Version Space Reduction Based on Ensembles of Dissimilar Balanced Perceptrons

Karen Braga Enes[1,2], **Saulo Moraes Villela**[1], **Raul Fonseca Neto**[1]
[1]Department of Computer Science, Universidade Federal de Juiz de Fora, Brazil
[2]Department of Computer Science, Universidade Federal de Minas Gerais, Brazil
karen@dcc.ufmg.br, {saulo.moraes, raulfonseca.neto}@ufjf.edu.br

Abstract

A classifier able to minimize the generalization error of a particular problem for any set of unseen samples is named Bayes-optimal classifier. The hypothesis induced by such classifier is equivalent to the optimal Bayes point, which is approximately equivalent to the center of mass of the version space. However, there are only a few methods for estimating the center of mass and most of them are computationally expensive or impractical, especially for large datasets. In this paper we present the Version Space Reduction Machine (VSRM), a new method that obtains an approximation of the center of mass. The method works by means of successive reductions of the version space which are consistent with an oracle's decision. This oracle is represented by the majority voting of an ensemble, whose components must contain a reasonable diversity level to ensure an effective approximation. We conduct an experimental study on microarray datasets and assess the performance of the proposed method compared to Support Vector Machine and Bayes Point Machine. Our method consistently outperforms the others. Such result indicates that the proposed method provides a better approximation of the center of mass.

1 Introduction

Considering the classification problem in a Bayesian approach, the hypothesis given by a linear classifier, that is able to predict which class minimizes the error for any given sample is called a Bayes-optimal classifier [Michalski *et al.*, 2013]. However, the problem arises when find this hypothesis involves analyzing every single solution that belongs to the hypothesis space. Unfortunately, in that case, find the Bayes-optimal classifier is absolutely expensive and cannot be implemented for large datasets.

The advantage of this technique is that this hypothesis outperforms every other solution belonging to the hypothesis space, in average [Bishop, 1995]. The Bayes-optimal classifier induces an hypothesis represented by a point in the hypothesis space, named optimal Bayes point. In [Watkin, 1993] and [Opper and Kinzel, 1996], the authors proved that the optimal Bayes point solution can be approximated by the center of mass of the version space. Watkin and others refer to this point as the "optimal Perceptron". However, find the center of mass of the version space is equally complicated. In this sense, we are looking for methods that are capable to generate a hypothesis that most closely approximates to the center of mass. Many researches have been developed in order to estimate the center of mass accurately. Although it is a well-studied problem, the solutions proposed so far besides not effective in most cases, are also not applicable in real world datasets. Therefore, it is still an open problem.

Some methods focus on accurately calculate the center of mass while others try an approximation of it. In [Graepel *et al.*, 1999], the authors propose the Bayesian Transduction (BT) for approximate the center of mass of the version space. BT works according to the classification of new samples which splits the version space into two half-spaces. One with the lowest volume is discarded. However, this approach is slow and works only on transduction problems. [Trafalis and Malyscheff, 2002] propose the Analytic Center Machine (ACM) to calculate the analytic center of the version space, an approximation of the center of mass. ACM solves a minimization problem using Newtons method and a logarithmic barrier function. The main problem of this approach is the treatment given to the redundant constraints that deviate the solution of ACM from the real analytic center. Based on the recursive method proposed by [Lasserre, 1983] to compute the volume of a convex polytope, [Maire, 2003] proposes the Balancing Board Machines (BBM) to compute an approximation of the centroid of higher dimensional polytope. However, the method is absolutely expensive and not applicable in practice. In [Herbrich *et al.*, 2001], the authors present two implementations of the Bayes Point Machine (BPM) to estimate de the center of mass. The first one considers the computation of the volume of the polytope introduced by [Ruján, 1997] expanded to kernel space. The second algorithm proposed considers the hypothesis equivalent to an average of an ensemble of Perceptrons as the center of mass. This approach has been considered the state-of-art in Bayesian-optimal classifiers and will be discussed later in this paper.

