Alexandru Ioan Cuza University

Faculty of Computer Science

Summary of Doctoral Thesis

A Distributed Statistical Binary Classifier. Probabilistic Vector Machines

Principal advisor:
Prof. Dr. Henri Luchian

Author:
Andrei Sucilă

November 29, 2012

Chapter 1

Introduction

The technological progress of society in the 20th century, especially regarding digital information, its processing, transmission and storage, has lead to a sizeable increase in the volume of data available. Fields such as biology, finance, physics, engineering and many more now dispose of huge amounts of data in electronic format. This encouraged and fueled the development of Data Mining , Pattern Recognition and Machine Learning, all proeminent fields which aide and direct the analysis and processing of such large amounts of data in the pursuit of deriving usable knowledge from it.

A large part of the problems addressed in these fields represent *learning problems.* Tom Mitchell defines this class of problems as:

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Machine learning algorithms are nowadays pervasive in the lives of many. From recognizing the $ signs hand written in a check filled to the bank, to detecting fraudulent bank transactions, exploring medical databases, learning the human genome, optimizing the performance of the engine in a car and many others represent important applications of Machine Learning which influence the activity of millions of

1.1 Research Topic

This thesis is focused on problems regarding linear binary classification, the main purpose being to construct a new binary classifier which simultaneously satisfies a series of criteria which are of high importance to practical applications, criteria which which have not been so far satisfied by any other current approach.

In the class of learning problems, classification takes on an important role. A learning problem for which the sought function has a finite set of output values is named a classification problem.

The function obtained in classification problems as a result of the learning process is referred to as a decision function, the values of which are called classes or labels. When the labels have only two possible values, the problem is named a binary classification problem.

Using decision trees, any classification task can be reduced to a set of binary classification problems.

Among the algorithms which have had great success in binary classification, hyperplane classifiers are some of the most proeminent. Such algorithms use points in a Hilbert space as input data. They build a hyperplane, named separating hyperplane, which divides the space into two semispaces. The points in one of the semispaces are labeled positively whilst the points in the other semispace are labeled negatively. Due to the fact that semispace belongness can be decided by looking at a single scalar product, hyperplane classifiers end up belonging to the linear classifier class, having decision functions which are linear.

The objective of this thesis is to present a new hyperplane classifier which simultaneously satisfies the following criteria:

1. The problem objective should be strongly correlated to the classifier's probability of error. Thus, the objective should take into consideration the distribution of the training set.

2. The mathematical modelling should allow efficient and complete resolution of the underlying optimization problem.

3. The model should allow for kernel function usage. This should facilitate solving nonlinear separation problems.

4. The algorithm should be stable with regard to outliers and should not necesitate linear separation of the training set.

The approaches thus far presented in current literature generally satisfy the third criteria and some satisfy the fourth. None of them simultaneously satisfy both the first and the second criteria.

The algorithm proposed in this thesis will be shown to satisfy all of these criteria simultaneously, representing an unique contribution from this standpoint.

Chapter 2

A Motivating Application of Binary Classification: microRNA Sequences

In recent years, many practical problems have evidenced the need for well perfoming classification algorithms. One such problem comes from the field of bioinformatics, having important implicatinos in the understanding of how living organisms mechanisms function and in the treatment of some difficult diseases, such as cancer.

DNA and RNA sequences represent major components in biological mechanisms. The information retained in coding DNA genes is used in producing proteins. Transmission of this information is usually accomplished by the use of RNA messsenger sequences, mRNA. There are, however, some RNA sequences which do not transmit information. Some of these may actually block the transcription of the information which tipically occurs in the protein creation process, allowing such sequences to regulate the transmission of information.

An important class of sequences with transcription inhibitory role is represented by microRNA sequences. Many recent papers refer this type of sequences as having a very high potential in treating diseases by regulating the expresivity of the genes involved in said diseases. Identifying such sequences is, thus, very important.

This problem can be solved with the help of a binary classifier that would decide whether a sequence is microRNA or not. Such an approach is represented in the yasMiR algorithm. YasMiR builds for each RNA sequence a description composed of specific distances to a fixed set of sequenced, called pivot sequences. This allows each RNA sequence to be represented as a point in $\mathbb{R}^n$. Sequences which have a known type constitute the training set for building a decision function. The used binary classifier is Support Vector Machines (SVM).

