T.C.
BAHÇEŞEHİR ÜNİVERSİTESİ

REDUCING LEARNING COMPLEXITY IN MULTI-VIEW CLASSIFICATION MODELS

Master Thesis

HEYSEM KAYA

İSTANBUL, 2009

T.C.

BAHÇEŞEHİR ÜNİVERSİTESİ

INSTITUTE OF SCIENCES

COMPUTER ENGINEERING

REDUCING LEARNING COMPLEXITY IN MULTI-VIEW CLASSIFICATION MODELS

Master Thesis

HEYSEM KAYA

Supervisor: ASST. PROF. DR. OLCAY KURŞUN

İSTANBUL, 2009

T.C.

BAHÇEŞEHİR ÜNİVERSİTESİ

INSTITUTE OF SCIENCES

COMPUTER ENGINEERING

Name of the thesis: Reducing Learning Complexity in Multi-View Classification Models

Name/Last Name of the Student: Heysem KAYA
Date of Thesis Defense: 26 August 2009

The thesis has been approved by the Institute of Sciences.

Prof. Dr. A. Bülent ÖZGÜLER
Director

_____

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Science.

Examining Committee Members	Signature
Advisor: Asst. Prof. Dr. Olcay KURŞUN	_____
Prof. Dr. Nizamettin AYDIN	_____
Prof. Dr. Ali GÜNGÖR	_____

ACKNOWLEDGMENTS

This thesis is dedicated to **my family,** from the youngest to eldest, who encouraged my pursuit of high objectives. I would like to express my kind thanks to my eldest nephew **Ergin** who provided me a continuous zest of study with his triggering questions.

I would like to express my gratitude to my supervisor **Dr. Olcay KURŞUN** for everything he taught throughout my Master's study.

I thank **Dr. Selim MİMAROĞLU** for the sound equipment he provided in the Data Mining course.

ABSTRACT

REDUCING LEARNING COMPLEXITY
IN MULTI-VIEW CLASSIFICATION MODELS

KAYA, Heysem

Computer Engineering

Supervisor: Asst. Prof. Dr. Olcay KURŞUN

August 2009, 48 Pages

In pattern recognition, using all the available features as a single input vector to a classifier is known to worsen the generalization of the learning algorithm due to the phenomenon known as the curse of dimensionality, which stands for the diminishing coverage of the feature space with fixed number of data points as the feature set size increases. Most studies so far concerned with features individually, however some high dimensional datasets do contain features naturally organized into several groups, which are known as “views” in the literature. Techniques in multi-view learning exploit multiple views of the data samples, one of the typical examples of which is the audio versus video of a human speaking. Such different modalities as audio and video could help each other in making improved classification if their decisions are fused. Multi-view methods can be more successful than single view learning techniques in that they can exploit independent properties of each view and more effectively learn complex distributions. As the features in a view is a natural combination, feature selection

techniques are not directly applicable to such datasets because that would involve picking some features from each view and fusing them into a single feature vector, resulting in the aforementioned curse of dimensionality or over-learning considerations. In this thesis, several methods for feature selection are tailored to fit to the context of multi-view classification so as to avoid the curse of high input dimensionality. Aim of the study was to find efficient methods for selecting those views, which cooperatively perform as well as or better than the single-view counterpart (i.e. the whole set of features fused into a single feature vector for each sample of the dataset) and besides, extracting features from those views to enhance subsequent learning process. The results of these methods are compared to draw a road map in multi-view classification problems.

Keywords: Feature Selection; Feature Extraction; Curse of Dimensionality; Multi-View ARTMAP; Multi-View Nearest Neighbor; Multi-View Naïve Bayes; Protein Sub-nuclear Location Classification; Diagnosis of Parkinson's Disease; Data Mining; Pattern Recognition.

ÖZET

ÇOK BAKIŞLI SINIFLANDIRMA MODELLERİNDE
ÖĞRENME KARMAŞIKLIĞININ AZALTIMI

KAYA, Heysem

Bilgisayar Mühendisliği

Tez Danışmanı: Yard. Doç. Dr. Olcay KURŞUN

Ağustos 2009, 48 Sayfa

Örüntü tanımada, mevcut bütün değişkenlerin bir sınıflandırıcıya tek bir girdi vektörü olarak verilmesi öğrenme algoritmasının genelleştirme yeteneğini boyutsallığın laneti olarak bilinen olgudan dolayı zayıflatır ki bu olgu değişken kümesinin büyüklüğü arttıkça, değişken uzayının sabit sayıda veri noktasıyla daha az karşılanmasını ifade eder. Şu ana kadarki çoğu çalışma değişkenler ile bireysel olarak ilgilendi, ancak bazı yüksek boyutlu verikümeleri literatürde "bakış" olarak bilinen çeşitli doğal gruplara ayrılmış değişkenler içerir. Çok-bakışlı öğrenmedeki teknikler veri örneklerinin farklı bakışlarından, ki bir insan konuşmasının görüntü ve sesi buna tipik bir örnektir, en üst düzeyde faydalanır. Görüntü ve ses gibi farklı boyutlar eğer kararları birleştirilirse birbirine daha iyi sınıflandırma yapmak için yardımcı olabilir. Çok-bakışlı yöntemler her bakışın bağımsız değişkenlerinden yararlanabilmeleri ve karmaşık dağılımları daha etkin bir şekilde öğrenmeleri noktalarında tek bakışlı yöntemlerden daha faydalıdır.

Bir bakış içindeki değişkenler doğal bir kombinasyon olduğundan değişken seçim teknikleri bu tür verikümelerine doğrudan uygulanamaz çünkü her bakıştan bazı değişkenleri seçip bunları tek bir değişken vektörü içinde birleştirmek önceden bahsedilen boyutsallığın laneti hususu ile aşırı öğrenme hususundan dolayı verimsiz olabilir. Bu tezde, değişken seçimi için kullanılan çeşitli yöntemler yüksek girdi boyutsallığının lanetinden sakınmak amacıyla çok-bakışlı sınıflandırma bağlamına uyarlanmıştır. Bu çalışmanın amacı, birarada olduğunda en az verikümesinin tek bakışlı hali (verikümesindeki her örnek için bütün değişkenlerin tek bir değişken vektörü teşkil edecek şekilde birleştirilmesi) kadar iyi bakışları seçmek ve bunun yanında bir sonraki öğrenme sürecinde kullanılmak üzere bu bakışlardan değişken özütlemektir. Bu yöntemlerin sonuçları çok-bakışlı sınıflandırma problemlerinde bir yol haritası çizmek için karşılaştırılmıştır.

Anahtar Kelimeler: Değişken Seçimi; Değişken Özütleme; Boyutsallığın Laneti; Çok Bakışlı ARTMAP; Çok Bakışlı En Yakın Komşu; Çok Bakışlı Naïve Bayes; Protein Çekirdekaltı Yer Sınıflandırma; Parkinson Hastalığının Teşhisi; Veri Madenciliği; Örüntü Tanıma.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Artificial Neural Networks	:	ANN
Forward Selection	:	FS
k-Medoids	:	k-M
k-Medoids and Naïve Bayes	:	K-MNB
k-Medoids and Fuzzy ARTMAP	:	K-MART
k-Nearest Neighbor	:	k-NN
Maximum Relevance Minimum Redundancy	:	mRMR
Naïve Bayes	:	NB
Ranking Selection	:	RS
Radial Basis Function	:	RBF
Support Vector Machines	:	SVM
View	:	V

LIST OF SYMBOLS

Conditional probability of H on X	:	$P(H \mid X)$
Joint probability of H and X	:	$P(H , X)$
Learning rate parameter of Fuzzy ARTMAP	:	β
Mutual information between X and Y	:	$I(X; Y)$
Vigilance parameter of Fuzzy ARTMAP	:	ρ

1. INTRODUCTION

Feature selection/extraction is a preprocessing for subsequent pattern recognition/machine learning tasks. This is needed because as the feature set size increases, reliable classification is impaired by the diminished coverage of the feature space with the fixed number of data points obtained by costly experimental processes, which is a phenomenon known as the curse of dimensionality (Bishop, 1995). Reducing the feature space dimensionality to a minimal yet descriptive size is crucial for effective classification/regression models (Guyon and Elisseeff, 2003).

In some datasets, features are naturally organized into several groups, which are known as “views” in the literature (Yarowsky, 1995; Blum and Mitchell, 1998; Christoudias et al., 2008). Just like in the single-view problems with high input dimensionality that need feature selection as preprocessing, the multi-view methods also need mechanisms to fuse information from different views to overcome the problems with high input dimensionality. In this thesis, several methods for feature selection and extraction are adapted to the multi-view classification setting because single-view feature selection techniques are not directly applicable to such multi-view datasets. Single-view feature selection methods can pick some features from each view and merge them into a single feature vector. However, this approach would run into the curse of dimensionality and over-learning problems.

In some areas, such as chemistry, medicine, and bioinformatics it is hard and time consuming to attain data samples, and moreover, the data samples may have a huge number of features. Therefore the results of this study especially apply to the latter where data is limited in samples and whose highly-dimensional feature space contains natural groups of variables (views).

Replacing single-view feature selection with using multiple views, it is possible to dramatically lower computational demands to combining classifiers (Okun and Priisalu, 2005). When having to work with, for example, thousands of variables naturally

organized into tens of views, the computational complexity reduces from millions to only hundreds by several orders of magnitude. The number of feature subsets chosen heuristically and evaluated by feature selection methods will have to increase by merging all the views to get a single feature vector, out of which feature selection to be applied. This would cause the danger of overfitting because it is more likely to find a feature subset that fits well with the dataset at hand. However, the success which stems from this probable overfitting will be controversial due to under-sampled, unevenly distributed, multivariate nature of data. Therefore implications of feature selection methods which are also computationally very costly will not have a scientific validity.

Secondly, the views can correspond to very different modalities such as video data and audio data, in such a case fusing the low level video features with low level audio features is not desirable because the low level features in each view must first be combined within their own views in order to yield useful high-level descriptors, which can then be fused with the high-level descriptors from the other views, hierarchically. On the other hand, an example of a bad low level feature combination can be an attempt to evaluate a pixel feature together with an amplitude feature, neither of which is yet high-level. This is not just an inefficiency consideration caused by fusing all the views, this unnecessary expansion of the input space, combined with small sample sizes which are typical of experimental sciences, would greatly complicate the learning task (the curse of dimensionality).

Moreover, in some cases views may contain features which have many-to-many interrelations. This notion is referred as *View Conditional Independence* (Blum and Mitchell, 1998; Christoudias et al., 2008). In these cases input fusion (combining data without any evaluation process) will require more samples for generalization than output fusion which is the case for multi-view evaluation. This problem takes us back to curse of dimensionality problem in under-sampled datasets.

In this study, a series of methods and techniques were elaborated for selection, extraction, and classification purposes for multi-view datasets. The experimental datasets used in this study are two biomedical datasets: 1) Protein Structure Prediction

dataset which was introduced in the work of Nanuwa and Seker (2008), 2) Parkinson's Disease Diagnosis from Vocal Features (Little et al., 2008). The datasets under study were classified using Fuzzy ARTMAP Neural Network, k-Nearest Neighbor and Naïve Bayes. Also, k-Medoids clustering algorithm was used to provide an intermediate output for classification by Naïve Bayes and Fuzzy ARTMAP. These stacking settings are called K-MNB and K-MART, respectively. Fuzzy ARTMAP and k-NN were also used to rank views according to their individual classification power. The views are sorted according to their individual classification accuracy and loaded as input into the classifier iteratively. After incremental loading and testing, the set of views with maximum performance was selected. Forward Selection served as a benchmark to compare with both total set (all views fused into a single view) and only selected views. Other non-heuristic selection techniques such as random selection and brute force (evaluation of all subsets, exponentially) (Okun and Priisalu, 2005) are not covered in this thesis. A simple and well known method, namely k-NN, was firstly used for classification. The performance of k-NN with full set of features considered as a baseline. Then I exploited variants of ARTMAP method on the fused features of individually best views. I applied Naïve Bayes approach as the multi-view technique. Each view is evaluated individually using a simple k-Medoids approach or a more complex Support Vector Machines (SVM) approach. Then the classifications of these methods on all the views are given to Naïve Bayes (NB) for fusing these probability estimates. Although, even as its name implies, the NB approach is very simple, it was found to be very effective because it used all the views independently and then merged their prediction outputs. Investigation of K-MNB led to design of K-MART stacking network which provided the best results in this study. Successful stacking methods suggest that fusing several views into a single-view is not as effective as fusing the classifier outputs of the views.

