Pattern Anal Applic (2019) 22:573–582
https://doi.org/10.1007/s10044-017-0654-3

THEORETICAL ADVANCES

SVM with a neutral class

Marek Śmieja[1] · Jacek Tabor[1] · Przemyslaw Spurek[1]

Received: 31 October 2016 / Accepted: 26 September 2017 / Published online: 7 October 2017

Abstract In many real binary classification problems, in addition to the presence of positive and negative classes, we are also given the examples of third neutral class, i.e., the examples with uncertain or intermediate state between positive and negative. Although it is a common practice to ignore the neutral class in a learning process, its appropriate use can lead to the improvement in classification accuracy. In this paper, to include neutral examples in a training stage, we adapt two variants of Tri-Class SVM (proposed by Angulo et al. in Neural Process Lett 23(1):89–101, 2006), the method designed to solve three-class problems with a use of single learning model. In analogy to classical SVM, we look for such a hyperplane, which maximizes the margin between positive and negative instances and which is localized as close to the neutral class as possible. In addition to original Angulo's paper, we give a new interpretation of the model and show that it can be easily implemented in the primal. Our experiments demonstrate that considered methods obtain better results in binary classification problems than classical SVM and semi-supervised SVM.

Keywords Classification · SVM · Semi-supervised learning · Cheminformatics

1 Introduction

One of the machine learning paradigms states that one should take into account all existing information in building a learning framework. For instance, in the semi-supervised learning, the classifier is allowed to use unlabeled data from underlying classes for improving its classification accuracy [5, 28]. In universum learning, we might use unlabeled data samples that do not belong to either classes [29, 32]. Integrating pre-defined additional information into a learning framework would usually yield improvement in the classification results and obtaining better insight into data.

In this paper, we support the above hypothesis and show that neutral instances can be easily handled with a use of Tri-Class SVM model [2]. Our motivation of including neutral class in a training comes from cheminformatics and computer-aided drug design, in which we focus on detecting compounds acting on a particular protein (biological receptor). A compound is considered active if its binding constant $K_i \in [0, +\infty)$ (measured in a laboratory) is lower than a threshold $a = 10^2$, while for inactive compounds a binding constant must be greater than $b = 10^3$[1] [31]. Consequently, we get a third class of compounds with an intermediate activity level such that $K_i \in [10^2, 10^3]$, which forms a neutral class. Although it is a common practice to ignore this neutral class in the learning process [26], we show that its use allows to explore the chemical space better, see Fig. 1.

Tri-Class SVM [2] is a generalization of classical SVM [8], which builds a single learning model for three-class problems and avoids pairwise coupling strategy. To use instances of neutral class in the learning process, we develop its two parameterizations: $\text{SVM}_{\{0\}}$ and $\text{SVM}_{[-1,1]}$. In analogy to classical SVM, we look for such a hyperplane which maximizes the margin between positive and negative examples

✉ Marek Śmieja
marek.smieja@ii.uj.edu.pl

[1] Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland

[1] One of the reasons why we put $b >> a$ is that the laboratory measurements might be very imprecise and we do not want to create drugs which act only on a selected group of patients.

