention that the linkage metric we used to group two genes together in the hierarchical process, we used the *average link metric*. For two clusters K_i and K_j, the distance given by the average link metric is expressed by the following equation:

$$d(K_i, K_j) = \frac{\sum_{a \in K_i} \sum_{b \in K_j} d(a,b)}{\mid K_i \mid \times \mid K_j \mid}. \tag{2.3}$$

The algorithm stops when the reached number of clusters is equal to the number of clusters formerly determined using the heuristic described in Section 2.2.2.

2.2.3.3 The Fuzzy Dynamic Clustering of Gene Expression (FDCGE) Approach

The first step of this approach consists of applying the *fuzzy c-means* algorithm on the initial data set, the one in which each gene is characterized by m expression values, at m time points. It is known that, at the end of the *fuzzy c-means* clustering, all genes are closer to the mean of their cluster that to any other mean.

We will use the inequality (2.2) in order to determine inside each cluster the genes that are likely to remain in their cluster, instead of moving into other cluster after extending the feature set. These genes form the *nucleus* of their cluster. The idea of our approach is to compute, for each cluster K_j, its nucleus, denoted by $Nucleus_j$.

Let us denote by $StrongNucleus_j = \{G_i^{j\prime} | G_i^{j\prime} \in K_j^{\prime}, G_i^{j\prime}$ satisfies the inequality (2.2) $\forall\, l \in \{m+1, m+2, \ldots, m+s\}\}$. We denote by $WeakNucleus_j$ the set of genes in $K_j^{\prime}$ satisfying inequality (2.2) for at least the average number of features. The idea behind computing, for each cluster $K_j^{\prime}$, the $StrongNucleus_j$, $WeakNucleus_j$ is the same as for the algorithm *CBDCGE*, described in Section 2.2.3.1.

The *cluster nuclei* selected as mentioned above, will be considered as starting point in the adaptive fuzzy clustering method. If the clusters remain unchanged, then all genes from $Nucleus_j$ will certainly remain together in the same group. This will not be the case for all genes in the nuclei, but for most of them.

The algorithm begins by computing the clusters' nuclei, which will be further used as initial clusters in the iterative process. Then, the FDCGE performs the same steps as the classical *fuzzy c-means* method. We have to mention that if at a certain iteration a cluster from the partition becomes empty, it is removed from the partition, and consequently the number of clusters in the partition is decreased.

2.3 Experimental evaluation

In this section we aim at experimentally evaluating our dynamic clustering algorithms on gene expression data. The data set used in our experiments, the evaluation measures, as well as the obtained results are presented in the following sections. The results are original and were introduced in [14, 58, 59, 60]

2.3.1 Gene expression dataset

For the computational experiments developed in order to evaluate the performance of our method we used a real-life data set, taken from [24]. Microarray technology was used by the authors of [24] to measure the levels of expression of 6400 genes belonging to the organism *Saccharomyces cerevisiae*, during its metabolic shift from fermentation to respiration. Gene expression levels were measured at seven time points during the diauxic shift: 0, 9, 11.5, 13.5, 15.5, 18.5 and 20.5 hours. After appying a pre-processing step, the data set is reduced to a total number of 614 genes.

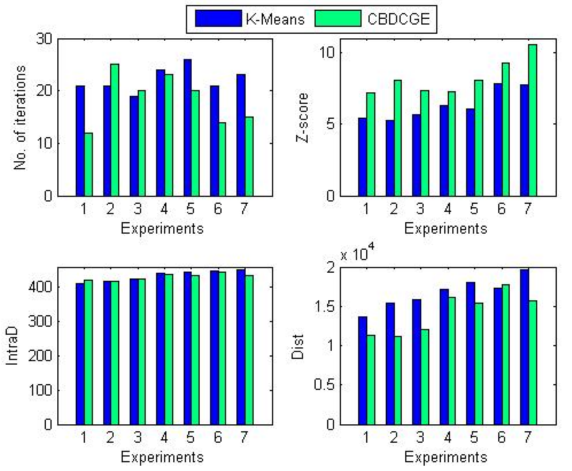
2.3.2 Evaluation measures

In order to measure the performance of our algorithms, the following evaluation measures are used: *IntraD*, *Dunn Dist* and *Z-score*. Good partions are reflected in low *IntraD* and *Dist* and high *Dunn* and *Z-score*. The first three measures evaluate the partition from the clustering point of view, while the last one from a biological perspective.

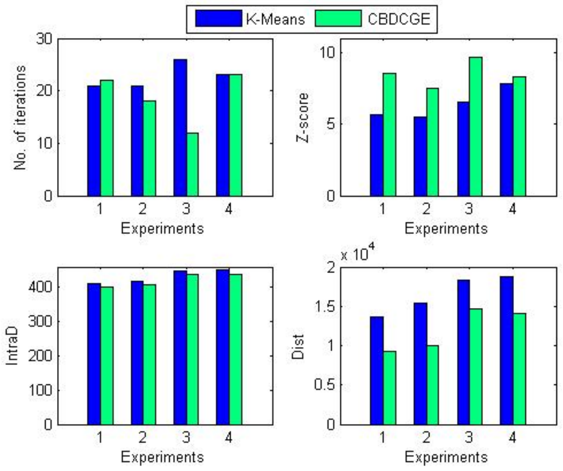
2.3.3 Results

Considering an initial number of features (denoted by m) characterizing the genes from the considered data set (Subsection 2.3.1), and different values for the threshold $distMin$ used for determining the initial centroids in the *k-means* and *fuzzy c-means* processes (see Subsection 2.2.2), the experiments are conducted as follows:

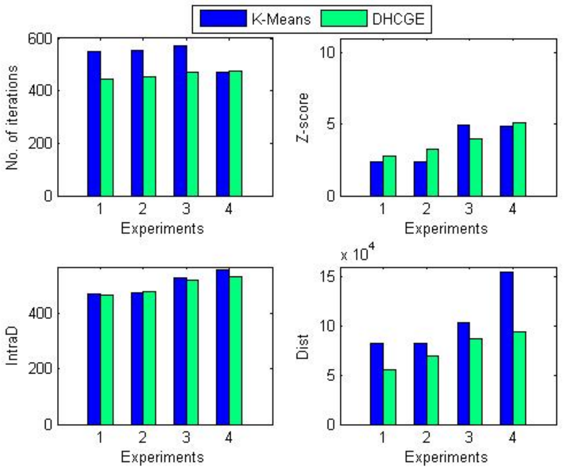
1. The *k-means*, *HACA*, respectively *fuzzy c-means* clustering algorithm is applied on the data set consisting of m-dimensional genes, starting from the identified centroids and a partition $\mathcal{K}$ is provided.
2. The set of features is now increased with s ($s \geq 1$) new features, numbered as $(m+1), \ldots, (m+s)$. The *CBDCGE*, *DHCGE*, respectively *FDCGE* adaptive algorithm is now applied, by adapting the partition $\mathcal{K}$ and considering the instances extended with the newly added s features.
3. The partition into clusters provided by *CBDCGE* algorithm (denoted by $\mathcal{K}_{alg_name}$) is compared with the one provided by the $k-means$ algorithm applied from scratch on the $m+s$-dimensional instances (denoted by $\mathcal{K}'$).



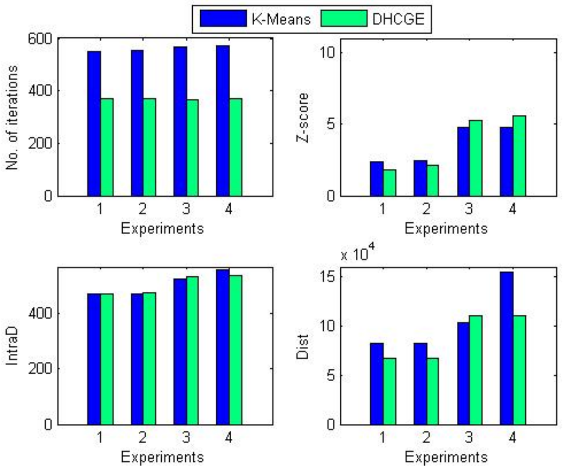
(a) Results for CBDCGE when $m = 5$ and $s = 2$



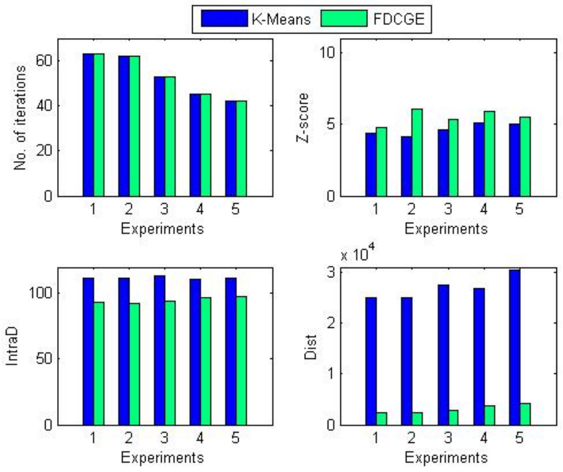
(b) Results for CBDCGE when $m = 6$ and $s = 1$