The thesis layout is as follows. In Section 2, the literature on combining multiple learners is reviewed pointing out to similarities and differences of ensemble and multi-view. In Section 3, the methods and techniques used in this study are introduced. In Section 4 the experimental studies and results are given. In Section 5 the conclusions are provided and recommendations for future works are discussed.

2. LITERATURE REVIEW

2.1. Combining Multiple Learners

Machine learning studies elaborated on several combination techniques which benefits from decisions of multiple learners with different algorithms, hyperparameters, subproblems and training sets (Alpaydın, 2004). The rationale depends on the fact that there is no algorithm that is always accurate (No Free Lunch Theorem).

Most commonly used learner combining methods are voting, bagging, boosting, mixture of experts, stacking and cascading.

When used in classification, voting is a weighted summation for each class label where weights should sum up to 1. For example, weights can be assigned to be identical (1/n) or determined empirically by using the classifier accuracy on a validation set.

$$y_i = \sum_{j=1}^{N} w_j d_{ji} \qquad 2.1$$

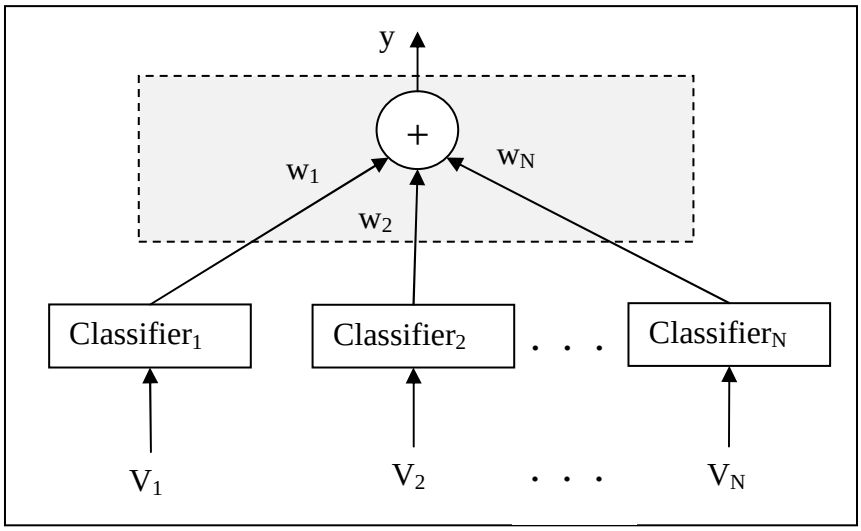

Figure 2.1: Voting Mechanism

While boosting iteratively handles misclassified samples in order to form a composite classifier by a weighted vote (Alpaydın, 2004), bagging creates a set of aggregate classifiers with bootstrapping (random sampling with replacement) whose votes are equally weighted.

In stacking, the combiner is another learner (Alpaydın, 2004), as it is shown in Figure 2.2, the outputs of individual learners are given as input to it. Note that individual learners need not be supervised learners.

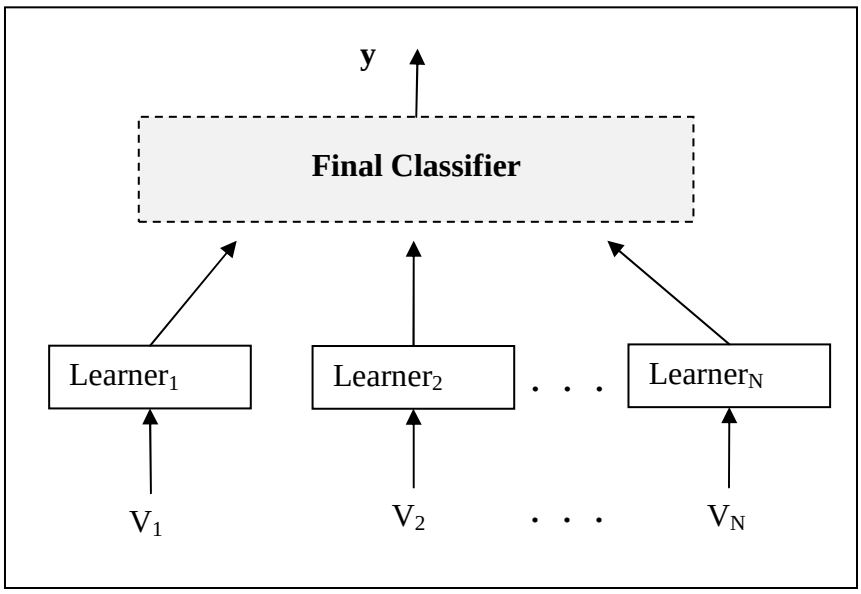

Figure 2.2: Stacking Mechanism

2.2. Comparison of Multi-View with Ensemble

2.2.1. Ensemble

Ensemble learning refers to a collection of methods that learn a target function by training a number of individual learners and combining their predictions. Ensembles can

be generated by subsampling the training examples, manipulating the input features, and modifying the learning parameters of the classifier. It is also possible to generate ensembles using views inherent in the dataset.

Accuracy and efficiency are advantages of ensembles. In terms of accuracy, a more reliable mapping can be obtained by combining the output of multiple experts due to No Free Lunch Theorem. On behalf of efficiency, a complex problem can be decomposed into multiple sub-problems that are easier to understand and solve (divide-and-conquer approach).

Uncorrelated errors of individual classifiers can be eliminated through averaging. The desired target function may not be implementable with individual classifiers, but may be approximated by ensemble averaging.

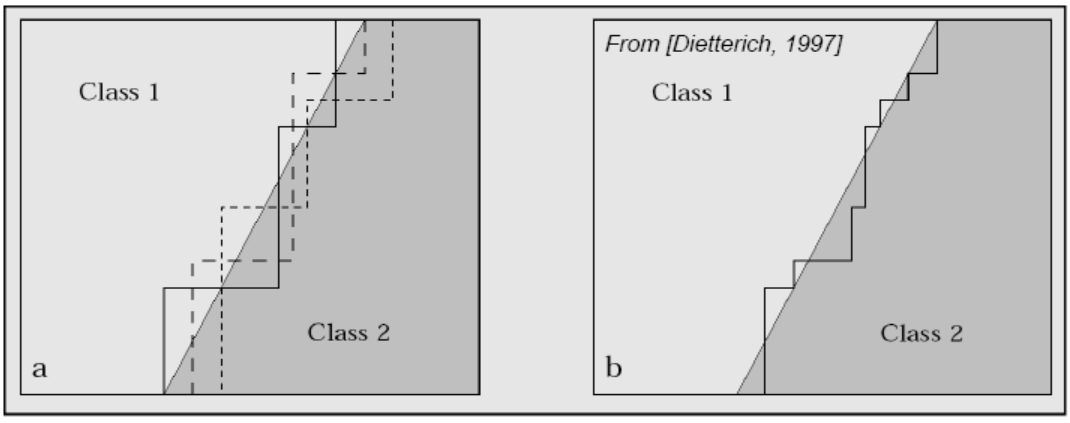

Figure 2.3: Smoothing by Ensemble Averaging

Dietterich (1997) explains the success of ensemble with statistical, computational and representational reasons. The statistical reason is that there is no sufficient data. The computational reason is the trap in local minimal. The representational reason is the same with No Free Lunch Theorem.

Although the terminology differs in ensemble and multi-view, in this thesis they will be used interchangeable since the ensembles will be generated from independent views.

2.2.2. Multi-View

Multi-view learning refers to learning the target concept from several disjoint subsets (views) of features each of which are sufficient to learn the target concept. Multi-view learning is useful when examples are not all labeled identically by classification from each view and given the label of any example, its descriptions in each view are independent (Blum and Mitchell, 1998; Muslea et al., 2002; Christoudias et al., 2008).

Increasing the classification accuracy is the common goal in both ensemble and multi-view learning techniques. However multi-view is especially used when the dataset has natural views and when learning is semi-supervised. Table 2.1 summarizes the differences between the two.

	Ensemble	**Multi-view**
Problem setting	Partition feature into multi-view	Given multi-view
Framework	Supervised learning	Semi-supervised learning

Table 2.1: Difference between Ensemble and Multi-View Learning Methods

2.3. Prominent Studies in Multi-View Learning

The techniques using multiple views in learning exploit independent properties of each view and more effectively learn complex distributions. In other words, the reason to use

multiple views instead of using one view is that combinations of views are able to explain more than single view (Bickel and Scheffer, 2004; Okun and Priisalu, 2005).

Empirical success of multi-view approaches has been noted in many areas of computer science including Natural Language Processing, Computer Vision, and Human Computer Interaction (Christoudias et al., 2008). Multi-view classification attracts many researchers recently because there is yet no known “best” way of fusing the information in the views. Works on multi-view machine learning gained importance since Yarowsky (1995) and Blum and Mitchell (1998) pointed out that multiple views can lead to better classification accuracy than the union of all views. Bickel and Scheffer (2004) showed that multi-view clustering performs better than single view clustering even though the setting contains only two views which they argued either one suffices for learning.

Kakade and Foster (2007) also argue that the main importance of the multi-view technique is that weaknesses of one view are complemented by the others. This finding is also supported by studies of Dietterich (1997; 2000).

3. METHODS

3.1. Forward Selection

Forward selection algorithm for the multi-view setting is implemented similar to its traditional single-view use (Bishop, 1995), but with one exception: a group of variables (view) is selected at a time instead of a single variable.

Forward selection starts with an empty set of views and loads the view with best predictive power. In subsequent iterations the view giving the best predictive power together with already existing view(s) is merged into the set if the total prediction rate is increasing.

Algorithm: Forward Selection of views
Input: *D*: a data set containing m views
Output: A set of selected views
Method:

```
1.  Add all views to vector unselvw
2.  Instantiate vector selvw
3.  float maxsc= 0.0
4.  repeat
5.       vw= null
6.       for each v in unselvw do
7.            vector curvw= selvw U v
8.                 train(train_set,curvw)
9.            float sc=test(test_set,curvw)
10.           if (sc>maxsc)
11.                maxsc=sc
12.                vw=v
13.           end if
14.      end for
15.      if (vw != null)
16.           selvw.add(vw)
17.           unselvw.remove(vw)
18.           increase=true
19.      end if
20. until no increase;
```

Figure 3.1: Forward Selection Algorithm used in the study

3.2. Selection by Ranking

Ranking of views was provided by filter method mRMR (Peng et al., 2005), and classifier methods ARTMAP and k-NN. Ordered by their rank, views are fused incrementally and classifier performances were calculated. In this method set of views with best performance was to be selected.