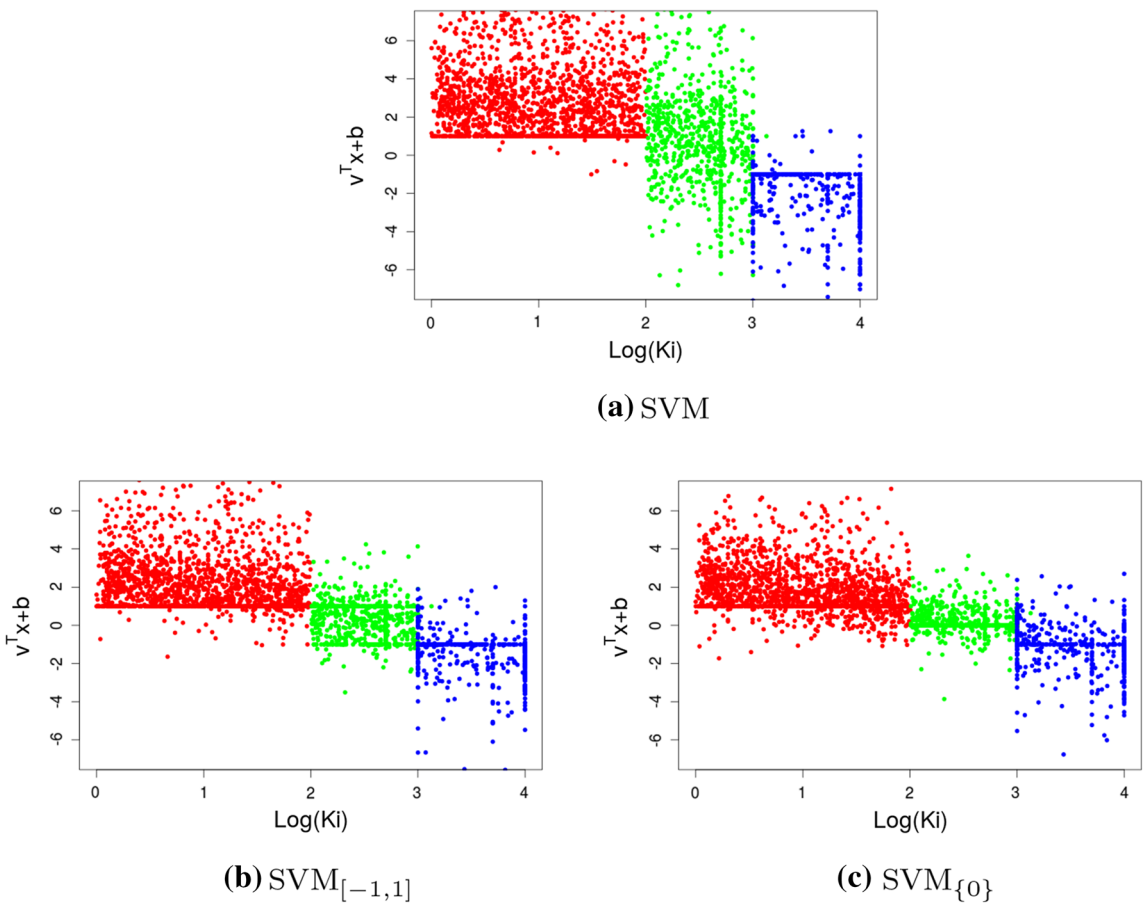


Fig. 1 Separation of active (red), middle active (green) and inactive compounds (blue) by classical SVM and our two variants of Tri-Class SVM. While SVM ignores completely the presence of neutral class, $SVM_{[-1,1]}$ and $SVM_{\{0\}}$ try to arrange it within the margin or directly on a decision boundary, respectively (color figure online)

and is localized as close to the neutral class as possible. The difference between introduced methods stems from the way of penalizing the model for inappropriate classification of instances of neutral class: $SVM_{\{0\}}$ aims at fitting the hyperplane along the neutral set, while $SVM_{[-1,1]}$ allows the neutral class to "move" freely in the whole space between the positive and negative classes, see Fig. 2 for a comparison between these methods and two classical approaches, SVM and S3VM (semi-supervised SVM). Contrary to the original formulation of Tri-Class SVM, we show that both models can be easily optimized and implemented in the primal: to find the solution of $SVM_{\{0\}}$ one can use subgradient approach, while $SVM_{[-1,1]}$ fits perfectly into the classical SVM procedure if we slightly modify a considered dataset[2], see Theorem 1.

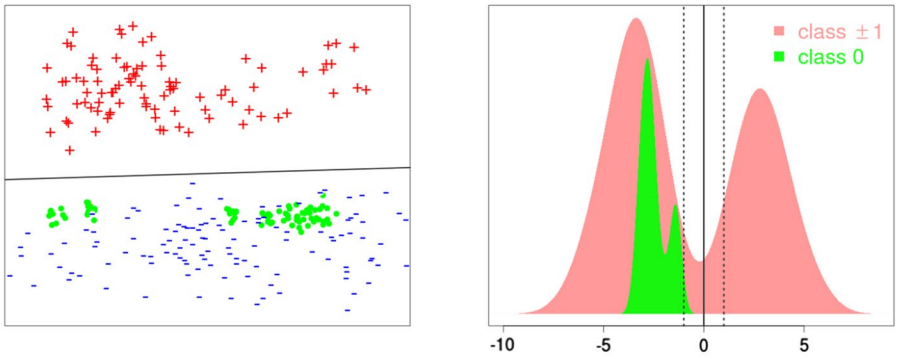
We showed experimentally that $SVM_{[-1,1]}$ usually leads to the improvement in the accuracy of binary classification given by classical SVM and S3VM, when an adequate sample of instances of neutral class is available. Moreover, the experimental study demonstrated that $SVM_{\{0\}}$ is able to explore less common patterns of data. In particular, we showed that a decision boundary constructed for ligands of one biological target (classification problem) delivers a substantial knowledge concerning other proteins (other classification problem), which could have practical consequences in cheminformatics and computer-aided drug design.