The important contribution to this classifier is the addition of feature selection. This is accomplished in two stages. The first one sorts the the features in order of descending relevance for classification, where relevance is measured by the Symmetrical Uncertainty (SU) score. In the second stage redundant features are eliminated with the help of a filter built using the results of Kolmogorov and Smirnov regarding the resemblance of distributions. Of two similar features deemed redundant, the one with the highest score is kept. Similar to Eratosthenes' sieve, each feature which has not yet been eliminated removes all other features redundant to it with a lower SU score.

Starting the selection process from a set of 169 pivots manually chosen, the feature set is reduced corresponding to 90% and 95% relevance levels. The reduced feature sets are shown to yield significantly improved results compared to the original set.

The method also allows for automatic selection of pivot sequences. This is achieved by starting from a set of 10000 randomly generated sequences for which feature selection is applied. The best 13 pivots are then shown to have comparable results to the best manually selected set.

Chapter 3

Probabilistic Vector Machines

The introduction of the SVM classifier allowed the handling of difficult problem, such as the one presented in Chapter 2. However, the SVM model does not satisfy the four criteria presented in Chapter 1. A number of classifiers has been proposed after the introduction of SVM, but none satisfy both the first and the second criteria and, implicitly, none satisfy all four.

In order to obtain a model that does satisfy the criteria, the objective must be based on the probability of error of a linear classifier. This chapter presents the formulation of such an objective which, although is not a convex function, models the maximum of false positive (FP) and false negative probabilities

(FN). It assumes a normal distribution of the signed distances of positively labeled points and similarly for the negatively labeled points. Under this assumption, it is shown that the proposed objective leads to the selection of a separation hyperplane which is optimal with regard to FP and FN probabilities.

Let $S = \{(x_i, y_i) \in H \times \{-1, 1\} | i = \overline{1..m}\}$ be the training set for the classifier. In most instances, the Hilbert space, H, will actually be the n-dimensional real space, $\mathbb{R}^n$, and thus will be identified as such in what follows. Let $S_+ = \{x_i | (x_i, y_i) \in S, y_i = 1\}$ and $S_- = \{x_i | (x_i, y_i) \in S, y_i = -1\}$. These will be named the positive and the negative training sets.

Suppose that a hyperplane in $\mathbb{R}^n$ is expressed through its normal unitary vector and its offset, $(w, b) \in \mathbb{S}_n \times \mathbb{R}$, where $\mathbb{S}_n$ represents the n-dimensional sphere. Then let $D_+(w, b) = \{< x_i, w > +b \in \mathbb{R} | x_i \in S_+\}$ and $D_-(w, b) = \{< x_i, w > +b \in \mathbb{R} | x_i \in S_-\}$ be the signed distances of positive and negative training points.

Let E_+, E_- be the averages of D_+ and D_- respectively. Let σ_+, σ_- be the standard deviations of D_+, D_-. The initially proposed objective is, then:

$$\min_{(w,b) \in \mathbb{S}_n \times \mathbb{R}} \max\{\frac{\sigma_+}{E_+}, -\frac{\sigma_-}{E_-}\}$$

The condition necessary for optimality of this model is that D_+ and D_- be sample sets of normal distributions. It is shown that this condition may be relaxed somewhat, allowing for a broader set of situations to be covered in an optimal manner. Figure 3.1 describes the idea behind choosing this objective.

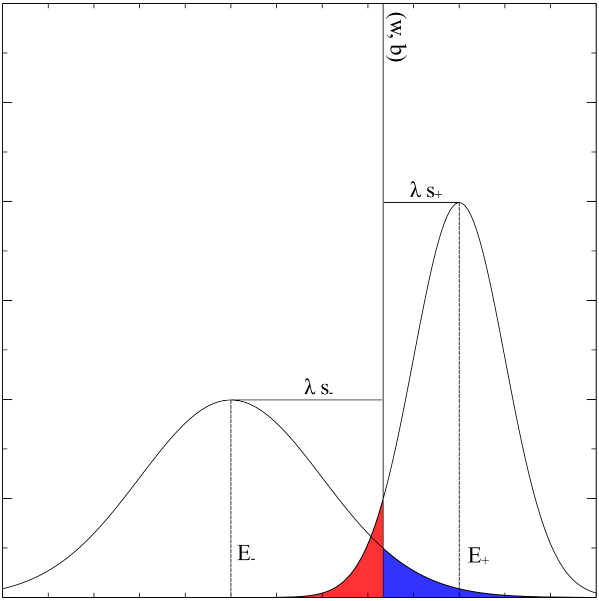

Figure 3.1: The red area corresponds to the FN probability, the blue area to the FP probability. The proposed objective tries to minimize the maximum between these two areas, or, equivalently, to maximize λ.