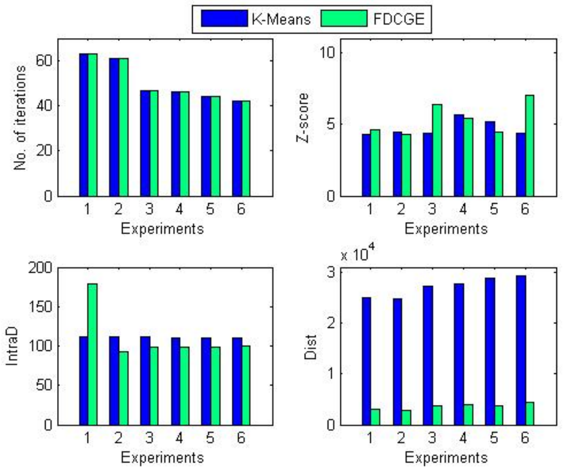
(a) Results for DHCGE when $m = 5$ and $s = 2$



(b) Results for DHCGE when $m = 6$ and $s = 1$



(a) Results for FDCGE when $m = 5$ and $s = 2$



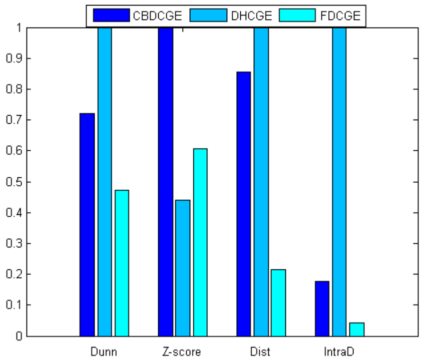
(b) Results for FDCGE when $m = 6$ and $s = 1$

2.4 Discussion

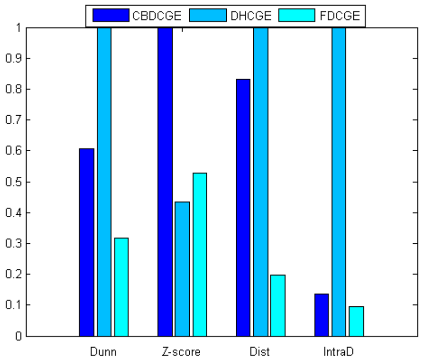
Considering the results above, we can conclude that for the partitions obtained adaptively by applying our dynamic clustering models (CBDCGE, DHCGE and FDCGE) are generally better than the ones obtained by applying k-means, hierarchical clustering or fuzzy c-means from scratch. Also, in most of the cases, the number of iterations is reduced by using our dynamic algorithms.

2.4.1 Comparative analysis of our algorithms

To our knowledge, except for the dynamic clustering algorithms that we have proposed in [14, 59, 60], there are no other approaches that deal with gene expression data sets for which new features (values for expression levels of genes at new time points) are added dynamically. For this reason, we will provide a comparison between the three models we proposed.



(a) Comparison of *CBDCGE*, *DHCGE*, *FDCGE*, starting with $m = 5$ features



(b) Comparison of *CBDCGE*, *DHCGE*, *FDCGE*, starting with $m = 6$ features

2.4.2 Comparison to related work

Since there are no other algorithms in the literature that approach the problem of the dynamic clustering of gene expression when new expression levels are added to the existing genes, we cannot provide a thorough comparison of our results to other. However, we note that the biological relevance of the partitions obtained using *CBDCGE*, quantified in the *z-score*, is significant.

Although our algorithm was designed with the purpose of providing an adaptive clustering technique for dynamic gene expression data sets, instead of a novel clustering method, we remark that, in terms of *z-score*, it outperforms other existing incremental clustering algorithms proposed for gene expression data sets, which are subject to change, in the sense that they are enriched with new instances [53, 22]. The *z-score* value acheived by our algorithm *CBDCGE* is **8.35**, while for *incDGC* 7.07 and *GenClus* 7.39.

2.4.3 Adaptive association rule mining of gene expression data

Starting from the idea of dynamic clustering of gene expression, when expression levels are added to the existing genes, we explored another dynamic process that takes place in the same context of feature-set extension: association rule mining.

Association rule mining [15] implies searching attribute value conditions which occur frequently together in a data set [65, 68].The *DRAR* method (*Discovery of Relational Association Rules*) was introduced for mining interesting relational association rules within data sets [57].

In the following, we introduce an algorithm called *ARARM* (*Adaptive Relational Association Rule Mining*) that is capable to efficiently mine relational association rules within a data set, when the feature set increases with one or more features.

Let us consider a data set $R = \{r_1, r_2, \dots, r_n\}$ consisting of n-dimensional *instances*. Each instance is described by a list of m attributes, $(a_1, \dots, a_m)$ and is therefore described by an m-dimensional vector $r_i = (r_{i1}, \dots, r_{im})$. Between the values of the features characterizing the instances from the data set, different types of relations can be defined. We denote by $\mathcal{R}el$ the set of all possible relations which can be defined between the features values.

The measured set of features is subsequently increased with s ($s \geq 1$) new features, numbered as $m+1, \dots, m+s$, the objects' vectors becoming $r_i^{ext} = (r_{i1}, \dots, r_{im}, r_{i,m+1}, \dots r_{i,m+s})$, $1 \leq i \leq n$. The set of extended instances is denoted by $R^{ext} = \{r_1^{ext}, r_2^{ext}, \dots, r_n^{ext}\}$.

Considering certain minimum support and confidence thresholds (denoted by s_{min} and c_{min}), we want to analyze the problem of mining interesting relational association rules within the data set R^{ext}, i.e. after object extension, and starting from the set of rules discovered in the data set R before the feature set extension. We aim at obtaining a better time performance with respect to the mining from scratch process. We denote in the following by $\mathcal{RAR}$ the set of interesting relational association rules with a minimum support and confidence within the data set R, and by $\mathcal{RAR}^{ext}$ the set of interesting relational association rules having a minimum support and confidence within the extended data set R^{ext} [19].

The *ARARM* algorithm identifies the interesting relational association rules using an iterative process which generates length-level rules, then verifies the candidates in order to comply the minimum support and confidence. *ARARM* performs multiple iterations over the data set R^{ext}. In the first iteration, the algorithm computes the support and confidence of the 2-length rules and decides which of them are interesting. All the subsequent iterations over the data are performed of two phases. The k-length ($k \geq 2$) rules from R^{ext} will certainly contain the k-length rules from $\mathcal{RAR}$ (the interesting rules discovered in the data set before extension) - if such rules exist. But, there is another possibility to obtain a k-length rule in the extended data set, through generating a candidate rule through joining two $k-1$-length rules from $\mathcal{RAR}^{ext}$ (generated at the previous iteration). In the second phase, a scan over R^{ext} is performed to calculate the support and confidence of the candidate rules generated as described above. The algorithm keeps the rules that are considered interesting (satisfy the minimum support and confidence requirements) and will use them in the next iteration. The process ends when no new interesting rules are found in the last iteration [17].

2.4.3.1 Experimental evaluation

In the following we present a set of experiments ment to evaluate the effectiveness of *ARARM* of the gene expression data set presented in 2.3.1. Like in the dyanamic clustering experiments, we are initially considering m attributes s and afterwards we extend the set of features with s attributes. We consider in the experiments different values for the confidence threshold (c_{min}) and different type of relational rukes (maximal rules vs. all rules). The minimum support threshold (s_{min}) is set to 1. For each experiment, the set of interesting relational association rules on the $m+s$ dimensional instances are obtained in two ways:

1. by applying the *DRAR* method from scratch on the data set after the feature set extension (containing all $m+s$ features).

2. by adapting (through the *ARARM* algorithm) the rules obtained on the data set before the feature set extension (containing m features).

Table 2.1 illustrates the performance of the *ARARM* method, for each of the performed experiment. It can be observed that the time needed to obtain the rules adaptively is less than the time needed to obtain the rules from scratch, indicating that our approach is more efficient for identifying association relational rules when the feature set is extended than applying the mining process from the beginning.