The paper is organized as follows. Next section compares our model with related methods. Section 3 presents the theory behind our model. In fourth section, we present the results of the experiments. Finally, a conclusion is formulated.

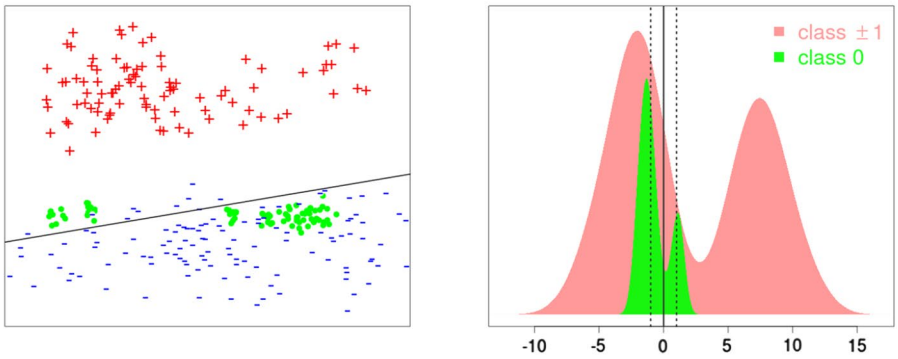
2 Related work

Neutral class usually appears in topics concerning natural language processing such as sentiment analysis or opinion mining [1], but it is also present in chemistry, medicine [12],

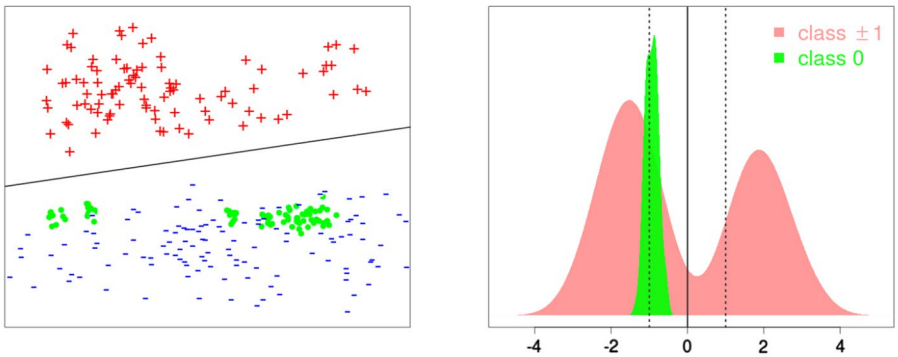
[2] We can simply double the examples of the neutral class and add them to positive as well as to negative class.



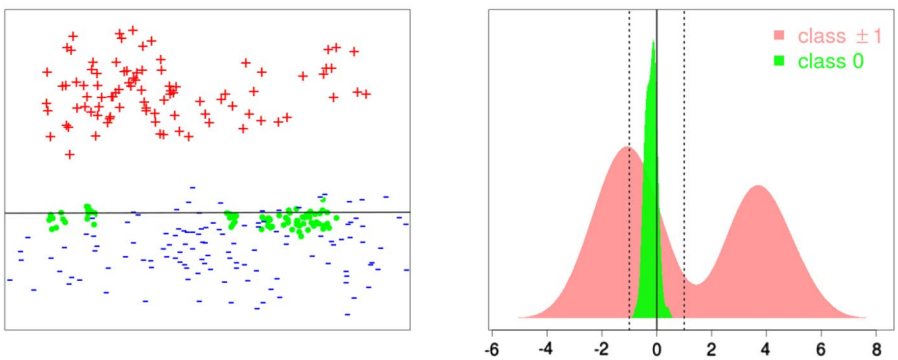
(a) SVM



(b) S3VM



(c) $\mathrm{SVM}_{[-1,1]}$



(d) $\mathrm{SVM}_{\{0\}}$

Fig. 2 Decision boundaries and corresponding densities estimated from positive and negative classes (red) compared with densities estimated from neutral class (green) after projecting onto vector normal to the decision boundary. $\mathrm{SVM}_{[-1,1]}$ fits such a decision boundary (solid line) to separate instances of positive and negative classes and to keep examples of neutral class within the margin (dotted lines) 1(**c**). It gives slightly similar effect to classical SVM, which, however, ignores the presence of neutral class 1(**a**). $\mathrm{SVM}_{\{0\}}$, in addition to separating positive from negative class, tries to build a decision boundary along the neutral class 1(**d**) which in turn is similar to the results produced by S3VM 2(**b**) (color figure online)

etc. Although the authors are aware of its importance, many of them ignore it and do not take it into account in both training and testing models [14, 34]. Clearly, this destroys a dataset since a particular group of instances are removed from a data space. Moreover, the removal of neutral class contradicts the well-known paradigm of machine learning which says that all available information should be used.