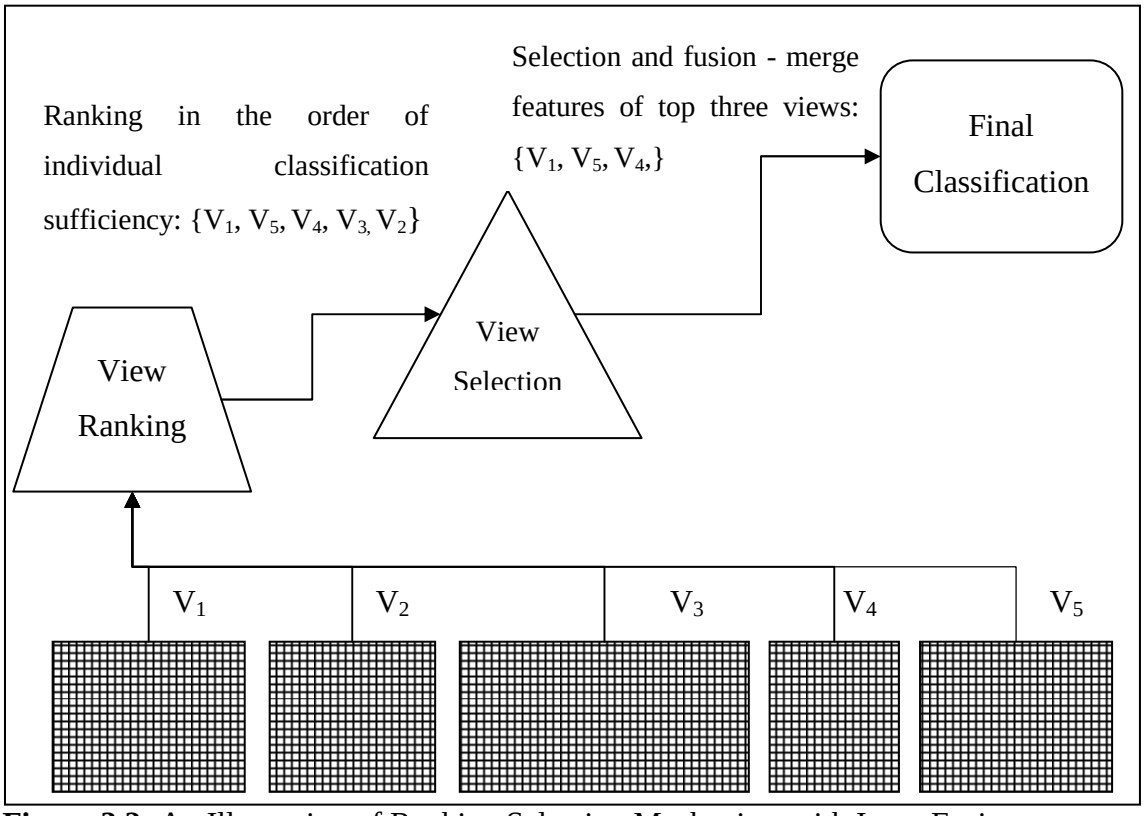

Figure 3.2: An Illustration of Ranking Selection Mechanism with Input Fusion

Figure 2.2 illustrates a RS mechanism with a hypothetical problem setting comprising five views. View ranking process results as shown in the figure: V_1, V_5, V_4, V_3 and V_2. Then view selection process fuses these views in the given order one by one performing a classification task for the resultant view set (i.e. at first $\{V_1\}$, next $\{V_1, V_5\}$, then $\{V_1, V_5, V_4\}$ and so on). Prediction rates of fused sets are recorded so that the view set with highest prediction success is proposed for subsequent learning. In the figure the view set $\{V_1, V_5, V_4\}$ is proposed. In output fusion, the only difference is that feature vectors of

views are evaluated by a method and then information such as cluster indices or class membership distribution is provided for selective classification.

3.3. k-Nearest Neighbor

k-NN is a widely used pattern recognition algorithm. There are multi-view variants of this method which utilize boosting and bagging. Recently other boosted k-NN variants are introduced. Koon (2007) proposes direct boosting using local warping of distance metric, in which incorrectly classified samples update the weights of their neighbors. The algorithm used in this study is an adaptation of the simple k-NN algorithm.

Algorithm: Modified simple k-NN
Input:
k: number of nearest neighbors used for majority voting
D: set of training samples
prob: training set label distribution (prior probability), complementary parameter
o: sample object to classify
Output: class label of sample
Method:

```
1. vector nn = get_nearest_neighbors(k,D,o) //gets nearest k neighbors of o from D
2. vector elected= majority_vote(nn) // does a majority voting and returns the class labels having the max vote
3. if (elected.size() > 1) //if there is a tie get the elected label having max prior prob
4.      return get_max_prior_prob(elected, prob)
5. else
6.      return elected.get(0)
```

Figure 3.3: k-NN algorithm used in the study

There could be tie among class labels having maximum votes for k>1. Inspired by decision tree generation algorithms, I have introduced an additional majority voting mechanism to handle such ties.

Ensemble of k-NN classifiers were not used in this study as in the work of Okun and Priisalu. However, data patterns of selected views are merged before entering this process.

3.4. Fuzzy ARTMAP Neural Network

Fuzzy ARTMAP is a fast and stable classification algorithm which is capable of incremental learning (Carpenter et al., 1992) hence superior to Multi Layer Perceptron (Busque & Parizeau, 1997). Fuzzy ARTMAP achieves a synthesis of fuzzy logic and adaptive resonance theory (ART) neural networks by exploiting a close formal similarity between the computations of fuzzy subsethood and ART category choice, resonance, and learning (Carpenter et al., 1992). Fuzzy ARTMAP also realizes a minimax learning rule that concointly minimizes predictive error and maximizes code compression, or generalization (Carpenter et al., 1992). Fuzzy ARTMAP is composed of two Fuzzy ARTs. Fuzzy ART is an ANN for unsupervised learning which was introduced by Carpenter, Grossberg and Reynolds in 1991.

3.4.1. Fuzzy ARTMAP Architecture

The Fuzzy ARTs contained in ARTMAP ANN are identified as ART_a and ART_b. The parameters of these networks are designated respectively by the subscripts a and b. The two Fuzzy ARTs are interconnected by a series of connections between the F_2 layers of ART_a and ART_b. The connections are weighted, i.e. a weight w_{ij} between 0 and 1 is associated with each one of them. These connections form what is called the map field F^{ab}. The map field has two parameters (β_{ab} - learning rate and ρ_{ab} - vigilance) and an output vector x_{ab}(Carpenter at al.,1992; Busque & Prizeau, 1997). Vigilance can be defined as sensitivity to new data patterns. While, small vigilance values increase code compression (generalization) leading to larger category boxes, bigger vigilance values result in increased number of categories. Category proliferation is hindered by normalizing input vectors at preprocessing stage (Carpenter at al.,1992).

During supervised learning ART_a receives an a stream $\{a^{(p)}\}$ of input patterns, and ART_b receives a stream $\{b^{(p)}\}$ of input patterns, where $b^{(p)}$ is the correct prediction given $a^{(p)}$. These modules are linked by an associative learning network and an internal controller that ensures autonomous system operation in real time. The controller is designed to create the minimal number of ART_a recognition categories, needed to meet accuracy criteria using a mechanism called *match tracking* (Carpenter, Grossberg & Reynolds, 1991).