Experiment	c_{min}	**No. of attributes** (m)	**No. of added attributes** (s)	**Type of rules**	**No. of rules**	**Time from scratch (ms)**	**Time adaptive (ms)**
1	0.3	3	4	Maximal	53	118	**38**
2	0.3	4	3	Maximal	53	118	**32**
3	0.3	5	2	Maximal	53	118	**19**
4	0.3	6	1	Maximal	53	118	**12**
5	0.3	3	4	All	114	31	**33**
6	0.3	4	3	All	114	31	**15**
7	0.3	5	2	All	114	31	**13**
8	0.3	6	1	All	114	31	**4**
9	0.4	3	4	Maximal	33	49	**13**
10	0.4	4	3	Maximal	33	49	**12**
11	0.4	5	2	Maximal	33	49	**12**
12	0.4	6	1	Maximal	33	49	**10**
13	0.4	3	4	All	64	20	**9**
14	0.4	4	3	All	64	20	**7**
15	0.4	5	2	All	64	20	**6**
16	0.4	6	1	All	34	20	**2**

Table 2.1: Results for the gene expression data set for $s_{min} = 1$

2.5 Conclusions and further work

This chapter presented three models for dynamic clustering of gene expression data in the context where expression levels for new time points are added to the existing genes. The algorithms are capable of adapting the previously obtained partitions when new features (measurements of gene expression levels) are added to the data set, without performing re-clustering from scratch. In the same context of dynamic gene expression data we also introduced a method for adaptive relational association rule mining that is able to adapt the previously obtained set of interesting rules when new gene expression levels are added to the existing genes, without applying the mining algorithm from scratch.

These algorithms were presented in the original papers [14, 59, 60, 18]. The experimental evaluation that was performed on a real-life gene expression data set show that, in most of the cases, the clustering is reached more effectively and is also more accurate by using our proposed methods than by using the *k-means*, *hierarchical agglomerative clustering*, respectively *fuzzy c-means* algorithm from scratch on the feature-extended data. Also, the relational association rules are achieved faster using our method than applying the mining algorithm from the beginning.

Still, there are some situations when is better to perform a full-repartitioning of the feature-extended object set, instead of adapting the existing partition. Examples of such situation could be: the addition of a large number of expression levels for new time points or the addition of attributes with large information gain and contradictory information against the old attribute set.

Further work can be done in order to determine conditions to decide when is more appropriate to adapt (using *CBDCGE*, *DHCGE* or *FDCGE*) the partition of the feature-extended object set than to recompute partition from scratch using a classic clustering approach. We also plan to extend the experimental evaluation on other publicly available data sets and to investigate methods to automatically identify the distance threshold for the clusters (e.g. supervised lear-ning).

Chapter 3

Pedestrian detection. Background.

In this chapter we are presenting the background knowledge related to the supervised classification problem approached in the thesis, the problem of *dynamic pedestrian recognition* with onboard cameras, which represents a very challenging task with great importance in real life.

3.1 Feature extraction

Object representation plays a very important role in an object recognition system. In our particular case of pedestrian detection, choosing the most pertinent features for pedestrian characterization is a very challenging problem, as it is still uncertain how the human brain performs the recognition based on visual information [49]. An important step before feature computing is finding a region of interest (ROI), an area where the potential pedestrian could be found. According to [30] there are several ways to obtain ROIs: standard background substraction, sliding window, detection of independently moving objects or stereovision. After ROI have been identified, the feature extraction process can start. Features can be divided in two classes: global and local. Global features take the image as a whole. They include contour representations, shape descriptors and texture features, e.g. Histogram of Oriented Gradients, Local Binary Patterns, Haar features. Unlike global features, local features are computed at multiple points in the image (points of interest), having the advantage of being more robust to occlusion and clutter [40]. Examples of local features are Scale-Invariant Feature Transform, Speeded Up Robust Features, Haaris. There are also hybrid systems that combine local and global features to exploit the advantages of both representations [40], [11]. From another perspective, features can be classified into static (the ones mentioned above) and motion, which are able to capture motion from a sequence of images.

In the following we are presenting an overview on several global and local features, along with a family of motion features.

3.1.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) [20] are local feature descriptors commonly used in computer vision and pattern recognition, based on the idea that local object appearance and shape can often be well represented by the distribution of local intensity gradients or edge directions, even without exactly knowing the corresponding gradient or edge positions.

Their implementation supposes dividing the image window into small spatial regions, named cells, and accumulating for each cell a local 1-D histogram of gradient directions or edge orientations over the pixels, the final representation being obtained by combining the histogram entries. Contrast normalization of the local responses is performed for better invariance to illumination or shadowing, by accumulating a measure of local histogram over larger spatial regions, called blocks, and using the results for normalizing all of the cells in the block.

HOG features are used by most of state-of-the-art pedestrian detectors.

3.1.2 Histograms of Flow

Histograms of Flow (HOF) are a set of motions features, introduced in [21]. Their advantage compared to other motion features like the once introduced in [69] is that they are able to capture human motion from moving cameras against dynamic background, unlike the other ones that work only with static camera and background.

HOF use differential flow in order to cancel the effects of camera motion and voting similar to the one used in HOG to achieve a robust coding. Walk et al. [71] introduced a lower-dimensional variant of HOF and used it in the state of the art pedestrian detector MultiFtr+Motion from the benchmark presented in [25].

3.1.3 Local Binary Patterns

Local Binary Patterns (LBP) [45] are one of most widely used texture descriptors in computer vision, due to its invariance to gray level changes.

In order to compute LBP, first the ROI is divided into cells, e.g. 8×8 pixels. Each pixel in a given cell is compared with the pixels from its neighbourhood, obtaining a bit string ("1" if the center pixels value is smaller than the neighbours value and "0" otherwise). A histogram is computed over each cell, based on the decimal valued of transformed bit-string. The final feature vector is obtained by concatenating and normalizing the histograms of all cells

3.1.4 Scale-Invariant Feature Transform

Scale-invariant feature transform (SIFT) [41] are local features, invariant to image scale and rotation, used in computer vision and pattern recognition. In order to compute SIFT, first searches over all scales and image locations are efficiently performed, using a difference-of-Gaussian function in order to identify potential interest points which are invariant to scale and orientation. Then location and scale is determined at each candidate location, keypoints being selected based on measures of their stability. Each keypoint location is assigned one or more orientations, based on local image gradient directions. All following operations are performed on image data which has been previously transformed relative to the assigned orientation, location and scale, in this way achieving invariance to these transformations. Finally, all local image gradients are measured at the determined scale in the region around each keypoint, being converted into a representation which can handle important local shape distortion and change in illumination.

3.1.5 Speeded Up Robust Features

Speeded Up Robust Features (SURF) [9] are local features, scale and rotation invariant, used in computer vision for tasks like object recognition or 3D reconstruction. They are partly inspired by SIFT descriptors, but the authors proved that SURF detector is several times faster than SIFT, due to the integral images they make use of. The descriptor is based on sums of 2D Haar wavelet responses around the point of interest, detected using an integer approximation of the determinant of Hessian blob detector, that can be very rapidly computed on an integral image.

3.1.6 Haar Wavelets

Haar wavelets [48] were the first features used in a real time vision system [70]. Their main advantage is the fast computation, making use of integral images. In [48] the main configurations were proposed: horizontal, vertical and corner, then in [69] they were extended. The features are obtained by computing the difference between the sum of intensities in two rectangular areas with different configurations and sizes.

3.2 Classification

The performance of a pedestrian detection system is influenced by two important factors: the extracted features and the learning model. Support Vector Machines and boosting are the most widely used due to their theoretical guarantees, good performance and extensibility [25]. In the literature there are also approaches using Artificial Neural Networks and more recently, Deep Learning, a set of machine learning methods based on neural networks, e.g deep neural networks (DDNs), convolutional deep neural networks (CNNs), deep belief networks (DBNs) or recurrent neural networks (RNNs) [23].

3.2.1 Support Vector Machines

Support Vector Machines (SVM) [67] are a set of supervised learning methods widely used for classification and regression. Generally, classification is defined for the situation when there are m objects, each one belonging to one of the n classes, and a classification task would be to assign the belonging class to a new given object. In the case of binary classification using SVM, being given a set of training examples, each labelled as belonging to one of two categories, the SVM training algorithm constructs a model which predicts if a new example falls into one category or the other.

SVMs apply linear classification to non-linear classification problems with a technique known as "kernel trick", which implies using a kernel function that maps data points from the input space into a higher dimensional space, where data becomes linearly separable.

3.2.2 Adaptive Boosting

Adaptive Boosting (AdaBoost) [29] is a machine learning algorithm that constructs a strong classifier by combining weak classifiers (often rule-of-thumb) in an iterative greedy manner. Each new classifier focuses on miss-classified instances. AdaBoost is well known due to its speed optimized by the use of cascades and can be combined with any classifier to find weak rules. Its disadvantage is that is sensitive to noisy data and outliers.