Another approach relies on using typical multi-class classifiers to handle neutral examples [24, 30]. Unfortunately, this methodology does not use internal relation between positive/negative and neutral classes. Moreover, the use of SVM in multi-class problem requires the construction of 3 base classifiers, which increases the complexity of the model [9]. Koppel and Schler [20, 21] showed that classical pairwise coupling methods do not work well with neutral class. Instead of selecting a class label based on majority voting in one-vs-one SVM, they proposed to use a stack, which allows for making a decision based on the ordering of support functions of base classifiers. There is also an extension of this strategy, where two binary classifiers (positive/non-positive, and negative/nonnegative) are trained, which corresponds to one-vs-all approach[3]. The authors of [33] use a hierarchical classification system, where the neutrality is determined first and the polarity is determined second.

In the context of sentiment analysis, Xia et al. [35] created a method, in which a classifier learns from pairs of sentiment-reversed reviews. Although the authors showed high performance of this technique, it is difficult to generalize their algorithm to other domains, because it requires the generation of opposite reversed reviews with opposite sentiments. To tackle a classification problem, where classes have specific ordering, ranking methods are also used [27]. This approach can be used for more than three classes, but its optimization is computationally hard in a comparison with typical classification models [10].

Including neutral examples to learning process is significantly different reasoning from the one used by semi-supervised SVM (S3VM), in which the unlabeled elements are considered as the instances of positive or negative class [18], see Fig. 2. To find a solution of S3VM problem, a lot of approximation schemes were designed [23, 25, 32]; however, most implementations still suffer from high computational cost.

Presented model is an adaptation of Tri-Class SVM proposed by Angulo et al. [2], which deals with general three-class problems by encapsulating a third class in a δ-tube (the area with a width δ along the separation hyperplane). We extended the above strategy to include the instances of neutral class directly on a decision boundary in the case of $\mathrm{SVM}_{\{0\}}$ or within the margin in the case of $\mathrm{SVM}_{[-1,1]}$. Combining the ideas from universum learning [6, 7], we present that this formulation suits well into the problem of learning with neutral class. In addition to Tri-Class SVM, we show that both considered models can be easily and efficiently optimized in the primal.

Analogical approach to $\mathrm{SVM}_{\{0\}}$ could also be applied to classifiers related to SVM. In the case of logistic regression, one could maximize the probability that neutral examples are equally likely to belong to both positive and negative classes, i.e., its posterior probability equals 0.5. Transforming $\mathrm{SVM}_{[-1,1]}$ to the case of logistic regression could be harder, because there is no margin in classical formulation of logistic regression.

3 Theoretical model

For a convenience of the reader, we start with a formulation of classical SVM and next motivate the construction of cost functions for $\mathrm{SVM}_{\{0\}}$ and $\mathrm{SVM}_{[-1,1]}$. After that we discuss their relation with Tri-Class SVM and present optimization procedures used in the paper.

Let us recall that SVM [4, Chapter 2.3] aims at finding such an affine function $x \to v^T x + b$ which minimizes the cost function given by

$$\begin{aligned}\mathrm{SVM}(v,b) = \tfrac{1}{2}\|v\|^2 &+ C \sum_{y_i=-1} \max(0, 1 + (v^T x_i + b)) \\ &+ C \sum_{y_i=1} \max(0, 1 - (v^T x_i + b)),\end{aligned} \tag{1}$$

where $X = (x_i)_i$ is a dataset and $y_i = \pm 1$ denotes the class membership of x_i. The first term $\frac{1}{2}\|v\|^2$ plays the regularization role, while the expression $\max(0, 1 - y_i(v^T x_i + b))$ measures a distance of the point $v^T x_i + b$ from the set $[1, +\infty)$, for $y_i = +1$ (or from $(-\infty, -1]$, for $y_i = -1$). Thus, we may rewrite the above formula in the form

$$\begin{aligned}\mathrm{SVM}(v,b) = \tfrac{1}{2}\|v\|^2 &+ C \sum_{y_i=-1} \mathrm{dist}(v^T x_i + b; (-\infty, -1]) \\ &+ C \sum_{y_i=1} \mathrm{dist}(v^T x_i + b; [1, \infty)),\end{aligned}$$

where the last two terms introduce a penalty for inappropriate classification. The final classification of point x is based on the sign of $v^T x + b$.