In this paper, we present the Version Space Reduction Machine (VSRM) to approximate the center of mass of the version space. We determine the approximation by successive reductions of the version space. The reduction process is de-

fined by considering a cutting plane method that uses as an oracle the majority voting of an ensemble, whose components are diverse and well distributed. Moreover, we successively introduce cutting constraints in order to allow us to rebuild the reduced version space. By doing this, in the end we are able to generate a hypothesis equivalent to the representative majority voting ensemble. We suggest that this hypothesis is an approximation of the center of mass. Also, this method is efficient enough to be applicable to large datasets, since the reduction process is done from updating the constraints of the problem, instead of including additional restrictions.

In order to verify our claims, we apply our method to microarray gene expression data. This technology has a large impact on cancer diagnosis research and provides high dimensionality datasets, with a small number of samples. Our experimental study focus on 6 linearly separable cancer datasets. We compare the proposed VSRM method to the Support Vector Machine (SVM) [Boser *et al.*, 1992], [Vapnik, 1995] and the BPM. From the reported results, we observe that our method outperforms the other classifiers in almost all considered datasets. This result indicates that our method tends to provide a better approximation of the center of mass, since the generalization error provided by our hypothesis is lower than the error given by the other methods. We also present statistical tests, in order to confirm the relevance of our empirical study.

The remaining of this paper is structured as follows. In the following section, we review the basic ideas of binary classification problem with a particular focus on linear classifiers. We also define the version space and review two methods that work as approximations of the center of mass of the version space: SVM and BPM. In Section 3, we present the Variable Margin Perceptron (VMP), an algorithm that evolves a Perceptron solution maintaining this solution within a range of values defined by two margin values. The VMP corresponds to the base of VSRM, which is presented in Section 4. An extensive list of experimental results is presented in Section 5. In Section 6 we give some considerations and discuss some theoretical extensions of the presented method.

2 Binary classification

Consider the supervised learning paradigm, a binary classification task can be defined as follows. Let the input data, denoted by $Z = \{(x_i, y_i) : i \in \{1, 2, \ldots, m\}\}$, be a training set of m examples, for some unknown function h. The $x_i \in \mathbb{R}^d$ are vectors, whose components are discrete or real-valued, called features. The y values are mapped from a discrete set of classes, $y_i \in \{-1, +1\}$. Given a set of training examples, a learning algorithm outputs a classifier which consists of a hypothesis (represented by a hyperplane) about the function $h : \mathbb{R}^d \to \{-1, +1\}$. The classifier predicts the corresponding values of y when additional samples of x are presented. The additional samples are called testing set.

In those cases where the two classes are linearly separable, linear classifiers work on defining the hyperplane corresponding to the function h based on a linear combination of the features vectors and a weight vector w and b, the bias term.

The weight vector is updating according to the training set error. In this case, the generated hypothesis gives the probability of a new sample belongs to each of the two classes. We obtain the classifier (hypothesis) answer when all the training instances are correctly classified, thus

$$y_i \left(\langle w, x_i \rangle + b \right) \geq 0, \text{ for all } i \in \{1, 2, \ldots, m\}.$$

The set of all hypotheses given by the learning algorithm defines the hypothesis space. Besides that, we can also define the set of all hypotheses given by linear classifiers that are consistent with the training set as the version space.

2.1 Version space

According to [Herbrich, 2001], given the training set Z and the consistent hypothesis (solution) set H of a problem, the version space $V(Z)$ is defined in terms of H by

$$V_H(Z) \stackrel{def}{=} \{h \in H : h(x_i) = y_i\} \subseteq H,$$

for all $i \in \{1, 2, \ldots, m\}$, i.e., the set of all consistent classifiers with the training set. In particular, for linear classifiers, the version space can also be defined as the set of consistent weight vectors, for all $i \in \{1, 2, \ldots, m\}$, given by

$$V_W(Z) \stackrel{def}{=} \{(w, b) \in W : y_i \left(\langle w, x_i \rangle + b \right) > 0\} \subseteq W,$$

where $W = \{(w, b) : ||(w, b)||_2 = 1\}$ is the isomorphic unit sphere to the hypothesis space of the problem. To simplify the use of notation, we will represent the weight vector (w, b), equivalent to a point in the version space, only by w.