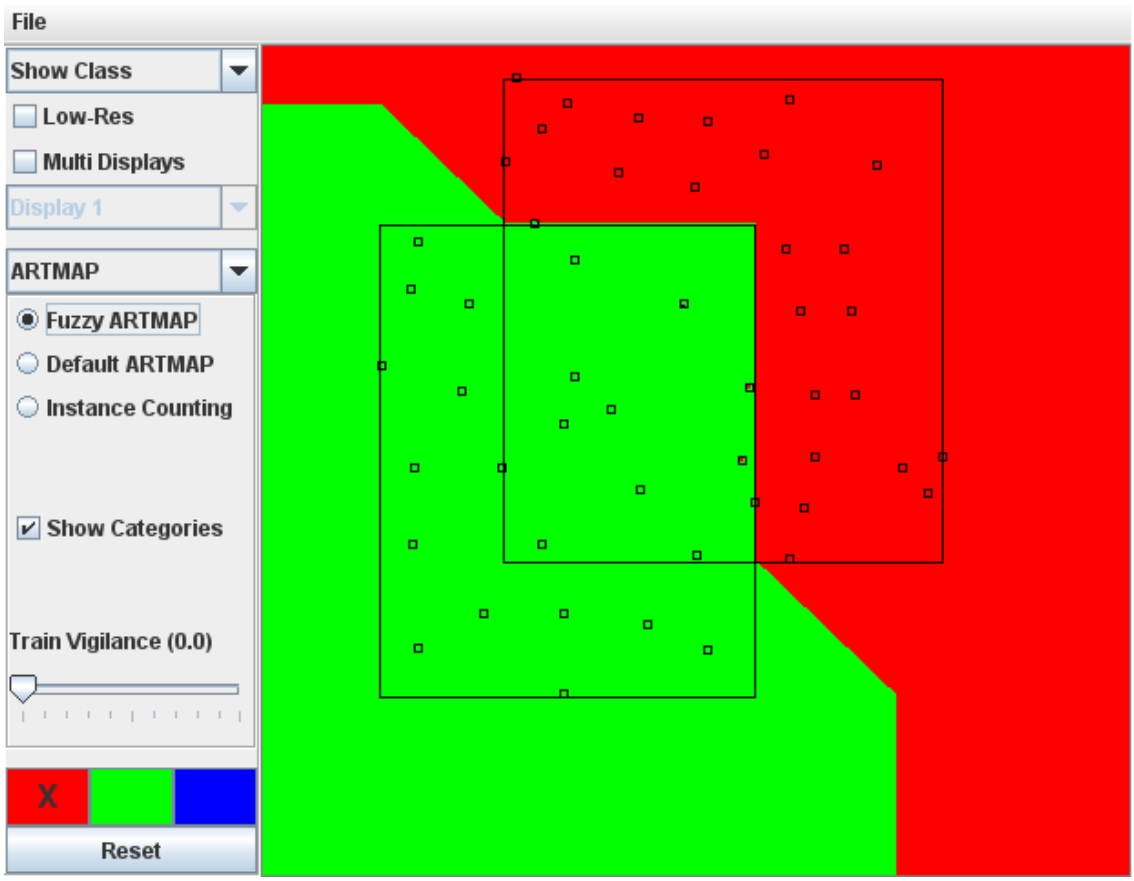

Figure 3.4: Fuzzy ARTMAP Illustration with $\rho = 0.0$

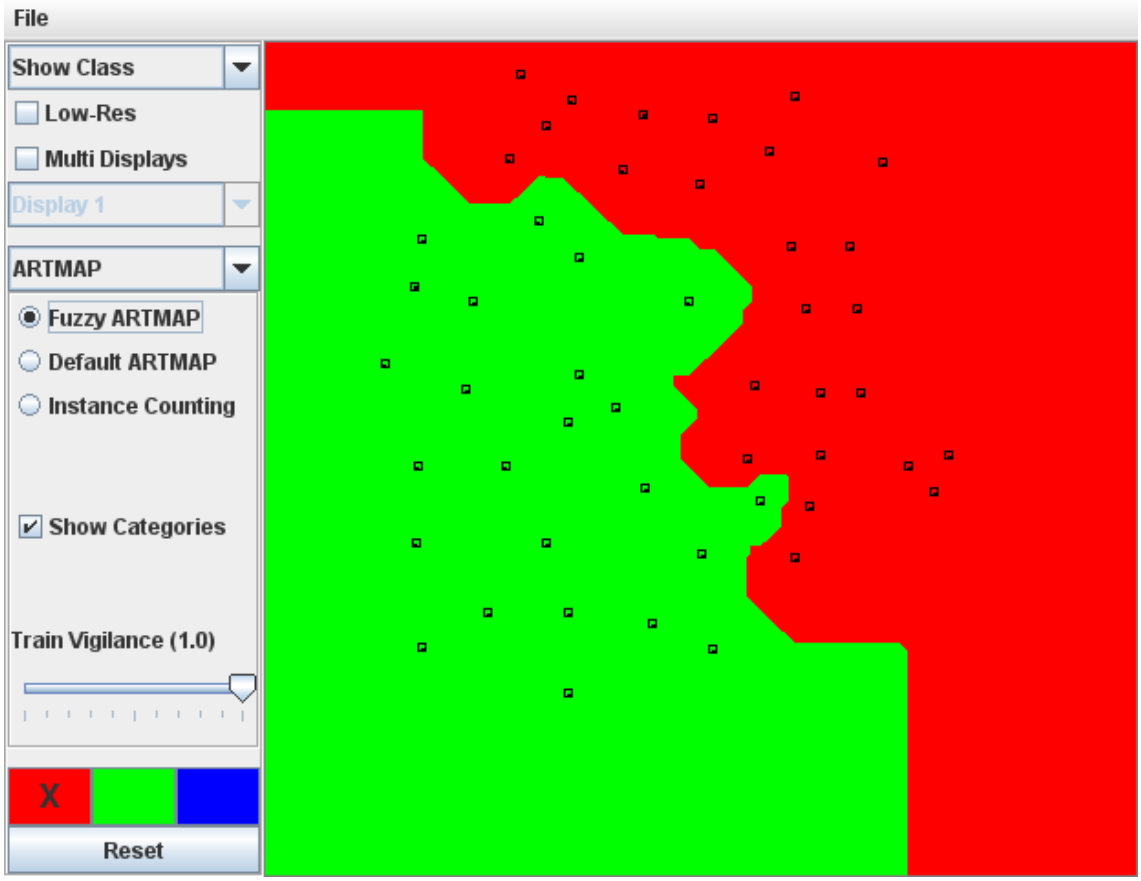

Figure 3.5 : Fuzzy ARTMAP Illustration with ρ = 1.0

For an illustration of ARTMAP with varying baseline train vigilance and network types, I created a toy dataset using the java applet provided by Boston University (http://techlab.bu.edu/classer/artmap_applet). The applet enables testing the spontaneously created dataset with k-NN along with Fuzzy, Default and Instance Counting (IC) ARTMAP. In k-NN, k value is allowed to manipulation in 1-9 discrete range, while for ARTMAP networks train vigilance could be set in 0-1 continuous range.

Illustrations in Figure 3.4 and 3.5 point out to the sharp difference in learning. In the toy dataset, there are two classes labeled with red and blue squares. While in Figure 3.4 all samples of both classes belong to the same categories respectively, in Figure 3.5 each sample represents a category. With 0.0 vigilance, we have a rough generalization,

whereas with 1.0 vigilance over-learning occurs with no generalization (also known as memorization).

Intermediate values show that baseline train vigilance can be tuned to overcome category clash without giving rise to over-learning. As in all ANNs tuning of this parameter depends on the dataset nature. Figure 3.6 illustrates learning with 0.5 and 0.75 vigilance values which seem better then the extremes.

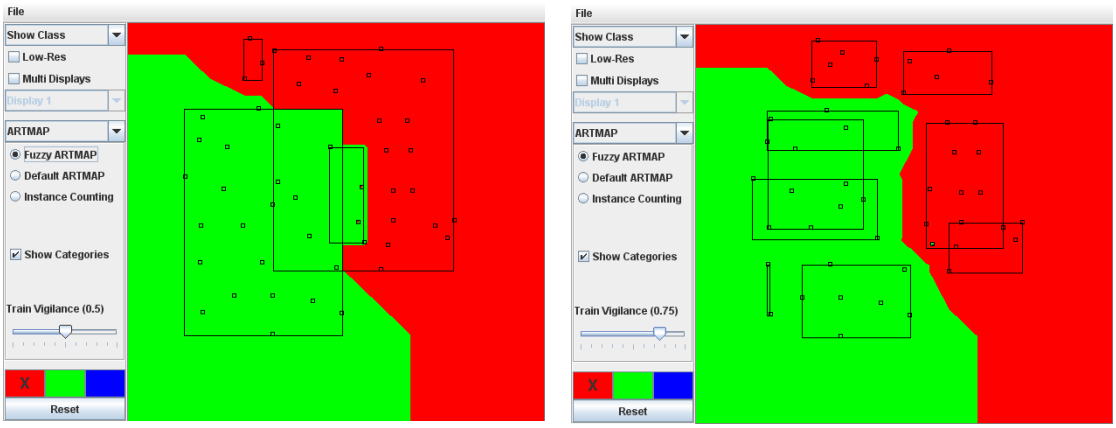

Figure 3.6 : Fuzzy ARTMAP Illustration with Intermediate ρ values

ARTMAP family has members apart from Fuzzy ARTMAP, namely, Default ARTMAP, Distributed ARTMAP and ARTMAP Instance Counting. In Fuzzy ARTMAP although the input is fuzzy, the output is not. This implies that there is no a fuzzy class membership function of a test pattern. However, other ARTMAP family members allow new input tuple to have a fuzzy membership. As it is seen in Figure 3.7, the network type does not affect the categorization of training samples, but the domain boundaries. It is also apparent that Default and IC network types do not significantly differ in their decisions.

Despite the fact that k-NN and ANNs are completely different in terms of algorithm, they are closely comparable in their decisions. For that reason ARTMAP applet implements a k-NN classifier.

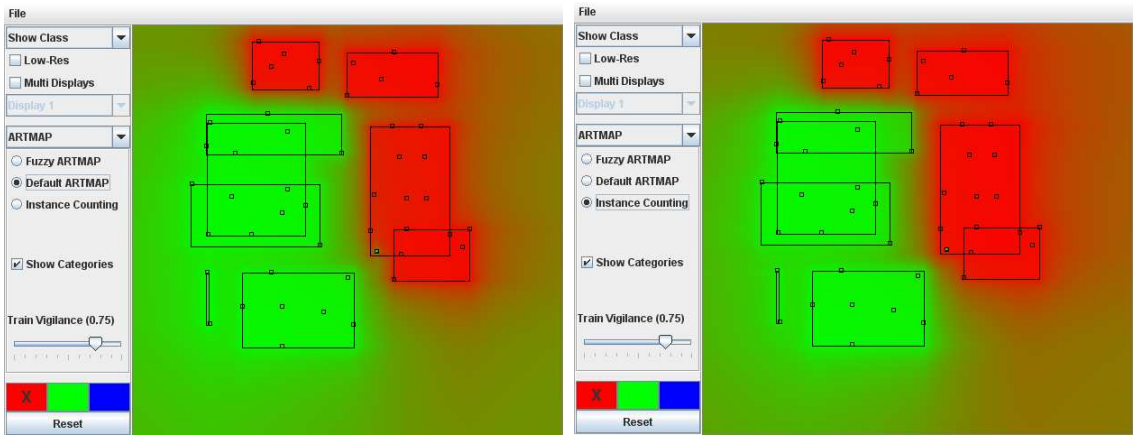

Figure 3.7: Illustrations in Default ARTMAP and ARTMAP IC with ρ= 0.75

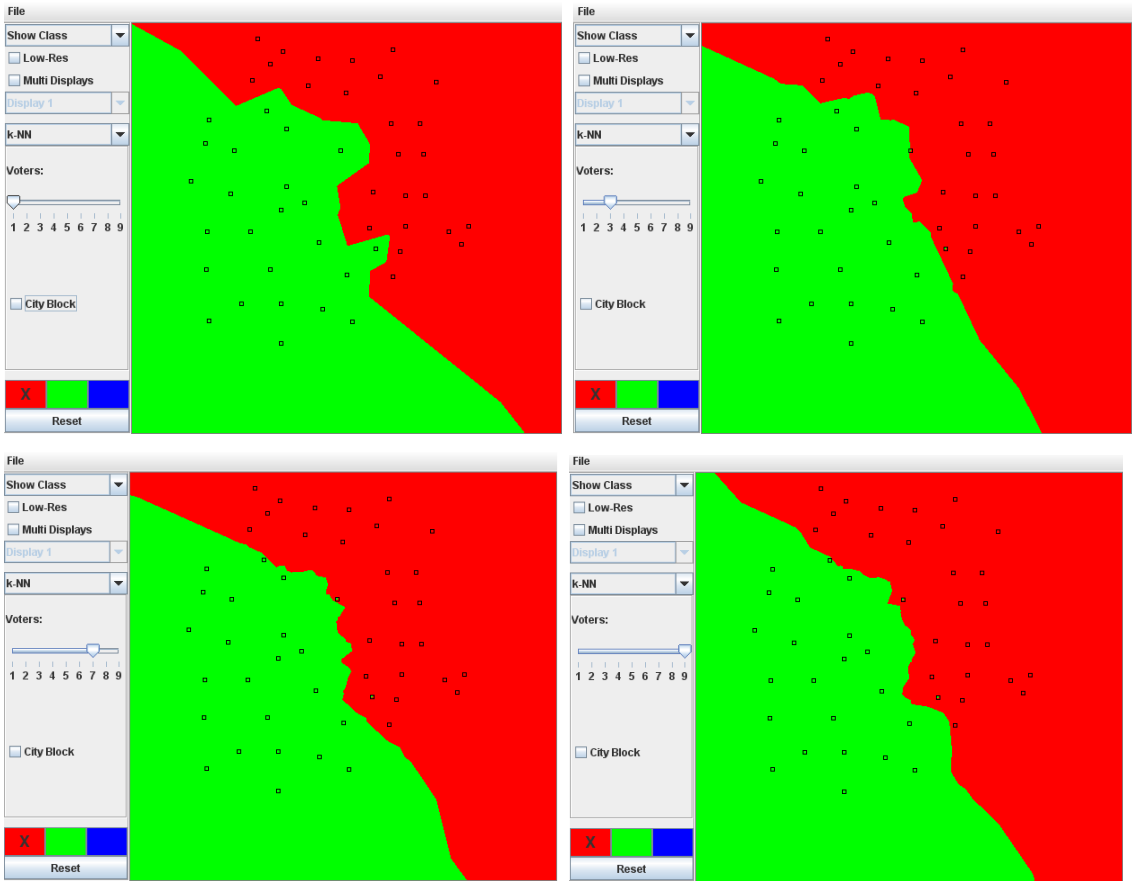

Figure 3.8: Illustrations in k-NN with varying k values (k = 1,3,7,9 as shown by the control-bar in the respective panels).

When k is set to 1 the output is very close to Fuzzy ARTMAP with baseline train vigilance = 1.0. In both cases over-learning is prominent. Higher k values yield smoother generalization as shown in Figure 3.8.

3.5. K-Medoids

K-medoids is a well known clustering method. It was preferred to k-means since it is more resistant to noise. The implemented algorithm is adapted from J. Han and M. Kamber's Data Mining book p. 435.

Algorithm: k-medoids. PAM, a k-medoids algorithm for partitioning based on medoid or central objects.
Input:
k: the number of clusters,
D: a data set containing n objects.
Output: A set of k clusters.
Method:
1. arbitrarily choose *k* objects in *D* as the initial representative objects or seeds;
2. **repeat**
3. assign each remaining object to the cluster with the nearest representative object;
4. randomly select a non-representative object, o_{random};
5. compute the minimal total cost, S, of swapping each representative object, o_j, with o_{random};
6. if S < 0 then swap o_j with o_{random} to form the new set of k representative objects;
7. **until** no change;

where

$$E=\sum_{j=1}^{k}\sum_{p \in C_j}|p-o_j| \quad (3.1)$$

$$S= E_{n+1}-E_n \quad (3.2)$$

Figure 3.9: PAM[1], the k-medoids partitioning algorithm used in the study

[1] In order to avoid misunderstanding, underlined words in the figure were added by Dr. Selim MİMAROĞLU in the Introduction to Data Mining course in Fall 2008 at Bahçeşehir University.