In the following we give a short description of the main steps of the algorithm, as presented in [54]. Given a set of n training examples $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$, where x_i is a feature and y_i a label, several iterations are performed. In each iteration, a distribution is computed over the training examples and a *weak* learning algorithm is applied in order to determine a weak hypothesis with low weighted error relative to the considered distribution. The final hypothesis is obtained by computing the sign of the weighted combination of weak hypotheses.

3.2.3 Artificial neural networks

Artificial neural networks (ANNs) are a type of machine learning model inspired by biological neural networks. They are composed by *neurons* organized in *layers* and interconnected by *axons*, allowing a large number of possible combinations. ANNs are able to provide a non-linear decision and can also work with raw data, without needing an explicit feature extraction process. Their advantage is that they are not sensitive to incomplete or noisy data, being able to learn complex patterns, but they need a high training time and have many parameters to be tuned.

3.3 Literature review

There is a significant amount of research carried out in the area of pedestrian detection and recognition. A recent direction of research, on which we are going to focus in the next chapter, is represented by the combination of different features (multi-feature) and modalities (multi-modality), extracted from visible domain, such as intensity, motion information extracted from optical flow and depth information from the disparity map. Choosing a suitable fusion scheme in order to combine the information extracted is crucial. In the literature there are two classical fusion schemes: the early fusion, at the low level of features and the late fusion, at the high level of matching scores. Of course, there are also approaches based on hybrid fusion schemes.

In [21] is proposed a detector that uses a combination of appearance descriptors extracted from a single frame of a video with motion descriptors extracted from either optical flow or spatio-temporal derivates of the subsequent frame. The authors apply both low-level and high-level fusion schemes. In the low-level fusion appearance and motion features are concatenated in a large feature vector that feeds a single classifier, while in the high-level fusion a two stage classification is used. First, individual classifiers are trained on each type of features, then a classifier is used to combine their outputs. Their experiments show that the high-level fusion approach is slightly better than the low-level one and could be improved by combining the components in a more complex manner.

In [51] is proposed a high-level fusion of spatial features derived from dense stereo and intensity images. Two classifiers are trained on features extracted from depth, respectively intensity images, then their outputs are combined utilizing a set of fusion rules: maximum, product, sum and SVM rules. Experiments on Daimler pedestrian data set show that their high-level fusion approach outperforms the state-of-art intensity only model and the low-level fusion approach using a joint feature space.

In [26] are used for the first time together appearance, motion and stereo features for pedestrian recognition. The authors propose a multi-modality approach, which combines features extracted from images in three modalities: intensity, depth and flow into a mixture-of-experts framework. Later, in [27] the framework is extended, being able to combine information from multiple features and cues. On pose-level are used shape cues based on Chamfer shape matching, on modality-level are considered intensity, depth and flow modalities and on feature level are used HOG and LBP. Individual expert classifiers on pose, modality and feature levels are integrated through a probabilistic model. Experiments on Daimler data set show a significant performance boost over the state-of-art classifier using intensity only and the joint feature approach.

In addition to the multi-feature and multi-modality approaches, there are models which fuse information from different domains (multi-domain) like visible and far infrared domains (see [4], [10]).

Chapter 4

New approaches to pedestrian classification

In this chapter, we address the problem of pedestrian recognition in single and multi-modality images. The models introduced in this chapter are original, and were introduced in [3, 61, 52, 62, 11].

4.1 Problem statement and relevance

Pedestrian detection is one of the most popular research directions in the domain of object detection and computer vision. The number of vehicles has exponentially increased on the road over the last two decades. As a consequence the number of car accidents has increased too and along with that grew the need to develop better traffic safety mechanisms. Pedestrians represent a great part of the traffic and in order to protect them, different pedestrian detection systems were developed. The aim of these systems, called ADAS, is to improve the perception of the driver, in order to avoid collisions to pedestrians. It is very difficult to develop a very precise ADAS, mostly because of the way a pedestrian's appearance can vary. A pedestrian can change pose, carry different objects, have different shapes and heights, wear different clothes.

As presented in the previous chapter, pedestrian detection is performed in several steps. Firstly, regions of interest (ROIs) are identified within an image frame (hypothesis generation), then ROIs are classified into *pedestrian/non-pedestrian* (hypothesis refinement). In this thesis we are focusing on the latter step, which is reduced to a binary classification problem.

4.2 Pedestrian recognition using kernel descriptors

Kernel descriptors (KDs) [12] can be seen as a generalization of orientation histograms (including HOG), which are a particular type of match kernels over *patches* (regarded as a collection of *blocks*). Moreover, kernel descriptors intend to overcome some drawbacks of histograms based approaches, in which similarity between different image regions is determined based on their histogram. In order to compute the histogram, pixel values as discretized into bins, therefore quantization errors might be introduced.

The similarity between different images regions is determined based on a match kernel function. In [12] are introduced three matching kernels: the gradient match kernel, the color kernel and the local binary pattern kernel.

The **gradient match kernel**, K_{grad}, is able to capture image variations and is based on three kernels: a normalized linear kernel, an orientation kernel and a position kernel. The **color match kernel**, K_{col}, is able to describe image appearance and is based on two kernels: a color kernel and a position kernel. The **shape match kernel**, K_{shape}, is able to capture local shape and is based on three kernels: a kernel of standard deviations of neighbour pixels, a kernel of binarized pixel value differences in a local window and a position kernel.

The similarity between image patches is computed in a principled way through match kernels, but when image patches are large evaluating kernels might be computational expensive [13]. For computational efficiency and representational convenience, the authors use an approach for extracting the compact low-dimensional features from match kernels that consists of uniformly and densely sampling sufficient basis vectors from support region in order to guarantee accurate approximation to match kernels and then learning compact basis vectors using the kernel principal component analysis.

4.2.1 KDs representation in SVM vs. GP

Since the most important conditions for achieving a performant classification algorithm are using a suitable representation of the objects to be classified and a powerful decision making algorithm on top of it, we aim to investigate how efficiently can kernel descriptors represent data for two different learning algorithms: SVM and GP. Features are extracted using three type of kernels: Exponential, Gaussian and Laplacian.

4.2.1.1 Feature extraction

For extracting features we used the framework proposed by L. Bo in [12], for which we tested different kernel functions.

In the first step, based on the code developed by Xiaofeng Ren for kernel descriptors, we have tested different kernel functions for extracting local features from an image. Since the data set we work with contains only gray images, we investigated only the gradient kernel descriptor [12], which is able to capture image variations. As presented in Section 4.2, the gradient kernel descriptor is composed of three kernels: a kernel of magnitudes, an orientation kernel and a position kernel.

The magnitude kernel is a linear kernel and its type cannot be changed, because it must be an equivalent of the histogram of gradients. The other two kernels from the composition of the gradient kernel descriptor, the orientation kernel which computes the similarity of gradient orientations and the position kernel that measures the spatial closeness of two pixels, are in fact Gaussian kernels. Therefore, we changed the implementation in order to evaluate other kernels like Exponential and Laplacian for both orientation and position kernels.

4.2.1.2 Learning algorithms

In order to learn a classifier that is able to discriminate between pedestrians and non-pedestrians, we used two machine learning algorithms: SVM and GP.

4.2.1.2.1 SVM

Regarding the SVM, we have considered the LibSVM tool [16] because it is reliable and has many features implemented. Unlike Bo's framework [12], which uses the primal formulation of SVM, we chose an implementation of the Sequential Minimal Optimization (SMO) algorithm [50], since it is able to quickly solve the quadratic programming optimization problem of SVM.

In order to identify the most appropriate kernel for the SVM, we applied a Linear kernel, a Gaussian kernel, a Polynomial kernel and a Normalised Polyomial kernel with different parameters, deciding for the linear one.

4.2.1.2.2 GP

For the evolutionary classifier a linear and efficient GP version is actually utilized: Multi Expression Programming (MEP)[46]. MEP uses a linear representation of chromosomes and a mechanism to select the best gene for providing the output of the chromosome, unlike other GP techniques which use a fixed gene for output. Furthermore, no extra processing for repairing newly obtained individuals is required.

The advantages of dynamic-output chromosome over fixed-output chromosome are notable mostly in the situations in which the complexity of the target expression is not known. Variable-length expressions can be implicitly provided, while other techniques like Grammatical Evolution or Linear GP require special genetic operators that insert or remove chromosome parts for achieving such a complex functionality. Moreover, due to code reuse, the MEP has exponential length while the encoded expression may have exponential length [47].