To define our model, we need to introduce the instances of additional neutral class to a dataset X. By the realizations of neutral class, we understand the elements with an intermediate state between positive and negative states. As an example one can consider a group of patients, who are diagnosed to be in the early stage of illness. In our model, we base on the observation that instances of a neutral class should lay somewhere in the middle between positive and negative classes. Clearly, this assumption may not be true in a given representation, and then

[3] https://lingpipe-blog.com/2008/01/02/positive-negative-and-neutral-sentiment/.

the application of some kernel functions is needed. Nevertheless, this issue will not be investigated in this paper. We put $y = 0$ to denote the label of elements of neutral class.

The expression (1) allows to formulate two natural additions to the SVM cost function in the case we are given a neutral class:

- we can penalize a point x from the neutral class by a distance of $v^T x + b$ from zero; in this case as the additional cost we put $|v^T x_i + b|$,
- we can penalize a point x from the neutral class by a distance of $v^T x + b$ from the interval $[-1, 1]$; in this case the additional cost equals $\text{dist}(v^T x_i + b, [-1, 1])$.

One can easily observe that

$$\begin{aligned}\phi(r) &:= \text{dist}(r, [-1, 1]) \\ &= \max(0, -r - 1) + \max(0, r - 1) \\ &= \max(0, r + 1) + \max(0, 1 - r) - 2.\end{aligned} \tag{2}$$

Thus, we obtain two models, which will be referred as $\text{SVM}_{\{0\}}$ and $\text{SVM}_{[-1,1]}$, with the cost functions given by

$$\text{SVM}_{\{0\}}(v, b) = \text{SVM}(v, b) + C \sum_{i:y_i=0} \left| v^T x_i + b \right|,$$

$$\text{SVM}_{[-1,1]}(v, b) = \text{SVM}(v, b) + C \sum_{i:y_i=0} \phi\left(v^T x_i + b\right),$$

where $\text{SVM}(v, b)$ is formulated by (1) and $\phi(r)$ denotes a distance of point r from the set $[-1, 1]$ (2). Observe that $\text{SVM}_{\{0\}}$ wants to fit the barrier along the neutral set, while $\text{SVM}_{[-1,1]}$ allows the neutral class to "move" freely in the whole space between the positive and negative classes, see Fig. 2.

Both models are variants of general Tri-Class SVM that allows to deal with three-class problems by building a single SVM machine. $\text{SVM}_{\{0\}}$ corresponds to $\delta = 0$ in [2, eq. 12], while $\text{SVM}_{[-1,1]}$ is parameterized by $\delta = 1$. We show that our models can be easily implemented in the primal, which is different from a typical way of realizing Tri-Class SVM.

Remark 1 In practice, there might occur a problem of imbalanced classes. If the size of neutral class is significantly greater (or smaller) than the remaining data, our model will fit stronger to this class. To reduce this negative effect, one could introduce an additional parameter $D > 0$, which varies the importance of neutral class. Then, the above cost functions are given by

$$\text{SVM}_{\{0\}}(v, b) = D \cdot \text{SVM}(v, b) + C \sum_{i:y_i=0} \left| v^T x_i + b \right|,$$

$$\text{SVM}_{[-1,1]}(v, b) = D \cdot \text{SVM}(v, b) + C \sum_{i:y_i=0} \phi\left(v^T x_i + b\right).$$

This is an analogical strategy to dealing with data imbalance to the one used in classical SVM, where parameter C for positive and negative classes is scaled by the ratios of respective classes [16].

Remark 2 Tri-Class SVM and our model assume that the examples of neutral class are localized close to the decision boundary between positive and negative classes. However, this assumption may not hold for a given data representation and the neutral samples can overlap with both positive and negative classes, which could drop the performance of the learning system. One way to deal with this problem is to decrease the importance of neutral class as described in previous remark.

Another way for resolving this issue relies on using kernel functions. The correct selection of kernel mapping allows for transforming data to another space, where the instances of neutral class lay in the middle between positive and negative examples and, in consequence, classes are linearly separable. The reader is referred to [2] for details of kernel approach for Tri-Class SVM.

$\text{SVM}_{\{0\}}$ can be solved by using a gradient[4] approach. As one can verify the gradients of $\text{SVM}_{\{0\}}$ cost function with respect to v and b are given by

$$\begin{aligned}\nabla \text{SVM}_{\{0\}}(v, b) = \begin{bmatrix} v \\ 0 \end{bmatrix} &+ C \sum_{i:y_i=-1} H(1 + (v^T x_i + b)) \begin{bmatrix} x_i \\ 1 \end{bmatrix} \\ &+ C \sum_{i:y_i=0} \text{sign}(v^T x_i + b) \begin{bmatrix} x_i \\ 1 \end{bmatrix} \\ &- C \sum_{i:y_i=1} H(1 - (v^T x_i + b)) \begin{bmatrix} x_i \\ 1 \end{bmatrix}\end{aligned}$$

where H denotes the Heaviside function. The above formula allows the easy implementation of $\text{SVM}_{\{0\}}$ in any package, which contains the gradient descent method.