As stated earlier, k-M was exploited as a component of K-MNB method. The preprocessing task of k-M before NB classification is more than discretization. Features constituting one view are collapsed into one representative variable via k-M clustering. Thereof, this data-summary information (cluster index) is provided to NB. Unlike ARTMAP and k-NN, the selected views were not merged but treated separately.

Training set was clustered according to given algorithm. However, in order to avoid bias, test set was not involved in clustering. Samples of test set were assigned to closest medoids for each view.

3.6. Naïve Bayes

Another commonly used classifier is NB. Bayesian classifiers are based on Bayes' Theorem. The theory is attributed to 18^{th} century English clergyman Thomas Bayes. The *naïve* adjective comes from the assumption that attributes are independent. This property simplifies computations, leading to very fast outcomes. Despite its naïve nature NB is very accurate. Due to these advantages NB has a wide range of use, such as spam filtering.

3.6.1. Bayes Theorem

NB is a statistical classifier. It predicts membership probabilities.
X: an evidence, object
H: hypothesis, class
P(H | X): conditional probability (posterior probability H conditioned on X)
P(H): prior probability

$$P(H \mid X) = \frac{P(H, X)}{P(X)} \qquad 3.3$$

$$P(X \mid H) = \frac{P(X, H)}{P(H)} \quad 3.4$$

Therefore Bayes Theorem gives us the following simplified equation

$$P(H \mid X) = \frac{P(X \mid H)\, P(H)}{P(X)} \quad 3.5$$

3.6.2. Naïve Bayesian Classification

Jiawei and Kamber describe Naïve Bayes Classification as follows:

1. Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, $X = (x_1, x_2, \ldots, x_n)$, depicting n measurements made on the tuple from n attributes, respectively, A1, A2, … , An.

2. Suppose that there are m classes, $C_1, C_2, \ldots, C_m$. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the Naïve Bayesian classifier predicts that tuple X belongs to the class C_i if and only if

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m;\ j \neq i. \quad 3.6$$

Thus we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem,

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad 3.7$$

3. As P(X) is constant for all classes, only $P(X|C_i)P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is,

$$P(C_1) = P(C_2) = ... = P(C_m) \quad 3.8$$

and we would therefore maximize $P(X|C_i)$. Otherwise, we maximize $P(X|C_i)P(C_i)$. Note that the class prior probabilities may be estimated by

$$P(C_i)=\frac{|C_{i,D}|}{|D|} \quad 3.9$$

where $|C_{i,D}|$ is the number of training tuples of class C_i in D.

4. Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X|C_i)$. In order to reduce computation in evaluating $P(X|C_i)$, the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$P(\boldsymbol{X}|C_i) = P(x_1|C_i) \cdot P(x_2|C_i) \cdot \ldots \cdot P(x_n|C_i), \quad 3.10$$

and we can easily estimate the probabilities $P(x_1|C_i)$, $P(x_2|C_i)$, ... , $P(x_n|C_i)$ from the training tuples.

NB implementation is adapted from utility library[2] developed by Basu, Melville, and Mooney from University of Texas. In order to avoid zero conditional probability which could be caused by a missing feature value in the training set, Laplacian smoothing was applied. Laplacian smoothing is done via adding 1 (imaginary) sample for each possible value of corresponding feature. If we have *n* samples and *t* attributes for a feature *x* then prior probability for class *C* after Laplacian correction becomes

[2] Available: http://www.cs.utexas.edu/users/mooney/ir-course/

$$P(x \mid C) = \frac{n_i + 1}{n+t} \quad \text{for } 1 \leq i \leq t \qquad 3.11$$

3.7. mRMR (maximum Relevance Minimum Redundancy)

mRMR is a feature ranking method which suggests incrementally selecting the maximally relevant variables while avoiding the redundant ones with the aim of selecting a minimal subset of variables that represents the problem (Peng et al., 2005 Sakar, 2008). mRMR ranking, decides to include m^{th} feature/view into the selected set, *S*, upon satisfaction of the following condition:

$$\max_{x_j \in X - S_{m-1}} \left[I(x_j, c) - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} I(x_j; x_i) \right], \qquad 3.12$$

which means the feature having maximum difference between its mutual information (MI) with the target variable and its MI with the already selected set is to be selected.

3.8. K-MNB

K-MNB is a setting consisting of k-Medoid clustering and Naïve-Bayes classification. The two components are commonly used. This setting is akin to RBF ANNs in principle. Radial Basis Function (RBF) Networks are universal approximators of any continuous functions in regression and classification (Skomorokhov, 2002). RBFs are usually chosen as radially-symmetric functions with a single maximum at the origin (Berthold & Hand, 1999). They tend to approximate multivariate data muting the effects of noise. We know that k-Medoids is a centroid based clustering algorithm which is resistant to noise in nature. Therefore in both ways k-Medoids extraction as a preprocessing for Naïve Bayes is akin to RBF Networks.

In short, K-MNB first divides the samples into sites, and then conquers these sites over representatives. The clusters shaped around the representatives are naïvely associated with labels. Prior statistics of those associations elicit posterior predictions.

In this setting, each view of training samples was subjected to clustering separately. Then test samples of corresponding view are associated with the closest medoid. After this extraction process, nominal view values (cluster indices) were used for NB classification.

3.9. K-MART

K-MART is a learning network arranged by stacking class probabilities obtained from view based K-Medoids clustering to Fuzzy ARTMAP. In this setting, each view was compressed by an individual K-M clustering. Later, class distributions within each cluster were calculated for each view based clustering. The process of extracting class distributions from K-M clustering was repeated many times (with randomly selected k value at each) the average result of which process is thought to increase reliability. The value range of k parameter was dependent on the number of classes in the corresponding dataset. As a final step, the average distributions were fused for Fuzzy ARTMAP stacking.

For an illustration, suppose that we have 7 views with a total of 21 variables. Also suppose that the dataset has 2 classes. In fact, this is the case with Tele-monitoring of Parkinson's disease dataset. If we compress these views using k-M clustering, we will have 7 clusterings. Since we had 2 classes the k parameter value of k-M will be selected from 10-59 range (in case range shift coefficient is 5, and range width is 50). For each view the class distribution of clusters are calculated. Cluster information is replaced by this 2 (number of class labels) dimensional distribution information. Since a distribution is a ratio in 0-1 range, it does not require further normalization before stacking to Fuzzy ARTMAP ANN. The clustering and distribution calculation is repeated sufficiently many times (studies revealed that there is no significant difference between 50 and 100 repetitions). The average distribution information is calculated as the last process before

classification. At this point we have $7 \times 2 = 14$ variables, where 7 is the number of views and 2 is the number of class labels.

Algorithm: K-MART, a method for stacking class posterior probabilities obtained from view-based k-Medoids clustering to Fuzzy ARTMAP

Input:
nV: the number of views,
nL: the number of classes (labels),
rW: the range width value for k param of K-M
rS : the range shift coefficient for k param of K-M
nC: the number of clusterings for each view,
D: the dataset arranged as combination of views,
trE: the row index to mark end of training dataset within D

Output: Test dataset classification results

Method:

```
1. double[][] trns_ds = double[D.length][nV*nL]; //transformed dataset
2. for i=0 to nC-1
3    k= rand(rW)+nL*rS;
4.   for v=0 to nV
5.        clustering= K-Medoids.find_Clusters(k,D,v,trE);
6.        double[][]                                    distrib= get_class_distrib(clustering,D.labels,trE);
7.        for t=0 to D.length-1
8.             for c=0 in nL-1
9.                  trns_ds[t][nV*v+c]+= distrib[t][c];
10.            end
11.       end
12.  end
13. end
14. calc_avg(trns_ds,nC);
15. train and test transformed dataset in Fuzzy ARTMAP;
16. return test results;
```

Figure 3.10: K-MART, a method for stacking k-M clustering to Fuzzy ARTMAP

It is important to note that since we have 2 class labels class distribution data is going to be complement of the other. Since Fuzzy ARTMAP ANN has internal complement coding this information becomes redundant. So for a dataset with 2 class labels and K-MART specific setting, the information could contain only one class label information.

Therefore in Parkinson's dataset, the dimensionality of extracted feature vector becomes 7. The tests proved that the performance of two alternatives is the same. The algorithm used in the study is given in Figure 3.10.

4. EXPERIMENTAL RESULTS

Due their appropriate nature for multi-view pattern recognition, two biomedical datasets were used in the study. One of them is known as Parkinson Dataset and the other is a recently collected Protein Dataset.

Both datasets were normalized at preprocessing stage. One of the reasons for such process was that Fuzzy ARTMAP requires analog data to be in 0-1 range. Besides, normalization was expected to provide more reliable calculations in EUCLIDIAN distance metric used in the study.

All algorithms were implemented in Java. Memory allocation provided by Eclipse IDE was sufficient for Parkinson Dataset. However, since views of Protein Dataset were very high dimensional (especially view number 2 with 400 dimensions), running ARTMAP ANN, which creates multitude of categories exploiting heap memory, gave "java.lang.OutOfMemoryError: Java heap space" exception. In order to be fair in testing, all methods used for Protein Dataset were run with 1024 MB memory allocation (using –Xmx Java option) which was sufficient for ARTMAP.

4.1 PARKINSON DATASET

Parkinson's disease (PD) is a serious neural disorder. Recently a study conducted by Little et al. (2008) revealed the relationship between vocal signals and PD. The corresponding dataset is available at UCI Machine Learning Repository 2008 Archive[3]. It was created by Max Little of the University of Oxford, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, who recorded the speech signals. The original study published the feature extraction methods for general voice disorders. Dataset is multivariate and features have natural groups. Therefore, it was feasible to apply view-based machine learning processes on the dataset.

[3] Available at: http://archive.ics.uci.edu/ml/datasets/Parkinsons (May,2009)

4.1.1 Dataset Description

Parkinson dataset is composed of a range of biomedical voice measurements from 32 people, 24 with Parkinson's disease. Each feature is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals (6-7 recordings per individual). The main aim of the data is to discriminate healthy people from those with PD, according to their class label which is set to 0 for healthy and 1 for PD.

Features have a natural grouping under 7 categories.

ViewID	Description	Feature Label	FEATURE
0	Basic vocal fundamental freq. statistics	MDVP:Fo(Hz)	**1**
		MDVP:Fhi(Hz)	**2**
		MDVP:Flo(Hz)	**3**
1	Several measures of variation in fundamental frequency	MDVP:Jitter(%)	**4**
		MDVP:Jitter(Abs)	**5**
		MDVP:RAP	**6**
		MDVP:PPQ	**7**
		Jitter:DDP	**8**
2	Several measures of variation in amplitude	MDVP:Shimmer	**9**
		MDVP:Shimmer(dB)	**10**
		Shimmer:APQ3	**11**
		Shimmer:APQ5	**12**
		MDVP:APQ	**13**
		Shimmer:DDA	**14**
3	Two measures of ratio of noise to tonal components in the voice	NHR	**15**
		HNR	**16**
4	Two nonlinear dynamical complexity measures	RPDE	**17**
		D2	**18**
5	Signal fractal scaling exponent	DFA	**19**
6	Three nonlinear measures of fundamental frequency variation (Last one, PPE, is the proposed measurement of dysphonia by Little et al.)	Spread1	**20**
		Spread2	**21**
		PPE	**22**

Table 4.1: Parkinson Dataset Feature Descriptions

4.1.1.1 How to Avoiding Misinterpretation of Data

Each client, no matter whether he/she is healthy or sick, has a vocal pattern. Any pattern recognition algorithm can more easily match those patterns in speech recordings of the client, if any other speech recording of that same client is already available in the training set. Therefore one should be careful when using leave-one-out testing to avoid bias. In other words, leaving *one record* out is not sufficient for fair testing. Any client should totally be in or totally out which implies that all recordings for a client can either be in training set or in test set. This issue was not realized in the work of Little et al. but pointed out by Sakar and Kursun (2009).