2.2 Support Vector Machines – SVM

SVMs are maximal margin kernel classifiers which were introduced by [Boser *et al.*, 1992], [Vapnik, 1995]. This technique aims to separate the training set by a hyperplane that maximizes the distance from members of opposite classes. In order to obtain the maximal margin hyperplane that correctly classifies all patterns in the training set, it is necessary to solve an optimization problem.

Considering the version space approach, SVMs provide as solution the center of the largest inscribed hypersphere in $V(Z)$. Usually, this solution is reasonable and, in most cases, keeps lower generalization error. However, if the version space is elongated or asymmetric, the solution provide by SVMs tends to present a notable decrease of the generalization error [Tong and Koller, 2002]. Besides that, [Herbrich *et al.*, 2001] show that SVMs can also be viewed as an approximation of the center of mass of version space in the noise free scenario.

2.3 Bayes Point Machines – BPM

BPMs are kernel classifiers that aims to return an approximation for the center of mass, following the Bayesian classification strategy. Such classifiers were originally presented in two versions by [Herbrich *et al.*, 2001]: a kernel billiard sampling algorithm and an approximative method based on an ensemble of Perceptrons. However, only the latter method can be applied to large datasets. For this reason, we only dispense attention to this approach.

The technique provides by the BPMs works by means of generating a fixed number of Perceptrons, considering for

each one $||w||_2 = 1$, trained with different permutations of the training set, and average them into a single classifier. The hypothesis induced by such ensemble of classifiers very closely approximates the Bayesian classification strategy [Herbrich, 2001]. Nowadays, BPMs are considered the state-of-art in Bayes-optimal classifier methods, since they provide a better approximation of the center of mass, outperforming other strategies for approximation such as SVM.

3 Variable Margin Perceptron – VMP

[Leite and Fonseca Neto, 2008] proposed an extension of the Perceptron algorithm [Rosenblatt, 1958] that finds the solution of a linearly separable problem given a fixed margin γ_f. The algorithm, called Fixed Margin Perceptron (FMP), can establish a fixed margin value that will be attended by a separating hyperplane, such that

$$y_i \left(\langle w, x_i \rangle + b\right) \geq \gamma_f ||w||_2, \text{ for all } i \in \{1, 2, \ldots, m\}.$$

We propose a modification of FMP, in order to introduce two distinct fixed margin values for each sample of the problem, defining a variable margin. We call this modified formulation as Variable Margin Perceptron. The purpose of this algorithm is to generate a Perceptron solution, feasible in $V(Z)$, belonging to a set of fixed margin interval. By doing this, we obtain an interior point in $V(Z)$. Those points belonging to the boundary of $V(Z)$ are not considered feasible. We define the VMP margin values as one lower γ^{low} and one upper γ^{upr}.

3.1 Correction rules

To construct the VMP algorithm, the correction rule of the FMP algorithm must be modified to allow each point be corrected in relation to each one of two margins. Thus, the viability criterion of a point must also be modified. A point is feasible in the version space if obeys the following inequalities

$$y_i \left(\langle w, x_i \rangle + b\right) > \gamma_i^{low} ||w||_2$$
$$y_i \left(\langle w, x_i \rangle + b\right) < \gamma_i^{upr} ||w||_2.$$

In that case where we can not satisfy the viability criterion according to γ_i^{low} for a given point w, i.e., if $y_i \left(\langle w, x_i \rangle + b\right) \leq \gamma_i^{low} ||w||_2$, we must applied the correction rule given by

$$w^{t+1} \leftarrow w^t - \frac{\eta \gamma_i^{low} w^t}{||w||_2} + \eta y_i x_i.$$

On the other hand, if the criterion of viability is not satisfied for a point w according to γ_i^{upr}, i.e., if $y_i \left(\langle w, x_i \rangle + b\right) \geq \gamma_i^{upr} ||w||_2$ or $-y_i \left(\langle w, x_i \rangle + b\right) \leq -\gamma_i^{upr} ||w||_2$, the following correction rule must be applied

$$w^{t+1} \leftarrow w^t + \frac{\eta \gamma_i^{upr} w^t}{||w||_2} + \eta y_i x_i.$$

Note that, although we are including a new viability criterion, the computational cost of the VMP remains the same as FMP.