Now we are going to show that $\text{SVM}_{[-1,1]}$ can be used with existing SVM software. To do so, we have to just add the instances of neutral class both for the positive and negative classes. This observation is proven in the following theorem:

Theorem 1 *Let $X_{-1,0,1}$ denotes the sequence of elements of the respective classes. Then the following two functions are equal:*

- $\text{SVM}_{[-1,1]}\text{cost}(v, b)$, *for the data* $X_{-1,0,1}$,
- $-2C \cdot \text{card}(X_0) + \text{SVM}(v, b)$, *for the data with positive class* $X_0 \cup X_{+1}$ *and negative class* $X_{-1} \cup X_0$.

[4] More precisely, a subgradient method.

Proof Clearly, $\mathrm{SVM}_{[-1,1]}$ cost function for the data $X_{-1,0,1}$ with the constant C equals

$$\begin{aligned}\frac{1}{2}\left\|v^2\right\| &+ C\sum_{i:y_i=-1}\max(0,1+(v^Tx_i+b))\\ &+ C\sum_{i:y_i=1}\max(0,1-(v^Tx_i+b))\\ &+ C\sum_{i:y_i=0}\phi(v^Tx_i+b),\end{aligned}\tag{3}$$

where by (2),

$$\phi(r)=\max(0,r+1)+\max(0,1-r)-2.$$

On the other hand, SVM cost for the data with a positive class $(X_{+1}\cup X_0)$ and a negative one $(X_{-1}\cup X_0)$ is given by

$$\begin{aligned}\frac{1}{2}\left\|v^2\right\| &+ C\sum_{i:y_i=-1}\max(0,1+(v^Tx_i+b)\\ &+ C\sum_{i:y_i=0}\max(0,1+(v^Tx_i+b))\\ &+ C\sum_{i:y_i=1}\max(0,1-(v^Tx_i+b))\\ &+ C\sum_{i:y_i=0}\max(0,1-(v^Tx_i+b)).\end{aligned}\tag{4}$$

Let us denote by $\psi(r)$ the following function:

$$\psi(r)=\phi(r)-\max(0,1+r)-\max(0,1-r).$$

By (2), we get that ψ is a constant function such that $\psi(r)=2$. Then, the difference between (3) and (4) equals

$$\begin{aligned}&C\sum_{i:y_i=0}\phi(v^Tx_i+b)\\ &-C\sum_{i:y_i=0}\max(0,1+(v^Tx_i+b))\\ &-C\sum_{i:y_i=0}\max(0,1-(v^Tx_i+b))\\ &=C\sum_{i:y_i=0}\psi(v^Tx_i+b)\\ &=C\sum_{i:y_i=0}(-2)=-2C\cdot\mathrm{card}(X_0),\end{aligned}$$

which completes the proof. □

Table 1 Summary of data used in the experiments

Dataset	X_{-1}	X_0	X_{+1}	# features
Heart disease	164	91	48	13
Housing	277	108	121	13
5-HT1a	1057	1486	3575	79/1024[a]
5-HT6	351	456	1363	79/1024[a]

[a]We consider two representations of chemical compounds: estate consists of 79 attributes, while extended contains 1024 features

Observe, that by the above theorem we can reduce the problem of minimizing of the cost function for $\mathrm{SVM}_{[-1,1]}$ to the problem of minimization of SVM for slightly modified dataset. Namely, we double the examples of the neutral class and add them to positive as well as to negative class.

4 Experiments

We evaluated our methods on several classification problems and compare the results with related methods. We used examples retrieved from UCI repository [3] and real datasets of chemical compounds [13].

All experiments were performed with a use of double fivefold cross-validation. In this approach, we randomly partitioned a dataset into five equally sized subsets. Then, a single subset was retained as test data while the remaining four subsets were used in training. This process was repeated five times—each of five subsamples was used exactly once as the test data, and the results were averaged. To tune hyperparameter C, we applied analogical procedure on each training set: it was again divided into five parts, where one was used as validation set, while other four parts were used in training. We checked the range $C\in\{0.1,1,10,100\}$ and choose the this value of C, which provided the best average score reported on validation set to train a final classifier.