4.1.2 View Ranking/Selection

Due to fact that the data is not highly dimensional as it will be in the protein dataset, ranking views through mRMR was not found so feasible. However, view ranking was done via ARTMAP ANN classifiers where it is appropriate. For those ranked views, RS technique was applied as it is described in section 2. Forward Selection (FS) was applied in all classification methods.

4.1.3 Quantitative Comparison using K-MNB

Composed by k-Medoids and Naïve Bayes, K-MNB has stochastic nature. Hence, tests for each technique and k value consisted 10 runs. Average values are considered for comparison. Variance found to be less than or equal to %0.1.

For this dataset, K-MNB was run without view selection (using all views) and with feature selection. The results obtained with varying k values are given below.

	% Success over k												
Method / k	3	4	5	6	7	8	9	10	11	12	13	14	15
Forward Selection	78.9	81.2	80.6	82.7	82.8	83.2	83.6	84.9	83.6	82.2	83.8	83.1	82.4
All Views	74.7	75.8	77.5	76.9	79.1	78.9	78.2	77.0	78.9	76.9	78.1	78.0	77.6

Table 4.2: Average K-MNB Results for Varying k Values

As it can be read in Table 4.2 and clearly seen on the Figure 4.1, the views selected using FS technique show significant difference from unselected set of views. The sharpest difference was observed at k=10 (%7.9), the average difference was 5.0%.

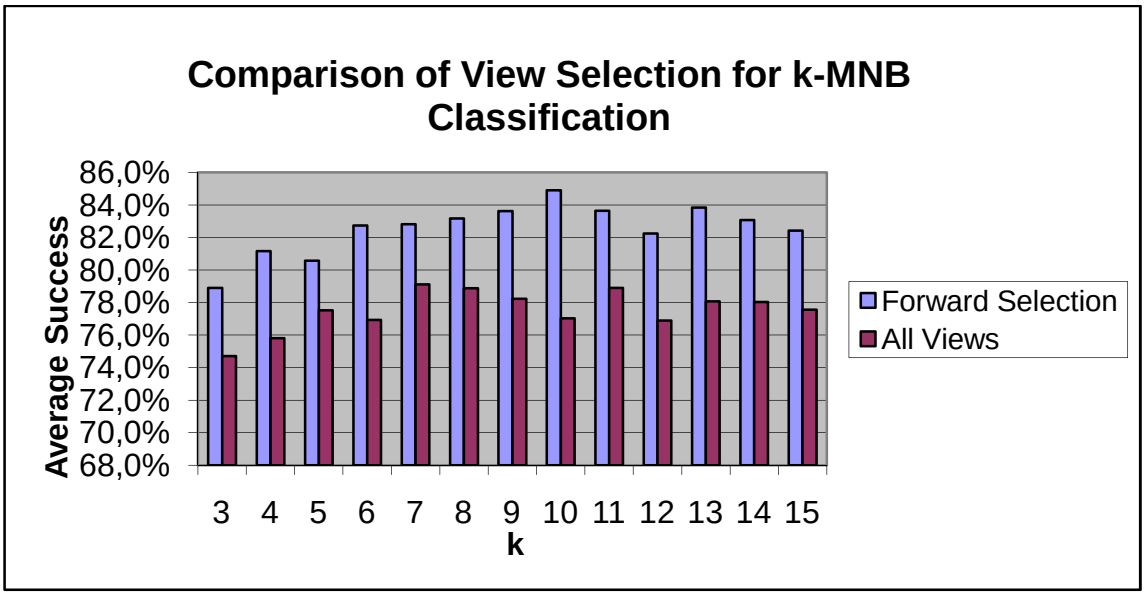

Figure 4.1: Comparison Graph of Average K-MNB Results for Varying k Values

Since the output list is long and tiring, runs for two k values were selected.

	k=4		k=10	
Run	**Selected Views**	**Pred. Success**	**Selected Views**	**Pred. Success**
1	[0, 1]	82.26%	[3, 6]	82.04%
2	[6, 1, 3]	87.48%	[0]	81.60%
3	[2]	77.42%	[0, 1, 6, 4]	89.90%
4	[2, 4]	78.39%	[0, 6, 4]	86.90%
5	[0, 1]	80.56%	[0, 4, 6, 1]	85.95%
6	[6, 4, 5, 0]	84.18%	[0, 6, 5]	82.62%
7	[6, 0, 4]	82.24%	[0]	82.09%
8	[4, 3]	79.43%	[1, 6, 0]	86.50%
9	[2]	77.42%	[6, 0, 1, 5]	87.09%
10	[2, 0, 4, 6]	82.21%	[0, 6, 4]	84.37%
Average		**81.16%**		**84.91%**
Variance		**0.10%**		**0.08%**

Table 4.3: Sample Summary Results of K-MNB Forward Selection

Run	k: 4	k=10
1	74.03%	74.11%
2	75.90%	76.94%
3	74.32%	80.68%
4	75.76%	79.08%
5	71.94%	79.63%
6	76.80%	74.07%
7	85.21%	77.91%
8	76.25%	76.62%
9	69.42%	74.34%
10	78.43%	76.88%
Average	75.81%	77.03%
Variance	0.18%	0.06%

Table 4.4: Sample Summary Results of K-MNB with All Views

Views Loaded	Success
[6]	77.5%
[5]	77.4%
[4]	77.4%
[2]	77.4%
[0]	77.4%
[1]	77.4%
[3]	77.4%
[6, 5]	77.0%
[6, 4]	76.4%
[6, 2]	75.3%
[6, 0]	76.5%
[6, 1]	79.0%
[6, 3]	70.8%
[6, 1, 5]	76.8%
[6, 1, 4]	80.0%
[6, 1, 2]	73.3%
[6, 1, 0]	82.6%
[6, 1, 3]	87.5%
[6, 1, 3, 5]	72.7%
[6, 1, 3, 4]	79.9%
[6, 1, 3, 2]	70.2%
[6, 1, 3, 0]	76.8%
Selected Views [6, 1, 3]	**87.5%**

Table 4.5: Sample Forward Selection run of K-MNB (k=4, 2nd run)

For further analysis of derivation mechanism, the sample run given in Table 4.4 can be traced. Note that Table 4.4 shows one run for corresponding k value out of 10 runs whose average was taken as performance index.

4.1.4 Quantitative Comparison using k-Nearest Neighbors Classification

In addition to FS technique, ARTMAP based RS was also used in k-NN testing.
Both selection techniques excelled the set of unselected views. The ARTMAP RS technique performed equal to or below FS. However FS has much higher computational complexity compared to RS.

Method	**% Success over k**						
	1	3	5	7	9	11	13
Forward Selection	80.8	81.2	82.3	81.3	82.3	82.8	82.8
ARTMAP RS	80.8	79.2	79.7	79.8	82.3	81.2	81.3
All Views	77.1	76.1	76.0	75.0	74.5	74.5	76.0

Table 4.6: k-NN Results for Varying k Values

Figure 4.2 depicts Table 4.6. Apparently, selected views outperformed the total dataset (used as single view). The best technique was found to be FS. On the other hand ARTMAPT RS performance was not found to be significantly lower than FS performance.

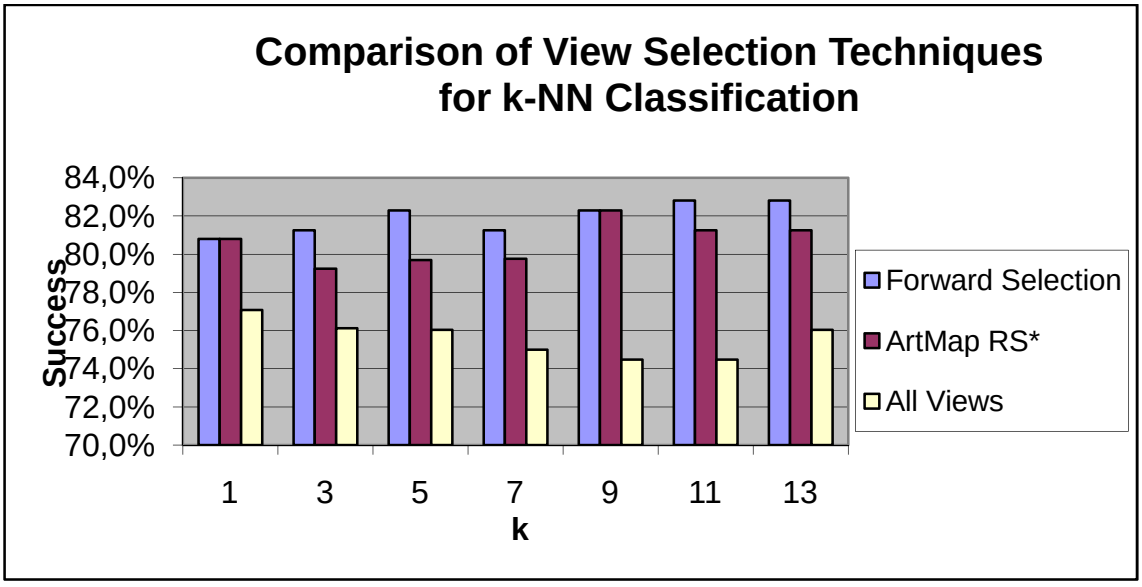

Figure 4.2: Comparison of View Selection Techniques for k-NN

Further insight can be gained through tracing view selection mechanism given in Table 4.7. Note that the order or views is not important because algorithm always selects the view with best individual performance first.

Views Loaded	Success
[6]	80.2%
[0]	75.2%
[1]	79.2%
[5]	61.5%
[4]	71.4%
[2]	63.7%
[3]	63.2%
[6, 0]	80.2%
[6, 1]	79.2%
[6, 5]	82.8%
[6, 4]	80.7%
[6, 2]	81.2%
[6, 3]	77.7%
[6, 5, 0]	78.1%
[6, 5, 1]	81.8%
[6, 5, 4]	81.8%
[6, 5, 2]	81.3%
[6, 5, 3]	81.8%
Selected Views [6, 5]	**82.8%**

Table 4.7: Sample Forward Selection run of k-NN (k=11)

In the first iteration view 6 (Spread1, Spread2 and PPE) is selected. The next iteration calculated the combined performance of remaining views with view 6. View 5 (DFA) was selected at this step. Since there was no improvement in the subsequent iteration, the algorithm terminated selecting views 6 and 5.

4.1.5 Quantitative Comparison using Fuzzy ARTMAP Classification

Similar to k-NN setting, ARTMAP classification was carried out using FS and ARTMAP RS techniques. Prediction success of all views was used here for comparison, too.

A preliminary study revealed that prediction success varies due to train vigilance (ρ) resulting in a fluctuating graph. On the other hand, there was a gradual increase in success as ρ increases. Hence, thesis study did not involve testing with small increments of ρ. Testing was carried out using four different ρ values.