Figure 1 illustrates the two possible updating cases. Consider the version space $V(Z)$ according to W. The introduction of two new feasible points w' e w'' having two distinct margin values, lower and upper, characterizes the two cases of VMP in relation to R_i constraint according to the x_i sample.

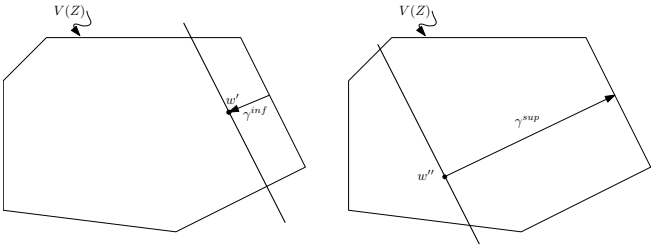

Figure 1: Introduction of two feasible points, given by the convergence of VMP.

3.2 Pseudocode

Algorithm 1 describes the Variable Margin Perceptron.

Algorithm 1: Variable Margin Perceptron

Input: training set $Z = \{(x_i, y_i)\} : i \in \{1, 2, \ldots, m\}$;
upper bound for the number of epochs max;
lower geometric margin vector γ^{low};
upper geometric margin vector γ^{upr};
learning rate η;
Output: normal vector w and bias b if a solution is found in less than max epochs;

```
begin
    initialize (w^0, b^0);
    j ← 0;
    t ← 0;
    stop ← false;
    while j ≤ max and ¬stop do
        error ← false;
        for i ← 1 to m do
            if y_i (⟨w^t, x_i⟩ + b^t) ≤ γ_i^low ||w||_2 then
                w^{t+1} ← w^t − ηγ_i^low w^t/||w||_2 + ηy_i x_i;
                b^{t+1} ← b^t + ηy_i;
                t ← t + 1;
                error ← true;
            else if y_i (⟨w^t, x_i⟩ + b^t) ≥ γ_i^upr ||w||_2 then
                w^{t+1} ← w^t + ηγ_i^upr w^t/||w||_2 + ηy_i x_i;
                b^{t+1} ← b^t + ηy_i;
                t ← t + 1;
                error ← true;
            end if
        end for
        if ¬error then
            stop ← true;
        end if
        j ← j + 1;
    end while
end
```

4 Version Space Reduction Machine – VSRM

We propose the VSRM method in order to provide an effective approximation of the center of mass of the version space. Thus, it is a Bayes-optimal classifier.

The reduction process applied in VSRM is based on the introduction of successive cutting planes. A cutting plane is generated by a parallel hyperplane to the solution given by the VMP algorithm. When we add a cutting plane, $V(Z)$ is divided into two half-spaces. To correctly reduce the version space it is necessary to discard the half-space with the lowest volume. The decision regarding to the discard of the half-space must agree with an oracle answer. For the oracle, any feasible classifier (or ensemble) in $V(Z)$ can be used. However, the construction of a good oracle can represent a better approximation.

4.1 The oracle and its diversity

For the choice of the half-space to be preserved, VSRM considers the correlation of the evaluated hypothesis according to the oracle. In those cases where the oracle is a single classifier, the correlation reflects the half-space that contains the hypothesis generated by the oracle. If the oracle is an ensemble of classifiers, the agreement reflects the half-space that contains the greater number of components of the ensemble, which is characterized as the majority voting of the oracle. This agreement tends to reflect the half-space with the highest volume. However, the biggest half-space is directly related to the distribution of the components. According to [Herbrich, 2001], an ensemble is well distributed in $V(Z)$ if the components are the most diverse possible. Figure 2 illustrates two cases of component distribution in the version space.

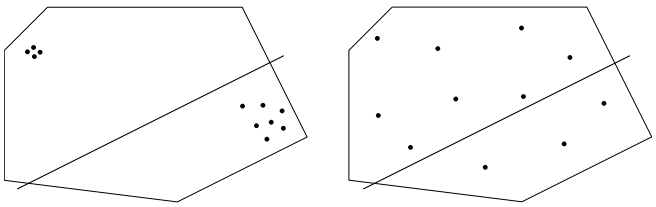

Figure 2: Poorly and well distributed ensemble, respectively.