4.1 Binary classification of UCI datasets

First, we have evaluated the proposed methods in binary classification task. For this purpose, two datasets from UCI repository were selected. The first one, *Heart Disease*, refers to the presence of heart disease in the patients. The chance of illness was quantified by an integer value ranging from 0 to 4. We identified a negative class by a number 0 (no disease) while the positive class was linked with numbers 3 and 4 (high level of disease). For a neutral class, we used intermediate values 1 and 2. The second dataset, *Housing*, concerns housing values in suburbs of Boston. The prices lower than 220,000$ were linked with a negative class, the prices

Table 2 MCC scores reported on test sets for binary classification task

Dataset	SVM	S3VM	$SVM_{\{0\}}$	$SVM_{[-1,1]}$
Heart disease	0.75 ± 0.02	$\mathbf{0.80 \pm 0.01}$	0.78 ± 0.02	$\mathbf{0.80 \pm 0.01}$
Housing	0.85 ± 0.02	0.83 ± 0.02	$\mathbf{0.87 \pm 0.01}$	$\mathbf{0.87 \pm 0.04}$
5-HT1a (Ext)	0.59 ± 0.02	0.59 ± 0.01	0.58 ± 0.01	$\mathbf{0.62 \pm 0.02}$
5-HT6 (Ext)	$\mathbf{0.77 \pm 0.02}$	0.74 ± 0.02	0.75 ± 0.01	$\mathbf{0.77 \pm 0.01}$

Bold values indicate the best result for each data set

greater than 260,000$ denoted a positive class, while the neutral class covered rest of values[5], see Table 1 for details.

We investigated whether the presence of neutral class could help to obtain a better binary prediction. The classifier was trained on a dataset containing instances of positive, negative and neutral class and then tested on the set of examples of positive and negative classes only. We compared the results returned by $SVM_{\{0\}}$ and $SVM_{[-1,1]}$ with classical SVM, which ignores the neutral class and with S3VM, which treats the examples of neutral class as unlabeled data (both implemented in SVM^{light} [17]).

We reported the mean value of Matthews Correlation Coefficient (MCC), which illustrates a type of correlation between prediction and ground truth [11]. It ranges from -1 to 1; the values ± 1 mean perfect positive or negative correlation, respectively, while 0 denotes no correlation. The main reason for choosing MCC, instead of classical accuracy, was the fact that MCC is also a good measure for imbalanced datasets.

It is evident from the results placed in Table 2 (first two rows) that the introduction of the neutral class improved the performance of SVM. Moreover, our methods outperformed S3VM in the case of *Housing* dataset, which means that it is also important to identify the neutral class, not only to include additional examples to the training process. This experiment suggested that the strategy of incorporating the neutral class used by $SVM_{[-1,1]}$ is more profitable than the one applied by $SVM_{\{0\}}$.

4.2 Detection of active compounds

To investigate deeper the influence of the introduction of neutral class on the performance of binary classification, we considered two real datasets of chemical compounds. Before presenting the results, let us first describe the problem from chemical point of view. Chemical compounds are often represented as fingerprints, i.e., binary sequences which encode their selected structural features. Since different features can be taken into account, then a multitude of fingerprints were introduced. In the present study, we used Extended fingerprint (Ext), which consists of 1024 bits and is considered as one of the most powerful representations [36].

The task undertaken in this experiment concerned the identification of compounds acting on two biological receptors 5-HT1a and 5-HT6, the proteins responsible for the regulation of central nervous system [22]. Compounds classified by a learning system as active in virtual screening process are usually further examined, and the most promising ones could be used in drug designing. The activity level is measured by a positive real valued number K_i: if $K_i \leq 100$, then a compound is active, $K_i > 1000$ describes inactive compounds, while the compounds with $100 < K_i \leq 1000$ are not classified to any of these groups and they are usually eliminated from a training stage. Table 1 presents details about chemical datasets.

In this experiment, we tested whether the introduction of compounds with intermediate activity levels allows to obtain better classification results. The experiment was conducted in the same manner as in previous subsection. The results presented in Table 2 (last two rows) show that $SVM_{[-1,1]}$ performed better than $SVM_{\{0\}}$ in the case of high-dimensional binary data. Moreover, $SVM_{[-1,1]}$ also gave higher MCC scores than SVM and S3VM for both datasets.

4.3 Chemical space exploration

As mentioned in previous subsection, compounds acting on a given biological receptor could be used in drug construction. However, in practice drug should act only on a single receptor. If a compound activates more than one target, then it often causes side effects. Therefore, we aim at finding such compounds which are active on one receptor and simultaneously are inactive on the other.