Results obtained from ARTMAP classification were very similar to those of k-NN. In both methods ARTMAP RS performed much better than unselected set of views. In all methods, (namely K-MNB, k-NN, and ARTMAP) FS was the best technique. If maximum performance attained in all three methods were to be compared, the descending ordering is K-MNB, ARTMAP and k-NN with 1% difference between successive methods (84.9; 83.9; 82.8). Classification results of ARTMAP are shown in Table 4.8 and depicted in the following Figure 4.3 for ease of analysis.

Method / ρ	0.25	0.5	0.75	0.99
Forward Selection	71.0%	78.6%	83.9%	82.3%
ARTMAP RS	71.0%	74.3%	79.2%	82.3%
All Views	58.6%	68.9%	73.7%	74.0%

Table 4.8: Comparison of Selection Methods for ARTMAP Classification

As it can be seen in Table 4.8 and Figure 4.3, the views proposed by both selection methods were able to significantly outperform the total dataset. Two selection methods performed equal at 0.25 and 0.99 baseline vigilance values, on the other hand at ρ =0.5 and ρ=0.75 FS prediction success was also significantly higher than ARTMAP RS.

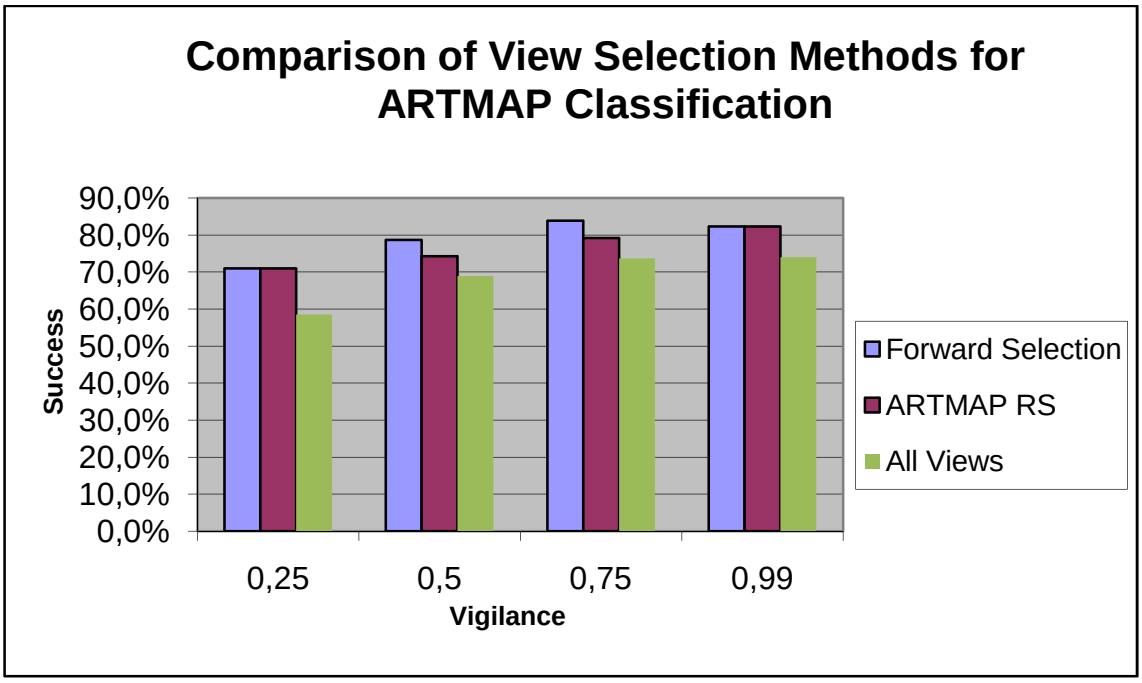

Figure 4.3: Comparison of View Selection Methods for ARTMAP Classification

4.1.6 Further Investigation using K-MART

Relative success of simple setting K-MNB led to further analysis of class distribution of view based clusters. Thus, several tests using raw data, class distribution probabilities and fusing of both raw data and class probabilities were carried out using Fuzzy ARTMAP neural network. K-MART denotes Fuzzy ARTMAP stacking of cluster class posterior probabilities which were averaged from 50 runs using randomly selected k values from 10-59 range.

Contrary to the argument stated in Section 4.1.1.1, the Parkinson dataset was handled similar to Little at al. (2008) for comparability purposes. The dataset was randomly shuffled disregarding client feature. Then it was split half and one fold was trained and the other was used for testing. It can be viewed as a 2-fold cross-validation setting. The tests were carried out with baseline vigilance value of 0.99.

Fold/Method	All Views	K-MART
1	91.75%	95.88%
2	84.69%	81.63%
Average	88.22%	88.75%

Table 4.9: Comparison of Fuzzy ARTMAP and K-MART

As it can be observed in Table 4.9, K-MART slightly increased the prediction rate of Fuzzy ARTMAP with raw features. The high prediction rate attained in Fold 1 by K-MART could be attributed to this method however one should also consider the testing bias which is explained in Section 4.1.1.1.

4.2 PROTEIN DATASET

One of the most commonly studied subjects in bioinformatics is in fact Protein fold classification. Due to the fact that proteins are produced over genetic coding and are responsible for controlling vital functions, protein folds have great interest. Understanding protein structures will also lead to appropriate drug production hence better treatment of diseases.

A recent study to form a highly dimensional Protein Dataset was carried out by Nanuwa and Seker (2008) in De Montfort University, Leicester. Uneven distribution of class samples combined with multivariate and under-sampled data posed a great challenge.

4.2.1 Dataset Description

Dataset is composed of 714 samples having 1497 features categorized in 53 views. Each sample had a label out of 9 available classes. No missing values existed in the dataset. However the distribution was so uneven that while the most frequent class had 307 samples in total, the least frequent class had only 13 samples. Though this case worsens pattern recognition/data mining studies, it is very common in bioinformatics.

Split half method was used to train and test samples. Training set was favored when a class has odd number of samples. Class descriptions and their corresponding sample distribution are shown in Table 4.10.

ClassID	Description	Training	Test	Total
1	Chromatin proteins	50	49	**99**
2	Heterochromatin proteins	11	11	**22**
3	Nuclear Envelope proteins	31	30	**61**
4	Nuclear Matrix proteins	15	14	**29**
5	Nuclear Pore Complex proteins	40	39	**79**
6	Nuclear Speckle proteins	34	33	**67**
7	Nucleolus proteins	154	153	**307**
8	Nucleoplasm proteins	19	18	**37**
9	Nuclear PML Body proteins	7	6	**13**
Total		**361**	**353**	**714**

Table 4.10: Class Distribution of Protein Dataset

Views of this dataset are sub-categories of the following main set of views: (1)Amino acid composition, (2) Dipeptide composition, (3) Normalized moreau-borto correlation, (4) Moran autocorrelation, (5) Geary autocorrelation, (6) Composition, Transition & Distribution, (7) Sequence Order and (8) Pseudo amino acid composition. Compared to multi-view protein fold recognition study or Okun and Priisalu (2005), this set contains almost the same number of samples however the number of inherent views is almost 9 times. Pointing out to the importance of proper view-selection, they utilized a selection algorithm based on cross-validation errors instead of random selection and validation error based selection. They have calculated the success of k-NN ensemble success over test errors, while this study dealt directly with pattern prediction success. On the hand, the research group, where this dataset is originated, reported that the best prediction success achieved using 10-fold cross validation was around 65%.

4.2.2 View Ranking/Selection

Data is challenged by three ranking methods mRMR, ARTMAP ANN and k-NN for RS technique as well as FS which works with self feedback. In order to increase prediction accuracy, combinations of ranking methods were used after certain reductions of views which decrease the cumulative performance.

4.2.3 Quantitative Comparison using K-MNB

Since the dominant process is feature extraction rather than selection, view-selection was not intended in K-MNB. Each view was represented with the extracted cluster indexes. However, in order to gain understanding about behavior of view combinations, FS and RS techniques were used for specific values of k.

Similar to results obtained with Parkinson dataset, prediction success gradually increased and then started to decrease after a certain value of k. Therefore it was

possible to find an optimum value for k. In Parkinson Dataset there were only two classes and a total of 7 views, so incrementing k with 1 was viable. However Protein Dataset contains 9 classes and 53 views, therefore incrementing was decided to be 10.

Prediction success results of all views with varying k are shown in Table 4.11. Recall that success of each k value was calculated over 10 runs.

K	10	20	30	40	50	60	70	80	90	100
Max	43.98%	48.74%	53.22%	52.66%	55.46%	55.18%	57.22%	56.09%	55.81%	56.37%
Average	43.03%	46.89%	51.62%	51.48%	52.72%	53.28%	55.16%	54.48%	53.31%	53.97%
Variance	0.02%	0.02%	0.02%	0.01%	0.02%	0.03%	0.01%	0.01%	0.01%	0.03%

Table 4.11: Prediction Success of K-MNB with Varying k

On the hand, FS did not perform well in this dataset as it did in Parkinson's. FS performance fell below performance of the set of unselected views with increasing k. In fact, as we shall see later in this section, no selection technique using K-MNB method performed better than the set of all views for Protein Dataset. In Table 4.12 average prediction performances of FS and All Views could be compared.

Method / k	10	20	30	40	50	60	70
FS	47.05%	47.11%	46.71%	46.32%	47.99%	47.45%	48.36%
All Views	43.03%	46.89%	51.62%	51.48%	52.72%	53.28%	55.16%

Table 4.12: Comparison of Average Prediction Success of FS and All Views

Studies concerning mRMR ranking with K-MNB revealed that the number of views loaded has positive correlation with performance. This steady increase implies that all possible view combinations would fell below All Views performance. Preliminary work on dataset (where 500 samples were used for training and remaining 214 samples were

used for testing) using mRMR RS is shown in Table 4.13. It seemed as if K-MNB was expecting more views to perform better.

load / k	30	70
10	44.81%	49.91%
20	47.90%	53.69%
30	48.60%	55.09%
40	49.72%	56.54%
53	53.83%	56.68%

Table 4.13: Average Success Results of K-MNB mRMR RS

Parallel results were found in ARTMAP RS as well. Results shown in Table 4.14 were gathered after samples were split half.

load / k	70
10	48.22%
20	51.27%
30	52.89%
40	53.09%
53	55.16%

Table 4.14: Average Success Results of K-MNB ARTMAP RS

In this dataset, K-MNB behaved quite different than k-NN and ARTMAP in terms of view selection performance. While, k-NN and ARTMAP increased All Views performance around %5 using selection techniques, K-MNB did not. However, best prediction success of K-MNB, even with All Views, was better than the best scores in other methods.

4.2.4 Quantitative Comparison using k-Nearest Neighbors Classification

Several RS techniques applied using k-NN for Protein Dataset. k-NN, mRMR and ARTMAP RS made significant difference over All Views. However, no RS technique was found to outperform FS. It seemed that k-NN worked best at k=1 for FS. FS success was 3-7% higher than All Views success, which validates the fact that reducing learning complexity and selecting appropriate views increase pattern recognition performance. Comparative FS success can be analyzed using Table 4.15 and 4.16.