The first case presents a poorly distributed ensemble and the second case is a well distributed ensemble. It is easy to see that the distribution of the ensemble directly affects the choice of the discarded space, according to the majority voting of the components and based on the cutting plane represented on the figure.

Ensemble of Balanced Perceptrons

In particular, in this study, we choose as oracle the Ensemble of Balanced Perceptrons with a dissimilarity measure (EBPd) [Enes *et al.*, 2015]. This choice was based on two main aspects: it is a kernel method, with a Perceptron as base classifier, and for its good distribution and diversity of the components.

The EBPd works by means of combine some base classifiers called Balanced Perceptron (BP) which are selected by a set of three diversity measures. Basically, the authors proposed the BP as an alternative to improve the individual accuracy of the original Perceptron, by balancing the final hyperplane. The balanced Perceptrons generated are diverse since, for each Perceptron, a different initial weight vector are presented and a different permutation of the training set is constructed. Also, the components are selected by a dissimilarity measure based on the Euclidean distance. In the end, the selected components are combined by a majority voting strategy. For further and detailed information, refer to [Enes *et al.*, 2015].

4.2 Formulation

The first step of VSRM is the oracle construction from the EBPd, whose components are consistent hypotheses in $V(Z)$. At each iteration of the algorithm, VPM generates a feasible point and, from VSRM, we construct a cutting hyperplane, passing through w and parallel to the restriction R_i. We define the direction of the hyperplane based on the direction of R_i which is orthogonal to the normal vector given by $y_i x_i$.

This hyperplane divides the $V(Z)$ into two half-space. The voting process, based on the evaluation of each component of the ensemble, determines the choice of which half-space must be preserved. Consider $h_k = (w_k, b_k)$ as a component of the ensemble. Thus, for each sample in the training set Z, we have the following voting possibilities

$$\begin{cases} y_i \left(\langle (w_k - w), x_i \rangle + (b_k - b) \right) \geq 0, low \leftarrow low + 1 \\ y_i \left(\langle (w_k - w), x_i \rangle + (b_k - b) \right) < 0, upr \leftarrow upr + 1, \end{cases}$$

Both margins γ^{low} and γ^{upr} are associated to the votes of low and upr, respectively. As a result, the update of one of the two margins and the subsequent discard of the half-space, depends on the outcome of the vote. If the number of votes for low is greater than the votes for upr, the following updating rule must be applied

$$\gamma_i^{low} = \left(\langle w, x_i \rangle + b \right) / ||w||_2,$$

otherwise, we apply the updating rule according to γ^{upr}

$$\gamma_i^{upr} = \left(\langle w, x_i \rangle + b \right) / ||w||_2.$$

We can observe that, as the point is feasible in the version space, this updating value does not violate the defined range of VMP for the margin values, maintaining the relationship $\gamma_i^{low} < \gamma_i^{upr}$.

The iterative procedure of the VSRM should be repeated while exists a feasible point in the reduce version space, given by the convergence of VMP. Note that, in the end of the procedure, all constraints will be updated according to the two margin values, attending the following condition

$$\gamma_i^{low} < \left(\langle w, x_i \rangle + b \right) / ||w||_2 < \gamma_i^{upr}.$$

The last feasible point, found by VSRM, defines the approximation of the center of mass of the version space.

Is worth mentioning that, we consider the reduced version space just for updating the point w to generate a new feasible solution and a new cutting plane. However, in order to define the half-space that agrees with the oracle, the original version space must always be considered and thus all components of the ensemble are used in the voting process. Also, note that we reduce the version space without including any new constraints to the problem.

Figure 3 shows the convergence of the VSRM, characterizing the step-by-step of the method on a hypothetical problem with 6 restrictions in $V(Z)$. For example, consider the voting scheme presented in Figure 3-(b). We obtain, by the voting process, 9 votes for updating γ^{low} against 2 votes for updating γ^{upr}. In this case, we are able to discard the half-space with less components, given by γ^{low}. On the other hand, consider now Figure 3-(c). In this case, we discard the half-space given by the update of γ^{upr} (1 vote against 10 votes).