In order to facilitate obtaining an optimization problem that may be solved completely, such as a convex problem, the standard deviation is replaced with the average deviation. The problem may be expressed as:

$$\begin{cases} \min_{(w,b)\in\mathbb{S}_n\times\mathbb{R}} \max\{\frac{\sigma_+}{E_+}, -\frac{\sigma_-}{E_-}\} \\ E_+ = b + \frac{1}{|S_+|-1}\sum_{x_i\in S_+} < w, x_i > \\ E_- = b + \frac{1}{|S_-|-1}\sum_{x_i\in S_-} < w, x_i > \\ \sigma_+ = \frac{1}{|S_+|-1}\sum_{x_i\in S_+} | < w, x_i > +b - E_+| \\ \sigma_- = \frac{1}{|S_-|-1}\sum_{x_i\in S_-} | < w, x_i > +b - E_+| \end{cases} \tag{3.1}$$

It is first shown that the model maintains its properties and yields the same separating hyperplane when $w \in \mathbb{R}^n$, as opposed to having $w \in \mathbb{S}_n$. It is then shown that the modelling of σ_+ and σ_- may be relaxed to a liniar formula, transforming the model to:

$$\begin{cases} \min_{(w,b)\in\mathbb{S}_n\times\mathbb{R}} \max\{\frac{\sigma_+}{E_+}, -\frac{\sigma_-}{E_-}\} \\ E_+ = b + \frac{1}{|S_+|-1}\sum_{x_i\in S_+} < w, x_i > \\ E_- = b + \frac{1}{|S_-|-1}\sum_{x_i\in S_-} < w, x_i > \\ \sigma_+^i \geq | < w, x_i > +b - E_+|, \forall x_i \in S_+ \\ \sigma_-^i \geq | < w, x_i > +b - E_-|, \forall x_i \in S_- \\ \sigma_+ = \frac{1}{|S_+|-1}\sum_{x_i\in S_+} \sigma_+^i \\ \sigma_- = \frac{1}{|S_-|-1}\sum_{x_i\in S_-} \sigma_-^i \end{cases} \tag{3.2}$$

The important result is that the (3.1) and (3.2) optimization problems have the same optimal set and are, thus, equivalent. This allows a modelling of the which uses only linear inequal-

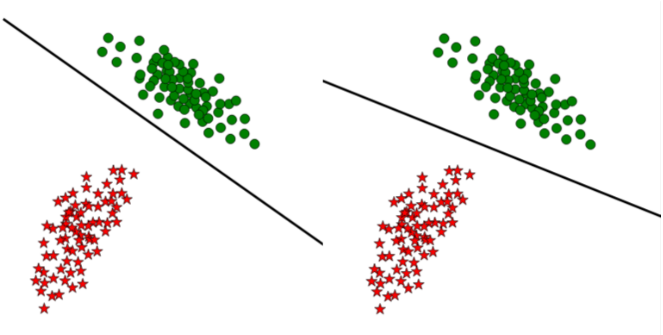

Figure 3.2: PVM separation on the left side; SVM separation on the right. The separation induced by PVM takes into account the distribution of the two training sets, not just the border members.

ities. The classifier obtained by solving this system is entitled Probabilistic Vector Machines (PVM).

It is important to nota that, due to the formulation based on a statistical model of the data, the resulting classifier is robust to outliers and will not require linear separation of the data. To better depict the idea of the classifier, Figure 3.2 compares the separation induced by PVM with that induced by SVM. Figure 3.3 shows the effect of introducing outliers.