4.2.1.3 Experimental evaluation

Several numerical experiments about how the discussed learning algorithms (an SVM and a GP-based classifier) are able to solve a particular image classification task (pedestrian recognition) are presented in this section. To evaluate the performance of the considered classifiers, the Daimler-Chrysler (DC) crop wise data sets (18 × 36 pixels image size) have been used as provided in [44]. Actually, a binary classification problem was solved: separate the images that contain pedestrians from the images that do not. 4480 images are considered: the decision model is trained on 2240 of them, while 2240 of images are used for testing.

In Table 4.1 are presented the accuracy rates (and their confidence intervals) by considering different kernels (when the image descriptors are actually constructed) and two learning algorithms (SVM and MEP). The performance measures are computed by taking into account the test images and the best identified classifiers (SVM with the best hyper-parameters and MEP with an optimal configuration).

	Gaussian	Exponential	Laplacian
SVM	0.657 ± 0.010	0.535±0.010	0.599 ± 0.010
MEP	0.682 ± 0.009	0.667 ± 0.009	0.737 ± 0.009

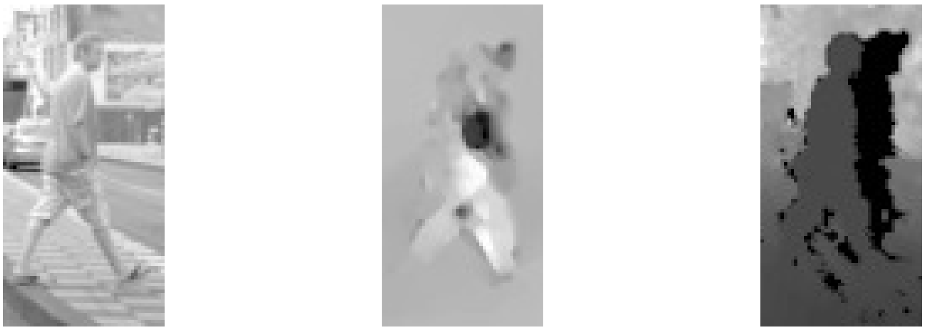
Table 4.1: Accuracy rates (%) obtained by SVM and MEP algorithms on images represented by different kernel descriptors.

4.2.1.4 Conclusions

A study on how two learning algorithms are able to perform pedestrian recognition in images is presented in this section. Daimler-Chrysler benchmark image dataset is involved in our numerical experiments.

The first step is to convert each image in a numerical representation relevant for the classifier. Several kernel descriptors are considered on this purpose: Exponential, Gaussian and Laplacian kernels. A statistical algorithm, SVM, and an evolutionary approach, MEP, are used for the learning phase for which the input data is represented by the previously extracted features.

Better accuracy rates are obtained when using the evolutionary model for all considered kernel descriptors. Regarding the kernel descriptors used, SVM learning indicates that the Gaussian is the best one, while MEP achieves the best results by using the Laplacian kernel. Therefore, we can not conclude which is the most efficient kernel descriptor and we intend to perform a further study of how the kernel selection influences the quality of recognition.



(a) Intensity (b) Flow (c) Depth

Figure 4.1: Pedestrian sample from Daimler dataset in (a) intensity, (b) flow and (c) depth images.

4.2.2 Single vs. multi-modality pedestrian recognition

In this section we aim at investigating how does the kernel descriptor based classifier work for single vs. multi-modality images. For the multi-modality approach we considered three modalities from visible domain: intensity, depth and flow. We decided to address only the visible domain, since it's less expensive and employs less complex models.

4.2.2.1 Feature extraction

For the feature extraction phase we chose kernel descriptors [12], presented in Section 4.2, because they overcome some drawbacks of histograms based techniques as mentioned in Section 4.2.

4.2.2.2 Learning algorithm

For the learning phase we use Support Vector Machines (SVM) [66]. There are two types of kernels used by SVMs: linear and non-linear. Even if the non-linear SVMs perform better than the linear ones, the improvement of results is paid with high computational costs and memory which is a big problem for pedestrian detection systems which must perform in real-time. From these reasons, we chose Liblinear [28], a very popular implementation of linear SVM.

4.2.2.3 Experimental evaluation

In the experiments we use Daimler pedestrian dataset [26], which contains a collection of manually labelled pedestrian and non-pedestrian bounding boxes in images captured from a vehicle-mounted calibrated stereo camera rig in an urban environment. The dataset contains 48 x 96 px image crops in three modalities: intensity, depth and optical flow, which represents the reason for choosing this dataset. Figure 4.1 illustrates a pedestrian in each of these modalities.

The training set contains a total number of 84577 examples (52112 pedestrians and 32465 non-pedestrians) and the testing set 41834 examples (25608 pedestrians and 16235 non-pedestrians).

4.2.2.3.1 Evaluation measures

Designing classifiers for image recognition is a complex task, traditionally conducted by optimizing a single criterion: prediction accuracy. A such performance measure falls short of expectations when data are described by skewed class distributions or in the case of unbalanced training data. A solution to this problem is represented by a more complex criterion utilised in the training phase of the classification: the Receiver Operating Characteristics (ROC) curve.

In order to compare two models, we use the FPR at 90% detection rate, which is the reference value in the literature. Good performances are indicated by small values for FPR.

4.2.2.3.2 Parameter optimization

Since the performance on the classification process strongly depends on the parameters involved in both stages (feature extraction and learning), our approach involves an optimization phase dedicated to identify the best parameters of the decision model.

First of all, the parameters involved in the description computing are optimized. Taking into account that the images we work with are gray, we use the Gradient Kernel Descriptor. In order to obtain the best representation for our particular images containing pedestrians and non-pedestrians, the parameters of the kernel descriptor, *kdesdim* and *contrast*, have to be optimized on each type of image (intensity, depth and flow). The *kdesdim* parameter establishes the number of features that will be extracted from each patch, while the *contrast* is a parameter used by the gradient kernel. These two parameters are optimized by using a cascade approach. The parameters that minimize the FPR obtained at 90% detection rate (the ones in bold) are chosen for each type of image:

- *kdesdim* = 54 and *contrast* = 0.78 for intensity images
- *kdesdim* = 54 and *contrast* = 0.76 for depth images
- *kdesdim* = 56 and *contrast* = 0.69 for flow images

4.2.2.3.3 Single-modality classification

Using the optimal KD parameters dicovered through the optimization step, we extract KD features for each type of image from the entire training set and then we learn an SVM model.

The results obtained on the testing set for each of the three classifiers are presented in Figure 4.2. We can notice that the best results, reflected by the smaller FPR, are obtained on intensity images.

4.2.2.3.4 Multi-modality classification

After having analysed the effect of each modality independently for KD features, we now evaluate the effect of using modality fusion. The fusion is performed by joining the features extracted from intensity, depth and flow images. An SVM model is learned on the obtained feature vectors.

The results obtained on the testing set for each of the three classifiers are presented in Figure 4.2. We can observe an improvement when fusing the information provided by different modalities.

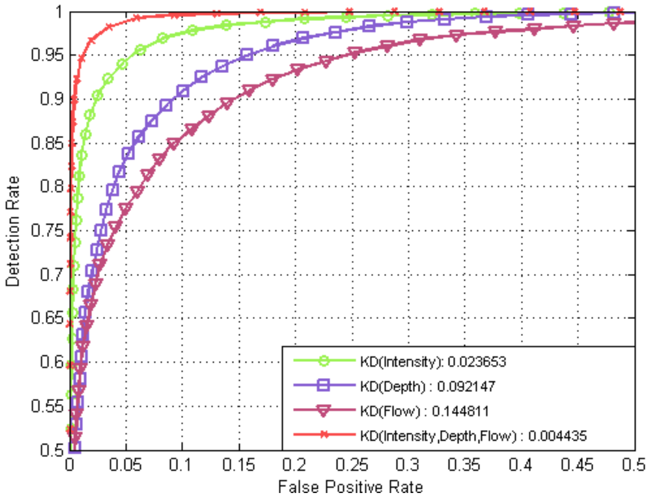


Figure 4.2: Single-modality *vs.*. multi-modality classification performance on testing set, using KD features. FPR at 90% detection rate.

4.3 Pedestrian detection using dynamic modalities

It was proved that the fusion of features extracted from multi-modality images like intensity, depth and flow, improves the performance of pedestrian recognition (see Section 4.2.2). We have also proved it for KD features in Section 4.2.2.3. However, in some conditions, pedestrians are recognized only in a particular modality. For example, a pedestrian in an low contrast image is more difficult to be recognized in intensity than in other modalities.

In this section we propose two machine learning based algorithms that are able to dynamically select the most discriminative modalities for each image sample, representing a ROI within an image frame, that will be further used in the classification process.