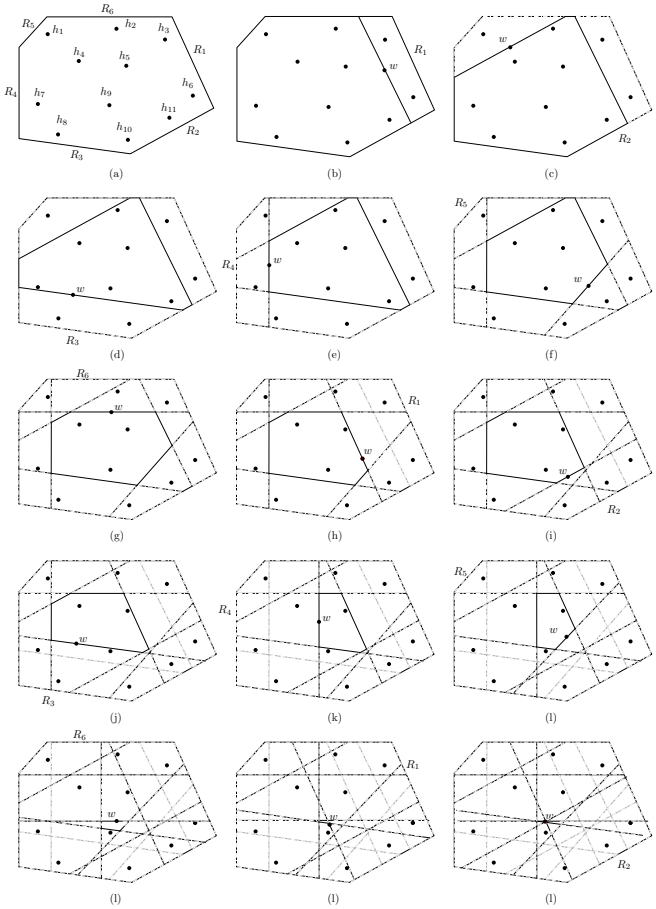

Figure 3: Illustration of the strategy employed by the VSRM.

4.3 Pseudocode

Algorithm 2 describes the Version Space Reduction Machine.

Algorithm 2: Version Space Reduction Machine

Input: training set $Z = \{(x_i, y_i)\} : i \in \{1, 2, \ldots, m\}$;
upper bound for the number of epochs max;
number of components of the ensemble $size$;
dissimilarity factor of the ensemble ε;
learning rate η;
Output: normal vector w and bias b;

begin
  $ensemble \leftarrow$ EBPd$(Z, \eta, size, \varepsilon, max)$;
  for $i \leftarrow 1$ **to** m **do**
    $\gamma_i^{low} \leftarrow 0$;
    $\gamma_i^{upr} \leftarrow \infty$;
  end for
  $i \leftarrow 0$;
  repeat
    $(w, b) \leftarrow$ VMP$(Z, \gamma^{low}, \gamma^{upr}, max)$;
    normalize (w, b);
    $low \leftarrow 0$;
    $upr \leftarrow 0$;
    for $k \leftarrow 1$ **to** $size$ **do**
      if $y_i(\langle (w_k - w), x_i \rangle + (b_k - b)) \geq 0$ **then**
        $low \leftarrow low + 1$;
      else
        $upr \leftarrow upr + 1$;
      end if
    end for
    if $low > upr$ **then**
      $\gamma_i^{low} \leftarrow (\langle w, x_i \rangle + b) / ||w||_2$;
    else
      $\gamma_i^{upr} \leftarrow (\langle w, x_i \rangle + b) / ||w||_2$;
    end if
    $i \leftarrow i + 1 \,\%\, m$;
  until *the convergence of VMP in* max *iterations is not achieved*;
end

5 Experimental Study and Results

5.1 Datasets

Clinical datasets such as DNA microarrays are particularly complex and difficult to work with, due to its high dimensionality and few samples, resulting in large computational costs. It is especially challenging to obtain generalizable decisions. We use six datasets derived from gene selection, which are related to cancer classification and genetic disease studies. The datasets can be found in [Glaab *et al.*, 2012], [Zhu *et al.*, 2007] or [Golub *et al.*, 1999]. Table 1 summarizes the main information.

Table 1: Information about the considered datasets.