System 3.2 uses linear constraints, but the objective itself is not a convex function. The fundamental observation that allows resolving this situation is that although the objective is

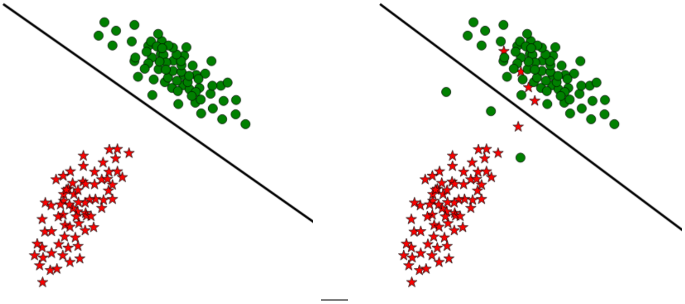

Figure 3.3: The result of training a PVM classifier with outliers. Figure 3.3a shows the result of training using the same data as Figure 3.2. Figure 3.3b shows the result of training after adding outliers to the same data.

not convex, it is, however, quasiconvex for positive values of the denominator, meaning each sublevel set is convex. This allows complete optimization via bisection, which generates a sequence of linear feasibility problems. The reduction to linear feasibility is also important due to the fact that it allows the usage of readily available linear optimization libraries.

The model presented thus far simultaneously fulfills criteria 1, 2 and 4. All that is left is to observe that the reduction to linear feasibility problems facilitates expressing the model only through scalar products. This allows for the usage of kernel functions, leading to the satisfaction of all four criteria outlined at the beginning of this thesis. It also bears mentioning that the statistical model used allows the usage of training weights, thus dealing with skewed data sets.

The results of comparing PVM and SVM presented in Table 3.1 clearly favors the PVM model, with the exception of one problem which has a distribution of data that is not normal and another which has too few training instances for the statistical model to be stable.

Data Set	SVM Acc	PVM Acc	PVM Obj.
Credit Screening	76.98	**84.13**	0.5397
Heart Cleveland	54.23	**82.11**	0.7437
Ionosphere	64.28	**82.18**	0.6112
Liver Bupa	**68.45**	66.91	1.715
Heart SPECT train	**76.45**	62.31	0.7545
Heart SPECTF train	56.87	**79.4**	0.3973

Table 3.1: Comparison between PVM and SVM using a scalar kernel. The results shown represent the average of 30 runs.

Chapter 4

Convex Feasibility

During the testing phase presented in Chapter 3, one of the critical points was choosing the linear optimization library. A large set of options was explored, but none of these proved viable in solving all the instances in which kernel functions are used or for problems with a record count higher than 1000 records. Especially for larger data sets, both the solve time and the memory requirements increased drastically with size. It soon became obvious that a new algorithm was required, which would be able to handle the specific requirements of the feasibility problems that arise in the course of PVM training.

The criteria such an algorithm should satisfy can be summed as:

1. Good memory scaling to allow large instances to be solved.

2. The possibility of distributed computation in order to be able to use more than one computational unit.

3. Stability in the case of severely constrained cases.

In order to fulfill these criteria it is essential to avoid any matrix inversions, which, although greatly accelerate the solution search process and can be relatively stable, scale poorly with problem size in regard to memory requirements. These requirements all point to projection algorithms.

Using as a starting point one of the most competitive algorithms which do not use matrix inversions, Component Averaged Row Projections (CARP), a new algorithm is proposed. However, only the distribution scheme was retained from CARP. On every local machine, however, the solution iteration is done via a dynamic weighted average of projections onto subsets of the semispaces which are defined by the system constraints. The weights used are recomputed at each iteration and are chosen as proportional to a power of the distances from the projection semispaces. Computing the weights in this manner implies that the constraints which are least satisfied bear the greatest influence on the solution computation process. The algorithm is named Distance Weighted Projection Operator (DWPO).

It is shown that DWPO converges when the problem is feasible, Furthermore, it is shown that the global iterations are in fact Fejér monotone with respect to the feasible set. It is important to note that DWPO requires only a projection operator. As a consequence of this, it can be used for solving general convex feasibility problems for which the spaces defined by the constraints allow an accurate projection operator. Another important aspect of the algorithm is that a number of classical and recent algorithms may be obtained as subcases of it.