4.3.1 Fusion models

In the literature, there are two classical fusion schemes applied for object classification: an early fusion, prior to classification, which involves the construction of a joint feature space that will forward feed the classifier, and a late fusion, posterior to classification, that involves a fusion of classification scores. Lets consider the case of the fusion among intensity, depth and flow.

The drawback to the late fusion is the high number of false positives, since the false positive rate increases from intensity to depth and flow (see [26]). On the other hand, some pedestrians are recognized only in a particular modality, for e.g. in flow. For instance, if the classifiers in intensity and depth decide that the image contains a non-pedestrian, the majority vote will lead to a missclassification. Because some pedestrians can be discriminated only in a particular modality, the problem of missclassification will occur also in other types of fusion that weight the importance of a modality in the same manner for all images (bounding boxes within a given frame), in a static way, or use the confidence indicator of the classifier in each modality.

The drawback to the eary fusion is the high number of false positives, since the false positive rate increases from intensity to depth and flow (see [26]). On the other hand, some pedestrians are recognized only in a particular modality, for e.g. in flow. For instance, if the classifiers in intensity and depth decide that the image contains a non-pedestrian, the majority vote will lead to a missclassification. Because some pedestrians can be discriminated only in a particular modality, the problem of missclassification will occur also in other types of fusion that weight the importance of a modality in the same manner for all images (bounding boxes within a given frame), in a static way, or use the confidence indicator of the classifier in each modality.

4.3.2 Modality selection component

The modality selection component represents the core of the dynamic models that we propose, being responsible of determining the most suitable modalities for classifying an image. Its design is based on a hybrid approach, which involves two types of classifiers: a pedestrian classifier, which is able to discriminate between pedestrians and non-pedestrians, and a modality pertinence classifier that decides whether a modality is suitable or not for a given image.

The main idea of the modality pertinence classifier is to learn from the experience of the pedestrian classifier in a particular modality if that modality is suitable or not for the classification of an image, representing a bounding box within a frame. We consider a modality suitable for an image, if the pedestrian classifier was able to correctly classify the image using that modality on a validation set.

4.3.3 Dynamic modality selection

The Dynamic Modality Selection (DMS) is able to dynamically select for each image, representing a bounding box in a frame, the most relevant image modality among intensity, depth and flow.

DMS uses the modality pertinence classifier presented in section 4.3.2 in order to find the modalities suitable for the current image and chooses one among them, according to a modality relative confidence indicator.

The steps performed by DMS are the following:

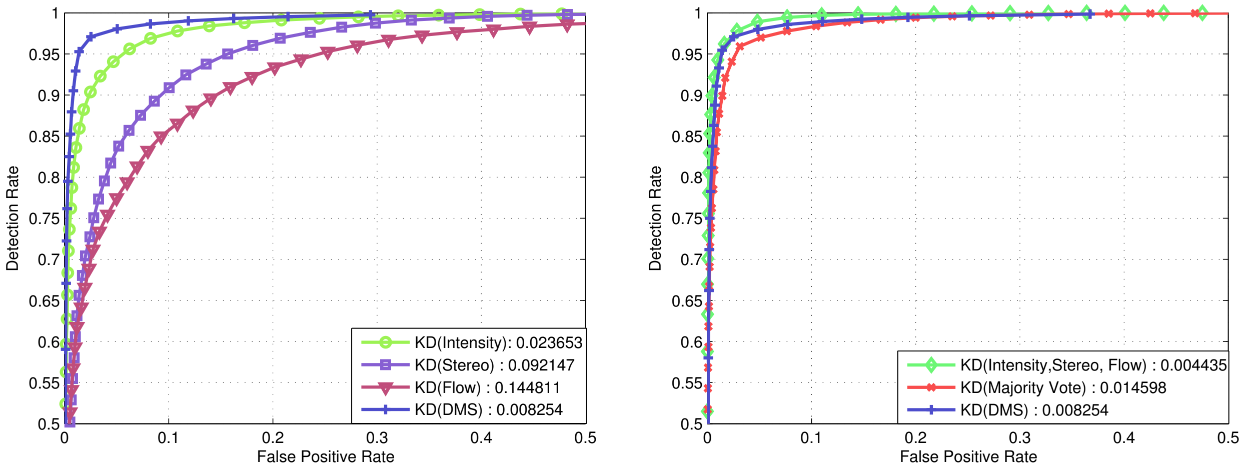
- check for all modalities if they are suitable for a given image, using the modality pertinence classifiers

- if there is more than one modality supposed to be suitable for the given image, choose the most confident one, which satisfies $Max\{|P_i^P - P_i^{NP}|, i \in \{I, D, F\}\}$, where P^P represents the probability of being a pedestrian, while P^{NP} the probability of being a non-pedestrian (modality relative confidence indicator)

- classify the image into *pedestrian* or *non-pedestrian*, using the most confident from the suitable modalities discovered in the first step, or the most confident among all modalities if none of them is considered suitable

4.3.3.1 Experimental evaluation

In Figure 4.3a we present the results obtained by the DMS classifier on the testing set, in comparison with the results achieved by the single-modality classifiers (intensity, depth and flow). In Figure 4.3b we compare our results to the ones obtained by the fusion models presented in section 4.3.1: the classifier trained on the concatenated modalities, evaluated for KD features in [61], and the majority vote classifier.



(a) Single modality vs. DMS classification performance on testing set, using KD features. FPR at 90% detection rate.

(b) Modality fusion vs. DMS classification performance on testing set, using KD features. FPR at 90% detection rate.

We can notice that DMS achieves better performance than the single-modality classifiers and the majority vote classifier, and is very close to the performance obtained by concatenating the three modalities.

4.3.3.2 Conclusions

We proposed a dynamic single-modality selection approach which is able to select the most suitable modality to classify an image.

The advantages of our method over the joint modalities approach consist of:

- *lower complexity* – the amount of time needed for training individual classifiers is less than the one needed to train a classifier on a large feature vector, obtained by joining all modalities
- the individual classifiers can be *trained independently* on different datasets

The results achieved by the joint modalities classifier are slightly better than the ones obtained by DMS, but taking into account that its performance is strongly influenced by the amount of experiences of the pedestrian classifier in each modality (number of training examples for the modality classifiers) from which it could achieve a performance boost, we find DMS a promising alternative for the joint modalities classifier.

In the next section, we propose a model which combines the modality selection and the joint modalities approaches in order to benefit from the advantages of both models.

4.3.4 Dynamic modality fusion

The Dynamic Modality Fusion (DMF) is able to dynamically determine the most suitable modalities for a given image and perform a fusion of the features extracted from them.

DMF also uses the modality classifier presented in section 4.3.2 in order to find the modalities suitable for the current image, but unlike DMS which selects only one modality, it takes into account all the modalities which were considered suitable and further uses them in the classification process.

The steps performed by DMF are the following:

- check for all modalities if they are suitable for the given image, using the modality classifiers
- if in the first step we get more than one modality that is supposed to be suitable for the given image, join the features extracted from all suitable modalities
- if no modality is considered suitable, join the features extracted from all modalities
- classify the image into *pedestrian* or *non-pedestrian*, using the modalities discovered in the first step

4.3.4.1 Experimental evaluation

In Figure 4.4a we present the results obtained by the DMF classifier on the testing set, in comparison with the results achieved by the single-modality classifiers (intensity, depth and flow). In Figure 4.4b the DMF model is compared to static fusion approaches where the classifier is trained on the concatenated modalities, evaluated for KD features in [61], and for a majority vote classifier.

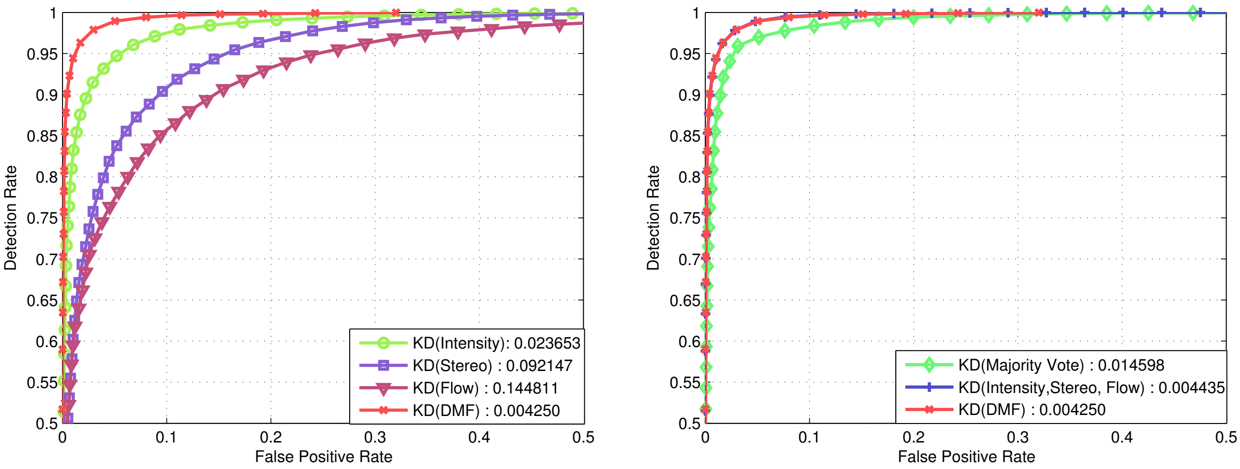
We can notice that DMF outperforms both models, but its improvement over the joint modalities classifier is rather small. In the next section we are going to present some perspectives of improvement of the DMF.