Set	Features	Samples		
		Pos.	Neg.	Total
Prostate	12600	50	52	102
Breast	12625	10	14	24
Colon	2000	22	40	62
Leukemia	7129	47	25	72
DLBCL	5468	58	19	77
CNS	7129	21	39	60

5.2 Experimental Setting

Considering the random nature of the methods, for all the experiments, we employ a 10x10-10-fold cross-validation strategy, expect in the SVM case, where we employ a 1x10-10-fold. These schemes were adopted in order to reduce the bias

of the methods. In a stratified cross-validation strategy, each fold always maintain the percentage os data points belonging to each class [Kohavi, 1995]. For more accurate comparisons we always select, for each dataset, the same training and test sets and also the same 10 subsets for cross-validations, preserving the generating seed associated to the random process. The learning rate parameter was set to $\eta = 0.05$, for all the Perceptrons.

We quantify the classification effectiveness of the classifiers through the conventional error rate, i.e, the percentage of test instances incorrectly classified. In order to check for statistical significance of the results we performed the Friedman Rank Sum and the post-hoc Nemenyi statistical tests [Demsar, 2006]. Also, we run the statistical tests using the following R package: Pairwise Multiple Comparison of Mean Ranks Package (PMCMR) [Pohlert, 2014].

For the SVM and the BPM implementations, we consider the Sequential Minimal Optimization [Platt, 1998] with hard margin and the implementation given by [Herbrich *et al.*, 2001], respectively. All methods consider the use of a linear kernel, since the datasets analyzed are linearly separable.

Considering the defined oracle, EBPd, all parameters, as the number of components and the dissimilarity factor, were taken from [Enes *et al.*, 2015]. In order to provide a clear information about these parameters, we set the number of components as 10+1 to avoid ties due to the majority voting strategy. The dissimilarity factor was defined as the highest possible value in such way that the algorithm is able to generate all required components. Table 2 summarizes the dissimilarity factors.

Table 2: Dissimilarity Factors (DF) for EPBd

Set	DF
Prostate	10
Breast	70
Colon	11
Leukemia	97
DLBCL	150
CNS	125

5.3 Numerical Results

Initially, to verify the correctness of the proposed method, we generated several oracles (ensembles) with only 1 component. We observed that the algorithm was able to converge to the point represented by the oracle, in all cases. Therefore, we could conclude that the method was able to choose the half-space which agrees with the oracle properly.

The following experimental study of VSRM was conducted on the linearly separable datasets presented earlier. A comparison was made among VSRM, SVM and BPM methods, aiming to identify if the proposed solution is a better approximation of the center of mass of the version space, by analyzing the generalization error. Table 3 shows the mean classification error and the standard deviation at the testing phase. The best results for each dataset are highlighted in bold.

Table 3: Comparison on datasets.

Set	SVM	BPM	VSRM
Prostate	9.58 ± 1.35	10.46 ± 1.57	$\mathbf{9.46 \pm 1.77}$
Breast	20.67 ± 2.49	20.35 ± 3.42	$\mathbf{19.83 \pm 2.39}$
Colon	18.69 ± 2.32	15.68 ± 2.35	$\mathbf{15.44 \pm 2.29}$
Leukemia	$\mathbf{2.75 \pm 0.88}$	5.50 ± 1.32	3.39 ± 1.30
DLBCL	3.81 ± 0.68	3.86 ± 0.79	$\mathbf{3.49 \pm 0.87}$
CNS	33.50 ± 1.99	33.33 ± 2.68	$\mathbf{32.87 \pm 2.15}$

From the reported results, we can observe that our method always outperforms BPM and overcome SVM in 5 of the 6 considered datasets. Also, we observe that the results of BPM are better than SVM, considering the generalization capacity, in 3 of the 6 considered datasets. Such result, according to [Watkin, 1993] and [Herbrich, 2001], indicates that our method provides a better approximation of the center of mass of the version space than the approximations given by SVM and BPM.

We performed the Friedman Rank Sum test which assumes, as null hypothesis, the equivalence of the generalization capacity of the 3 algorithms analyzed. Friedman test indicates significance ($\chi^2 = 6.3333$, $df = 2$, p-value = 0.04214), which allows us to reject the null hypothesis, since we obtained relevance at 5%. After, we applied the post-hoc Nemenyi test, which provides pairwise comparisons. From this test we concluded that the proposed VSRM differ significant from SVM and BPM at 10%, with p-value = 0.955 and p-value = 0.055 respectively.