Implementation of DWPO was done in C++ and allowed for a distribution of computations onto a multitude of computational units. Although the algorithm can be used to solve a large class of convex feasibility problems, the implementation was specific to the problems generated by the PVM training process. This allowed a substantial set of advantages over the general formulation. Specifically, by exploiting the problem structure. the memory requirements were reduced by a factor of 8, whilst speed was increased by a factor of 2.

Testing compared the proposed algorithm with the linear optimization packages available as well as with the feasibility algorithms which can be obtained as a subset of DWPO. Among these is also CARP. The substantial advantages over CARP are:

1. A reduced number of blocks into which the system needs to be divided. This has two important consequences:
    - Reduced network traffic.
    - Better synchronization among blocks.
2. Due to the specific implementation, a lower number of blocks also reduces the required memory.

Compared with the linear optimization packages, DWPO obtained similar solving times, with substantially reduced memory requirements. This allowed the resolution of far larger problems than the linear packages could handle. Comparisons were made using only one computational unit. However, due to DWPO having an excellent behaviour with regard to distribution, obtaining nearly a linear speedup in the number of computational units used, it would be trivial to obtain higher speeds that the simplex algorithms.

Chapter 5

Hybrid Clustering and Classification

The testing that was undertaken in Chapter 4 clearly showed that, in the event of a lack of an adequate linear optimization library, the current simplex solvers available allow only for rather small problems to be solved. In order to compensate for this, a new model which allows the hybridization of PVM with clustering is presented. This model makes use of the linearity of the statistical measures used.

The typical procedure when hybridizing these two algorithms is to replace the trainig set with the cluster centers obtained as output from a clustering algorithm, with the centers considered as the average of cluster points. This can introduce skeweness.

Using the number of points in a cluster as the weight for that cluster's center compensates for this and allows the E_+, E_- averages to have the same value as those obtained on the original training set. This method of weighing does not, however, recreate the values for the average deviations. When tested in this form, there is a substantial accuracy loss incurred.

Analyzing the problem, several causes for the accuracy decrease became obvious:

1. The substantial difference between the values of the average deviations obtained on the original training set and the clustered training set.

2. When using kernel functions, the fact that the cluster center is computed for the points in $\mathbb{R}^n$ and not in the space where they are projected by the kernel functions induces a substantial error in the E_+, E_- computation.

The second point can be addressed by modifying the mathematical model to allow for an implicit cluster center formulation which makes use of only cluster belongness of points. It is shown that this model will always correctly reproduce the E_+, E_- averages, irrespective of whether or not kernel functions are used.

In order to address the first point, suppose that all clusters have all their points with a signed distance above or bellow the associated average. Then the values for σ_+, σ_- computed for the clustered training set would coincide with the values computed for the nonclustered training set. This observation suggests that clusters with points both above and bellow the associated average should be split. For such a cluster, the relative aberration induced by it in the average deviation can be computed.

An iterative algorithm is built which, using the clustered data set as a starting point, first trains the weighted PVM classifier and then splits the clusters with the highest relative induced aberrations. This iterative process is stopped when the total relative aberration drops bellow a certain threshold. The algorithm is named Clustering Probabilistic Vector Machines (C–PVM).

Table 5.1 shows the results of C–PVM obtained with a relative aberration threshold of 5%. For comparison, the table also shows the results obtained by the normal PVM algorithm and by the SVM algorithm using the same kernel.

Data Set	SVM	PVM	C–PVM
Credit Screening	76.98	84.13	84.76
Heart Disease Cleaveland	54.23	82.11	80.91
Ionosphere	64.28	82.18	82.13
Liver	68.45	66.91	66.31
Heart Spect Train	76.45	62.31	66.90

Table 5.1: Results obtained for Clustering PVM using cluster division.

As is evident from Table 5.1, the results of C–PVM are very close to those obtained by PVM and both clearly outperform SVM. The C–PVM results can be improved by lowering the relative aberration threshold. In fact, when this threshold is 0, the resulting linear classifier is perfectly identical to that obtained by the normal PVM algorithm on the nonclustered data and, as a consequence, lead to the same result. The only drawback of using a threshold of 0 for the relative aberration is that there is a large number of iterations spent breaking only small clusters of 2-3 points. This leads to a substantial total training time increase, without a converse increase in accuracy.