4.3.4.2 Conclusions

We proposed a dynamic multi-modality fusion approach which is able to select the most suitable modalities to classify an image and join the features extracted from them.

The advantage of DMF over the joint modality approach is that is less prone to the errors caused by non pertinent modalities, while the advantage over the single modality classifiers rises from the situations when the individual modality classifiers cannot discriminate correctly between pedestrian and non-pedestrians, unless they join their features. Considering all the aspects mentioned above, we find DMF model, a promising approach for multi-modality pedestrian recognition.

Further work could be done to investigate methods to improve the modality selection component and also to apply DMF on other datasets. Some possible directions of improvement could be the



(a) Single modality vs. DMF classification performance using KD features. FPR at 90%.

(b) Static modality fusion vs. DMF classification performance, using KD features. FPR at 90%.

integration of image based modality-pertinence indicators, a deeper optimization of kernel descriptors (the Gaussian kernels from Gradient Kernel Descriptor and the dimensionality reduction step) or the addition of other features like Shape Kernel Descriptors.

Conclusions

In this thesis we have focused on applying dynamic machine learning models to solve *supervised* and *unsupervised* classification problems. The particular problems that we have decided to approach are gene expression clustering and pedestrian recognition, because they are very challenging, have a great importance in real life and are representative for the two types of classification: unsupervised and supervised.

In the first research direction, we have introduced three dynamic models for clustering gene expression data, in the context where expression levels for new time points are added to the existing genes. The algorithms (CBDCGE, DHCGE and FDCGE) are capable of adapting the previously obtained partitions when new measurements of gene expression levels are added to the dataset, without performing re-clustering from scratch. The experimental evaluations that we have performed on a real-life gene expression data set show that, in most of the cases, the clustering is reached more effectively and is also more accurate by using our proposed methods than by applying the k-means, hierarchical agglomerative clustering, respectively fuzzy C-means algoritms from scratch on the feature-extended data. But there are also some situation when the partitions are too difficult to adapt after the addition of new attributes and a full repartition should be considered.

In the same context of dynamic gene expression data, we have proposed a new adaptive association rule mining method (ARARM), which is capable to adapt the set of interesting rules discovered when new gene expression levels are added to the dataset, without performing re-mining from scratch. Experiments on the same gene expression dataset show that ARARM reaches the solution faster than applying the mining algorithm from the beginning.

Furher work in the first research direction could be done in order to determine conditions to decide when it is more appropriate to adapt (using CBDCGE, DHCGE or FDCGE) the partition of the feature-extended object set than to recompute partition from scratch using a classic clustering approach. We also plan to extend the experimental evaluations on other publicly available datasets and to investigate methods to automatically identify the distance threshold for the clusters (e.g. using supervised learning).

In the second research direction, we have addressed two problems. First, we have performed a study on the efficiency of using kernel descriptors in pedestrian recognition, since they obtained good results for visual recognition and to our knowledge have not been used for this task so far. Then, we have introduced two dynamic algorithms, DMS and DMF, that are able to determine the most suitable modalities among intensity, depth and flow to classify an image, representing a bounding box within an image frame, and further include them in the classification process.

Kernel descriptors have proved to achieve good performances both in single and multi-modality images. We have optimized the parameters of the gradient kernel independently on each modality, using a grid search algorithm. The selection of the most appropriate kernel (Exponential, Gaussian, Laplacian) depends not only on the images, but also on the learning algorithm. Even if they are considered a generalization of HOG, they do not reach their performances and this could be caused by the KPCA component used for dimensionality reduction.

Experimental evaluations on a pedestrian dataset show that the dynamic modality selection and fusion models, DMS and DMF, represent promising approaches for multi-modality pedestrian recognition. The first one has the advantages of lower complexity, individual training on modalities, while the second one achieves a higher performance boost. Moreover, the dynamic fusion schemes that we have proposed in our models are generic and could be applied to other problems which need a dynamic integration of the sources.

Further work in the second research direction could be done to extend and improve our dynamic models (e.g. integrating image based modality-pertinence indicators, adding other features to the fusion, or using others) and also to evaluate them on other datasets. Moreover, improvements could be brought to kernel descriptors by optimizing the Gaussian kernels (position kernel and orientation kernel for Gradient Kernel Descriptor) and adapting the dimensionality reduction process in order to retain the most relevant information.

Bibliography

[1] S. Albayrak and F. Amasyali. Fuzzy c-means clustering on medical diagnostic systems. In *Turkish Symposium on Artificial Intelligence and Neural Networks - TAINN*, 2003.

[2] L. An and R.W. Doerge. Dynamic clustering of gene expression. *ISRN Bioinformatics*, 2012:1–12, 2012.

[3] A. Andreica, L. Diosan, R. D. Gaceanu, and A. Sirbu. Pedestrian recognition by using kernel descriptors. *Studia Universitas Babes-Bolyai, Seria Informatica*, LVIII(2):77–89, 2013.

[4] A. Apatean, C. Rusu, A. Rogozan, and A. Bensrhair. Visible-infrared fusion in the frame of an obstacle recognition system. In *Automation Quality and Testing Robotics (AQTR), Cluj-Napoca*, pages 1 – 6, 2010.

[5] C. Arima, T. Hanai, and M. Okamoto. Gene Expression Analysis Using Fuzzy K-Means Clustering. *Genome Informatics*, 14:334–335, 2003.

[6] Abraham Bagherjeiran, Christoph F. Eick, Chun-Sheng Chen, and Ricardo Vilalta. Adaptive clustering: Obtaining better clusters using feedback and past experience. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, pages 565–568, Washington, DC, USA, 2005. IEEE Computer Society.

[7] A.M. Bagirov and K. Mardaneh. Modified global k-means algorithm for clustering in gene expression data sets. In *Proceedings of the 2006 workshop on Intelligent systems for bioinformatics*, WISB '06, pages 23–28, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[8] Z. Bar-Joseph, G. Gerber, D.K. Gifford, and T.S. Jaakkola. A New Approach to Analyzing Gene Expression Time Series Data. In *Proceedings of the sixth annual international conference on Computational biology*, RECOMB '02, pages 39–48, 2002.

[9] H. Bay, T. Tuytelaars, and L. J. Van Gool. Surf speeded up robust features. In *ECCV*, pages 404–417, 2006.

[10] Bassem Besbes, Sonda Ammar, Yousri Kessentini, Alexandrina Rogozan, and Abdelaziz Bensrhair. Evidential combination of SVM road obstacle classifiers in visible and far infrared images. In *Intelligent Vehicles Symposium*, pages 1074–1079. IEEE, 2011.

[11] Bassem Besbes, Alexandrina Rogozan, Adela-Maria Rus, Abdelaziz Bensrhair, and Alberto Broggi. Pedestrian detection in far-infrared daytime images using a hierarchical codebook of surf. *Sensors*, 15(4):8570, 2015.

[12] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. In *NIPS*, pages 244–252. Curran Associates, Inc, 2010.

[13] Liefeng Bo and Cristian Sminchisescu. Efficient match kernel between sets of features for visual recognition. In Yoshua Bengio, Dale Schuurmans, John D. Lafferty, Christopher K. I. Williams, and Aron Culotta, editors, *NIPS*, pages 135–143. Curran Associates, Inc, 2009.

[14] Maria Iuliana Bocicor, Adela Sirbu, and Gabriela Czibula. Dynamic core based clustering of gene expression data. *International Journal of Innovative Computing, Information and Control*, 10(3):1051–1069, 2014.

[15] Toon Calders, Nele Dexters, Joris J. M. Gillis, and Bart Goethals. Mining frequent itemsets in a stream. *Inf. Syst*, 39:233–255, 2014.

[16] C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines.* Online, 2001.

[17] Gabriela Czibula, Maria-Iuliana Bocicor, and Istvan Gergely Czibula. Promoter sequences prediction using relational association rule mining. *Evolutionary Bioinformatics*, 8:181–196, 04 2012.