6 Discussion

In this work, we proposed the VSRM, a novel method to estimate the center of mass of the version space. We define the EBPd as an oracle for the VSRM due to its kernel implementation and the diversity of the components provided by this method. From the reported experimental evaluation, we observed that the results obtained by our method were very relevant. In all analyzed cases, our method wase able to provide a better approximation of the center of mass than the BPM, considering the ideias proposed by [Watkin, 1993] and [Opper and Kinzel, 1996] according to the generalization capacity. Moreover, considering the approximation given by SVM, our method outperforms the SVM solution on 5 of the 6 datasets. Statistical tests show that our results are relevant according to both SVM and BPM.

As future work, we intend to apply our method to non-linearly separable datasets, extending its formulation to the kernel space. Also, we intend to investigate other oracles, in order to obtain a better distribution of the components of the ensemble in the version space.

Acknowledgments

The authors would like to thank CAPES and CNPq for the financial support.

References

[Bishop, 1995] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[Boser *et al.*, 1992] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[Demsar, 2006] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 2006.

[Enes *et al.*, 2015] Karen Braga Enes, Saulo Moraes Villela, and Raul Fonseca Neto. A novel ensemble approach based on balanced perceptrons applied to microarray datasets. In *Proceedings of the 4th Brazilian Conference on Intelligent Systems*, 2015.

[Glaab *et al.*, 2012] Enrico Glaab, Jaume Bacardit, Jonathan M Garibaldi, and Natalio Krasnogor. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. *PloS one*, 2012.

[Golub *et al.*, 1999] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 1999.

[Graepel *et al.*, 1999] Thore Graepel, Ralf Herbrich, and Klaus Obermayer. Bayesian transduction. In *NIPS*, volume 12, page I999, 1999.

[Herbrich *et al.*, 2001] Ralf Herbrich, Thore Graepel, and Colin Campbell. Bayes point machines. *The Journal of Machine Learning Research*, 2001.

[Herbrich, 2001] Ralf Herbrich. *Learning Kernel Classifiers: theory and algorithms*. The MIT Press, 2001.

[Kohavi, 1995] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th IJCAI*. Morgan Kaufmann Publishers Inc., 1995.

[Lasserre, 1983] Jean B Lasserre. An analytical expression and an algorithm for the volume of a convex polyhedron in rn. *Journal of optimization theory and applications*, 39(3):363–377, 1983.

[Leite and Fonseca Neto, 2008] Saul de Castro Leite and Raul Fonseca Neto. Incremental margin algorithm for large margin classifiers. *Neurocomputing*, 71:1550–1560, 2008.

[Maire, 2003] Frederic Maire. An algorithm for the exact computation of the centroid of higher dimensional polyhedra and its application to kernel machines. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 605–608. IEEE, 2003.

[Michalski *et al.*, 2013] Ryszard S Michalski, Jaime G Carbonell, and Tom M Mitchell. *Machine Learning: an artificial intelligence approach*. Springer Science & Business Media, 2013.

[Opper and Kinzel, 1996] Manfred Opper and Wolfgang Kinzel. Statistical mechanics of generalization. In *Models of neural networks III*, pages 151–209. Springer, 1996.

[Platt, 1998] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research, April 1998.

[Pohlert, 2014] Thorsten Pohlert. *The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR)*, 2014. R package.

[Rosenblatt, 1958] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958.

[Ruján, 1997] Pál Ruján. Playing billiards in version space. *Neural Computation*, 9(1):99–122, 1997.

[Tong and Koller, 2002] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[Trafalis and Malyscheff, 2002] Theodore B Trafalis and Alexander M Malyscheff. An analytic center machine. *Machine Learning*, 46(1-3):203–223, 2002.

[Vapnik, 1995] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[Watkin, 1993] TLH Watkin. Optimal learning with a neural network. *EPL (Europhysics Letters)*, 21(8):871, 1993.

[Zhu *et al.*, 2007] Zexuan Zhu, Yew-Soon Ong, and Manoranjan Dash. Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition*, 2007.