Method / k	1	3	5	7	9	11
All Views	48.44%	48.44%	47.59%	47.03%	47.88%	49.29%
FS	55.52%	53.26%	53.26%	54.39%	50.99%	53.54%

Table 4.15: Comparison of FS with All Views for k-NN Classification

As Table 4.15 shows success rates with corresponding k values, Table 4.16 additionally lists the selected views which could be used for further data mining processes.

k	Selected Views using Forward Selection	Success
1	[51, 42, 37, 0, 33]	55.52%
3	[51, 45, 35, 8]	53.26%
5	[51, 45, 30, 33]	53.26%
7	[0, 48, 33, 30, 3, 32]	54.39%
9	[51, 45, 30]	50.99%
11	[51, 16, 3, 4, 14, 36]	53.54%

Table 4.16: Set of Selected Groups using FS with Varying k for k-NN

Using k-NN for ranking, a success of 52.41% was attained with k=7, selected view set ={52, 1, 45, 49, 51, 42, 47, 44, 25, 41, 19, 11, 14, 50, 53, 6, 7, 16, 21, 17, 23}. Using ARTMAP ranking with ρ =0.75, k-NN classification success was 50.42% with the k=7 and selected view set ={38, 34, 2, 3, 1, 52, 40, 36, 13, 47, 37, 48, 49, 22, 10, 21, 43, 45, 46, 41, 50, 4 }. Without changing k parameter for k-NN classification, using mRMR for

ranking resulted in a success ratio of 48.44% with selected view set = { 2, 52, 15, 32, 41, 35, 33, 20, 43, 13, 4, 31, 29, 22, 49, 39, 23, 30, 38, 26, 27, 44, 53, 11, 1, 47, 24, 7, 37, 5, 40, 28, 10, 8, 12, 9, 46, 18, 45, 19, 16, 3, 48}. As stated in section 2 these were the best results obtained incrementally loading ranked views. Therefore mRMR RS was found to be well behind other selection techniques giving less than 1.5% difference over All Views.

4.2.5 Quantitative Comparison using ARTMAP Classification

Performing forward selection in ARTMAP with varying vigilance values, the best pattern recognition success for the protein dataset was attained (at ρ=0.99). Not surprisingly, performance was found to increase hand in hand with ρ. This was also the case for the parkinson's dataset. The other important but familiar fact was that FS performed better than newly introduced RS methods. FS performance compared with All Views is shown in Table 4.17. Selected views corresponding varying vigilance values are designated in Table 4.18.

Method / ρ	**0.25**	**0.50**	**0.75**	**0.99**
All Views	41.08%	44.76%	48.16%	52.12%
FS	45.61%	51.84%	53.54%	58.07%

Table 4.17: Comparison of FS with All Views for ARTMAP Classification

ρ	**FS Selected Groups**	**Success**
0,25	[33, 26]	45.61%
0,50	[0, 2, 21]	51.84%
0,75	[51, 31]	53.54%
0,99	[51, 40, 50, 30, 33]	58.07%

Table 4.18: Set of Selected Groups using FS for ARTMAP with Varying ρ

mRMR ranking for ARTMAP classification performed better than it was for k-NN. mRMR RS led to a success of 53.54% with ρ =0.75 and selected view set = { 2,52,15,32,41,35,33,20,43,13,4,31,29,22,49 }. This was equal to success of FS at the same vigilance. On the other hand, ARTMAP ranking (for its own classification) reached a success of 51.84% in the presence of same vigilance value and selected view set of { 39,35,3,4,2,53,41,37,14,48,38,49,50,23,11,22,44,46 }.

4.2.6 Further Investigation using K-MART

Parallel to the K-MART study with Parkinson dataset, a 2-fold cross-validation setting was prepared. The folds were training and test datasets used in previous studies with the same dataset.

Test results, as designated in Table 4.19, show that K-MART method performs significantly better than simple use of Fuzzy ARTMAP in highly dimensional Protein dataset. Here simple use refers to utilization of the raw features without any selection and/or extraction process.

Fold/Method	All Views	K-MART
1	52.12%	57.22%
2	57.06%	60.94%
Average	54.59%	59.08%

Table 4.19: Comparison of Fuzzy ARTMAP and K-MART

K-MART results were the best attained with Protein dataset. All other methods used in the study that do not exploit view selection were significantly below K-MART performance. Extended studies using SVM yielded an average classification accuracy of 44%. Therefore detailed SVM results were not included in this paper.

5. CONCLUSIONS

In this thesis, two biomedical datasets were classified using different multi-view methods for reducing the learning complexity and better accuracy was achieved than of the single-view methods. Mainly, I tried two stacking settings in this study that I named K-MNB and K-MART, respectively, which both utilized a simple form of feature-extraction via centroid based clustering method, K-M (K-Medoids). K-MNB uses NB (Naïve Bayes) and K-MART uses ART (Adaptive Resonance Theory) for fusion (stacking network) of K-M outputs. More specifically, these two methods fuse the class-posterior probabilities in clusters outputed by simple K-M method, where K-M clusterings are setup independently for each view. I have shown that such fusion of clustering outputs works better than not only single view classification by merging all the variables of all views together but also single view classification by merging a few of individually most potent views. In other words, K-MNB and K-MART techniques compare favorably to the single-view methods that choose and merge the most useful views into a single feature vector. Such a selected-view fusion (for the selection process, I used variants of Ranking and Sequential Forward Selection techniques), as expected, worked better than the whole set of views merged together; however, could not surpass the multi-view extension. The results also imply that K-MART, being a more sophisticated stacking network, is significantly better than K-MNB; and more generally, even a simple within-view clustering to extract class-posterior-probability-features from each view helps obtain better predictions than single-view methods.

REFERENCES

Books

Alpaydın E., 2004. *Introduction to Machine Learning*. MIT Press, Cambridge, MA.

Berthold M. & Hand D.J. (Ed.), 1999. *Intelligent data analysis: an introduction.* Springer-Verlag, Berlin.

Bishop, C.M., 1995. *Neural Networks for Pattern Recognition,* Oxford University Press, Oxford.

Hastie T., Tibshirani R., and Friedman J. H., 2001. *The Elements of Statistical Learning : Data mining, Inference, and Prediction : with 200 Full-Color Illustrations,* Springer, New York .

Jiawei H. & Kamber M., 2006. *Data Mining: Concepts and Techniques,* Morgan Kaufmann Publishers, San Francisco pp. 310-312, 406.

Periodical Publications

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., & Rosen, D.B., 1992. Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. *IEEE Transactions on Neural Networks*. Volume 3, Issue 5, pp. 698-713.

Carpenter, G.A., Grossberg S. & Reynolds J.H., 1991. ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by a Self-Organizing Neural Network. *Neural Networks*, vol. 4, pp. 565-588

Carpenter, G.A., Grossberg S. & Rosen D.B., 1991. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, vol. 4, pp. 759-771.

Dietterich T. G., 1997. Machine learning research: Four current directions. *AI Magazine*, 18(4), pp. 97-136.

Guyon I. and Elisseeff A., 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182.

Kurşun, O. & Favorov, O., 2004. SINBAD Automation of Scientific Discovery: From Factor Analysis to Theory Synthesis. *Natural Computing*, vol. 3, pp. 207-233.

Little, M.A., McSharry, P.E., Hunter, E.J., Ramig, L.O., 2008. 'Suitability of dysphonia measurements for telemonitoring of Parkinson's disease'. *IEEE Transactions on Biomedical Engineering*, 56(4), pp. 1015-1022.

Peng, H., Long, F., Ding, C., 2005. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8), pp. 1226-1238.

Sakar, C.O., Kursun, O., 2009. Telediagnosis of Parkinson’s Disease Using Measurements of Dysphonia, *Journal of Medical Systems* (to appear).

Other Publications

Bickel, S., & Scheffer, T., 2004. Multi-view clustering. *Proceedings of the Forth IEEE International Conference on Data Mining*. Brighton, UK, pp. 19–26.

Blum, A., & Mitchell, T., 1998. Combining labeled and unlabeled data with co-training. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, Madisson, WI, pp. 92–100.

Busque, M. & Prizeau, M., 1997. A Comparison of Fuzzy ARTMAP and Multilayer Perceptron for Handwritten Digit Recognition. Université Laval. Sainte-Foy (Quebec), Canada.
Available: http://w3.gel.ulaval.ca/~mbusque/reports/ARTMAP_digit.pdf

Christoudias C. M., Urtasun R. and Darrell T., 2008. Multi-View Learning in the Presence of View Disagreement, *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
Available: http://people.csail.mit.edu/cmch/pubs/view_disagree_uai08.pdf

Dietterich T. G., 2000. Ensemble methods in machine learning. In J. Kittler and F. Roli (ed.) *Proceedings of the First International Workshop on Multiple Classifier Systems*, Sardinia, Italy, pp. 1-15.

Kakade, S. M., & Foster, D. P., 2007. Multi-view regression via canonical correlation analysis. *Proceedings of the Twentieth Annual Conference on Computational Learning Theory*, San Diego, California, pp. 82-96.

Koon T.C.N., 2007. A Direct Boosting Algorithm for the k-Nearest Neighbor Classifier via Local Warping of the Distance Metric. *Thesis for the M.A. Degree*. Provo: Brigham Young University IS.

Muslea I., Minton S., Knoblock C. A., 2002. Active + Semi-Supervised Learning = Robust Multi-View Learning. *Proceedings of the 19th International Conference on Machine Learning*, Sydney, Australia, pp. 435-442.

Nanuwa S. & Seker H., 2008. Investigation into the role of sequence-driven-features for prediction of protein structural classes, *8th IEEE International Conference in Bioinformatics and Bioengineering*, Athens, Greece .

Okun, O. & Priisalu, H., 2005. Multiple Views in Ensembles of Nearest Neighbor Classifiers. *Proceedings of the ICML Workshop on Learning with Multiple*

Views (in conjunction with the 22nd International Conference on Machine Learning), Bonn, Germany, pp. 51-58.

Sakar O., 2008. A Novel Generalized Mutual Information Approach and Its Use in Feature Selection. *Thesis for the M.A. Degree*. İstanbul: Bahçeşehir University FBE.

Skomorokhov A., 2002. Radial basis function networks in A+. *Proceedings of the 2002 conference on APL: array processing languages: lore, problems, and application.* Madrid, Spain, pp. 198 – 213.

Yarowsky, D., 1995. Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings of the Thirty-Third Annual Conference of the Association for Computational Linguistics.* Cambridge, MA, pp. 189–196.

CURRICULUM VITAE

Name Surname : Heysem KAYA
Address : Bareli İş Merkezi Pazar Sk. No: 2-4 Kat: 5
34349 Dikilitaş / Beşiktaş / İstanbul

Birth Place / Year : Antakya / 1983

Languages : English (fluent) - Chinese (elementary)

Elementary School : Süleyman Nazif Elementary School, Antakya

High School : Hatay Anatolian Technical High School (Dept. of Computer)

BSc : Boğaziçi University - 2006

MSc : Bahçeşehir University - 2009

Name of Institute : Institute of Sciences

Name of Program : Computer Engineering

Work Experience : Software Development Specialist, R&D, CoreLink Teknoloji (Since July 2005)

Awards and Achievements

- Graduation with the top GPA of the Department in BSc
- High Honor and Outstanding Student Certificates of Boğaziçi University
- NOKIA Telecommunication Scholarship (throughout the university education)
- Graduation with the top GPA in High School

Projects Accomplished

- TRic HotSpot GateX (2009): A content filtering and logging appliance developed in accordance with 5651 Internet Regulation Law. The content filtering module is integrated with and accredited by Communications Presidency of Turkish Telecommunication Authority.

- Automatic Configuration System for AC MP202 VoIP Gateways (2008): An alternative approach for provisioning and management of AudioCodes MP202 VoIP gateways was designed and developed. This method was aimed to replace commonly used tr-69 protocol for MP202 product family, providing greater flexibility and easier management.

- MRP System for Textile Companies (2007-2009): A web based specialized MRP system for textile sector do not exist in Turkey. The software was designed according to needs of a specific company and implemented with step-by-step evolutionary approach.

- Orbital Integrated CRM for Telecommunication Companies (2005-2007): A web based CRM application which coordinates integrated ERP, Billing Systems and Telecommunication Infrastructure (SIP Server and Dialer Management System) and gathers all stakeholders (customers, resellers, operators and managers) was successfully developed for a Telco in Turkey.