The most important benefit of C–PVM is the effective reduction of the training set dimension. For the same data sets, Table 5.2 shows the relative sizes of the final implicit clusters, obtained as an average over 30 runs. The average relative size is 0.37 of the original data set. Consider that the feasibility systems have a number of nonzero terms of the order $\mathbb{O}(m^2)$ with regard to the training set size and that the linear optimization libraries use at least $\mathbb{O}(nz^2)$ memory, where nz is the number of nonzeroes. It becomes clear that the training set reduction obtained by C–PVM leads to a substantial reduction of the memory required for training. The total execution time also decreases because, for a linear optimization algorithm, the time required to solve a problem quickly increases with the number of nonzeroes – in the most unfavorable cases, it rises exponentially.

Data Set	Initial Size	Averaged Clustered Size
Credit Screening	690	225.03
Heart Disease Cleaveland	303	104.83
Ionosphere	351	145.7
Liver	341	67.4
Heart Spect Train	80	48.17

Table 5.2: Average sizes for training sets obtained by C–PVM

Chapter 6

Conclusions

The present thesis proposes a new binary classification algorithm, PVM, which resolves a set of existing issues with current algorithms. In the pursuit of widening its range of applicability, a complementary set of algorithms is also introduced. The first of these, DWPO, deals with the issue of convex feasibility and proves to be competitive, in terms of speed, with some of the best current algorithms. In terms of memory requirements, DWPO proves to be extremely efficient when compared to state of the art simplex algorithms.

The second of the complementary algorithm, C–PVM, proves to be extremely useful at reducing the size of training datasets. The size reduction leads to major benefits both in terms of memory usage and running times. These benefits do not incur any noticeable accuracy loss.

For future research, a more general and flexible implementation of DWPO will be sought, which should also allow different projection operators to deal with some highly constrained cases. Another important direction will be adding a merge procedure to C–PVM which should further reduce the dimension of training data sets.

Obtaining a model for PVM which can deal with very large data bases will allow it to be applied to important current problems, such as microRNA sequence classification.

Chapter 7

Scientific Contribution

In this thesis, the scientific contributions, in order of their appearance, consist of:

1. Applying the Kolmogorov-Smirnov filter in order to reduce the size of the microRNA sequence identification problem

2. Obtaining an objective for a linear classifier which reflects the FP and FN probabilities

3. Formulating the initial PVM model and obtaining an equivalent form which uses only linear constraints

4. Obtaining the sequence of feasibility systems which allows the optimal resolution of the PVM model
5. Introducing kernel functions and training weights in PVM
6. Formulation of the DWPO algorithm
7. Obtaining a sequential form for DWPO
8. Proving the convergence of DWPO
9. Formulating the halting criteria for DWPO which allows significant cost reduction in the first stage of the algorithm
10. Using training weights for reconstructing the statistical model on a clustered data set
11. Formulating the implicit cluster representation model which allows correct usage of kernel functions
12. The cluster division algorithm, C–PVM

These contributions have been presented in the following papers and conference participations:

1. Păsăilă D., Sucilă A., Panţiru S., Ciortuz L., "Yet Another SVM for MicroRNA recognition: yasMiR", tehnical report, Facultatea de Informatică, Universitatea Alexandru Ioan Cuza, Iaşi, 2010
2. Păsăilă D., Sucilă A., Mohorianu I., Panţiru S., Ciortuz L., "MiRNA Recognition with the yasMiR System: The

Quest for Further Improvements", Advances in Experimental Medicine and Biology, Software Tools and Algorithms for Biological Systems, vol 696, pp 17-25, 2010

3. Sucilă A., Henri Luchian, "Probabilistic Vector Machine", In Proceedings of the 7th International Conference on Data Mining, DMIN'11, pp 198-202, Las Vegas, USA, 2011

4. Sucilă A., Cimpoeşu M., Henri Luchian, "A Distributed Dense Linear Feasibility Systems Solver", Accepted at the 14th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2012, Timişoara, România, 2012

5. Sucilă A., Cimpoeşu M., Henri Luchian, "A Statistical Binary Classifier. Probabilistic Vector Machine", under review at Information Processing Letters

6. Sucilă A., Cimpoeşu M., Henri Luchian, "Clustering Probabilistic Vector Machine. A Hybrid Clustering and Classification Algorithm", under review at Computational Statistics & Data Analysis