[18] Gabriela Czibula, Istvan-Gergely Czibula, Adela Sirbu, and Gabriel Mircea. A novel approach to adaptive relational association rule mining. *Applied Soft Computing*, 36:519–533, November 2015.

[19] Gabriela Czibula, Zsuzsanna Marian, and István Gergely Czibula. Detecting software design defects using relational association rule mining. *Knowl. Inf. Syst*, 42(3):545–577, 2015.

[20] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005.

[21] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, pages II: 428–441, 2006.

[22] R. Das, D.K. Bhattacharyya, and J.K. Kalita. An Incremental Clustering of Gene Expression data. *World Congress on Nature and Biologically Inspired Computing. NaBIC 2009.*, pages 742–747, 2009.

[23] Li Deng and Dong Yu. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3-4):197–387, 2014.

[24] J.L. DeRisi, P.O. Iyer, and V.R. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, 1997.

[25] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012.

[26] Markus Enzweiler, Angela Eigenstetter, Bernt Schiele, and Dariu M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *CVPR*, pages 990–997. IEEE, 2010.

[27] Markus Enzweiler and Dariu M. Gavrila. A multilevel mixture-of-experts framework for pedestrian classification. *IEEE Transactions on Image Processing*, 20(10):2967–2979, 2011.

[28] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, (9):1871–1874, 2008.

[29] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS: Journal of Computer and System Sciences*, 55, 1997.

[30] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, 73(1):41–59, jun 2007.

[31] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*, volume 54. Morgan Kaufmann, 2006.

[32] J. Herrero and J. Dopazo. Combining hierarchical clustering and self-organizing maps for exploratory analysis of gene expression patterns. *Journal of Proteome Research*, 1(5):467–470, 2002.

[33] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.*, 31(8):651–666, 2010.

[34] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, 1988.

[35] J.-Y. Jiang, W.-H. Cheng, and S.-J. Lee. A dissimilarity measure for document clustering. *ICIC Express Letters*, 6(1):15–21, 2012.

[36] M. G. Kendall and A. Stuart. *The Advanced Theory of Statistics, volume III.* Griffin, London, 1966.

[37] K. Kim, S. Zhang, K. Jiang, L. Cai, I.B. Lee, L.J. Feldman, and H. Huang. Measuring similarities between gene expression profiles through new data transformations. *BMC Bioinformatics*, 8(29), 2007.

[38] Wei Li, Rujing Wang, Liangtu Song, and Xiufang Jia. Batch dynamically incremental c-means clustering algorithm based on rough fuzzy set. *Journal of Computational Information Systems*, 11(5):1553–1561, 2015.

[39] Yan Li, Snigdha Verma, Li Lao, and Jun-Hong Cui. SACA: SCM-based adaptive clustering algorithm. In *MASCOTS*, pages 271–279. IEEE Computer Society, 2005.

[40] Dimitri A. Lisin, Marwan A. Mattar, Matthew B. Blaschko, Mark C. Benfield, and Erik G. Learned-Miller. Combining local and global image features for object class recognition. In *In Proceedings of the IEEE CVPR Workshop on Learning in Computer Vision and Pattern Recognition*, pages 47–55, 2005.

[41] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[42] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S.J. Brown. Incremental genetic k-means algorithm and its application in gene expression data analysis. *BMC Bioinformatics*, 5(172), 2004.

[43] Tom Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[44] S. Munder and S. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(11):1863–1868, nov 2006.

[45] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, jan 1996.

[46] Mihai Oltean. Improving the search by encoding multiple solutions in a chromosome. In Nadia Nedjah and Luiza de Macedo Mourelle, editors, *Evolutionary Machine Design: Methodology and Applications*, Intelligent System Engineering, chapter 4, pages 85–110. Nova Publishers, 2005.

[47] Mihai Oltean, Crina Grosan, Laura Diosan, and Cristina Mihaila. Genetic programming with linear representation: a survey. *International Journal on Artificial Intelligence Tools*, 18(2):197–238, 2009.

[48] M. Oren, C. P. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *CVPR*, pages 193–199, 1997.

[49] I. Parra Alonso, D. Fernandez Llorca, M. A. Sotelo, L. M. Bergasa, P. Revenga de Toro, J. Nuevo, M. Ocana, and M. A. Garcia Garrido. Combination of feature extraction methods for SVM pedestrian detection. *IEEE Trans. Intelligent Transportation Systems*, 8(2):292–307, April 2007.

[50] J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

[51] Marcus Rohrbach, Markus Enzweiler, and Dariu M. Gavrila. High-level fusion of depth and intensity for pedestrian classification. In *DAGM-Symposium*, volume 5748 of *Lecture Notes in Computer Science*, pages 101–110. Springer, 2009.

[52] Adela-Maria Rus, Alexandrina Rogozan, Laura Diosan, and Abdelaziz Bensrhair. Pedestrian recognition by using a dynamic modality selection approach. In *IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, September 15-18*, pages 1862 – 1867, 2015.

[53] S. Sarmah and D.K. Bhattacharyya. An effective technique for clustering incremental gene expression data. *International Journal of Computer Science Issues*, 7(3):31–41, 2010.

[54] Robert E. Schapire. Explaining adaboost, 08 2013.

[55] G. Serban and A. Campan. Incremental clustering using a core-based approach. In *Proceedings of the 20th international conference on Computer and Information Sciences*, ISCIS'05, pages 854–863, Berlin, Heidelberg, 2005. Springer-Verlag.

[56] Gabriela Serban and Alina Campan. Hierarchical adaptive clustering. *Informatica*, 19(1):101–112, 2006.

[57] Gabriela Serban, Alina Campan, and Istvan Gergely Czibula. A programming interface for finding relational association rules. *International Journal of Computers, Communications and Control*, I(S.):439–444, jun 2006.

[58] Adela Sirbu. A study on dynamic clustering of gene expression data. *Informatica*, LIX(1):16–27, 2014.

[59] Adela Sirbu and Maria-Iuliana Bocicor. A dynamic approach for hierarchical clustering of gene expression data. In *Intelligent Computer Communication and Processing (ICCP)*, pages 3–6, Sept 2013.

[60] Adela-Maria Sirbu, Gabriela Czibula, and Maria-Iuliana Bocicor. Dynamic clustering of gene expression data using a fuzzy approach. In *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014, Timisoara, Romania, September 22-25, 2014*, pages 220–227. IEEE, 2014.

[61] Adela-Maria Sirbu, Alexandrina Rogozan, Laura Diosan, and Abdelaziz Bensrhair. Pedestrian recognition by using a kernel-based multi-modality approach. In *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014, Timisoara, Romania, September 22-25, 2014*, pages 258–263. IEEE, 2014.

[62] Adela-Maria Sirbu, Alexandrina Rogozan, Laura Diosan, and Abdelaziz Bensrhair. Pedestrian recognition using a dynamic modality fusion approach. In *Proceedings of IEEE International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, September 3-5*, pages 393 – 400, 2015.

[63] D. Stekel. *Microarray Bioinformatics.* Cambridge University Press, Cambridge, UK, 2006.

[64] Xiaoke Su, Yang Lan, Renxia Wan, and Yuming Qin. A fast incremental clustering algorithm. In *Proceedings of the 2009 International Symposium on Information Processing*, pages 175–178, 2009.

[65] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining.* Addison-Wesley, 2005.

[66] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[67] V. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[68] Renato Vimieiro and Pablo Moscato. A new method for mining disjunctive emerging patterns in high-dimensional datasets using hypergraphs. *Inf. Syst*, 40:1–10, 2014.

[69] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, jul 2005.

[70] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[71] Stefan Walk, Nikodem Majer, Konrad Schindler, and Bernt Schiele. New features and insights for pedestrian detection. pages 1030–1037. IEEE Computer Society, 2010.

[72] Fei Wu and Georges Gardarin. Gradual clustering algorithms. In *DASFAA*, pages 48–55. IEEE Computer Society, 2001.

[73] X. Xiao, E.R. Dow, R.C. Eberhart, Z.B. Miled, and R.J. Oppelt. Gene Clustering Using Self-Organizing Maps and Particle Swarm Optimization. *In: Proc. 17th Intl. Symposium on Parallel and Distributed Processing*, 2003.

[74] N. Yano and M. Kotani. Clustering gene expression data using self-organizing maps and k-means clustering. *Proceedings of SICE 2003 Annual Conference*, 3:3211–3215, 2003.

[75] Y. Yuhui, C. Lihui, A. Goh, and A. Wong. Clustering gene data via associative clustering neural network. *In: Proc. 9th Intl. Conf. on Information Processing*, pages 2228–2232, 2002.