In this experiment, we would like to check out whether a decision boundary constructed for one biological target allows to separate compounds with respect to their activity on other target as well. More precisely, we trained a classifier making use of actives, inactives and compounds with intermediate activity for one receptor and then test the performance of constructed decision boundary in separating active and inactive compounds with respect to the second receptor. In this experiment, we included one more fingerprint, Estate fingerprint (Est), which contains only 79 bits and is considered as a basic fingerprint representation [15].

The results presented in Fig. 3 show that decision boundaries obtained from classical SVM and $SVM_{[-1,1]}$ for one receptor do not provide any significant information about the activity with respect to the second protein. The interesting thing is that such a substantial knowledge can be explored by $SVM_{\{0\}}$. Negative MCC scores indicate that there is a negative correlation between predictions and ground truth. In other words, the compounds acting on the second receptor are located on the same side of decision boundary

[5] We also considered different thresholds for defining neutral class, but the results were similar the those presented in this paper.

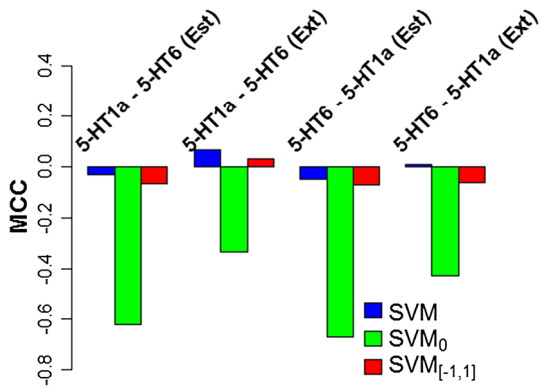


Fig. 3 MCC scores in the case when the classifier was trained on one receptor and tested on the other

Table 3 Accuracies of three-class classification of chemical compounds datasets

Dataset	One-vs-one	One-vs-all	$SVM_{\{0\}}$	$SVM_{[-1,1]}$
HT1a (Est)	0.62 ± 0.02	0.49 ± 0.01	0.62 ± 0.02	$\mathbf{0.63 \pm 0.02}$
HT1a (Ext)	$\mathbf{0.67 \pm 0.02}$	0.62 ± 0.01	0.62 ± 0.026	0.65 ± 0.02
HT6 (Est)	$\mathbf{0.68 \pm 0.02}$	0.62 ± 0.01	0.63 ± 0.02	$\mathbf{0.68 \pm 0.02}$
HT6 (Ext)	$\mathbf{0.77 \pm 0.02}$	0.75 ± 0.02	0.63 ± 0.02	0.76 ± 0.01

Bold values indicate the best result for each data set

constructed by $SVM_{\{0\}}$ as the compounds inactive on the first receptor and conversely. Consequently, we found that the introduction of neutral class allowed to explore larger region of chemical space.

4.4 Three-class classification

Since both $SVM_{\{0\}}$ and $SVM_{[-1,1]}$ learn from the examples of three classes, we investigated their capabilities in 3-class classification problems. For simplicity, we assumed the following classification rule[6] for an instance x:

- if $v^T x + b > \frac{2}{3}$ then class $(x) = +1$.
- if $v^T x + b < -\frac{2}{3}$ then class $(x) = -1$.
- otherwise, class $(x) = 0$.

Proposed approaches were compared with *one-vs-one* and *one-vs-all* variants of classical SVM.

We considered two datasets of chemical compounds from previous subsections in Extended fingerprint and Estate fingerprint representations. The goal was to predict actives, inactives and compounds with intermediate activity. Since we are dealing with multi-class problem, the results were measured by the accuracy, which is well defined for any number of classes [11].

Table 4 Relative number of correctly ordered elements of three-class problem

Dataset	SVM-rank	$SVM_{\{0\}}$	$SVM_{[-1,1]}$
HT1a (Ext)	0.63 ± 0.02	0.62 ± 0.01	$\mathbf{0.66 \pm 0.02}$
HT6 (Ext)	0.82 ± 0.02	0.83 ± 0.02	$\mathbf{0.87 \pm 0.01}$

Bold values indicate the best result for each data set

The results placed in Table 3 show that $SVM_{[-1,1]}$ gave comparable accuracy to one-vs-one SVM strategy. On the other hand, both proposed methods outperformed one-vs-all variant which occurred non-adequate in this example of data. It is worth to mention that $SVM_{[-1,1]}$ and $SVM_{\{0\}}$ build a single classification model while comparative approaches contain three different base SVM classifiers.

4.5 Comparison with SVM-rank

One can say that the proposed methods try to order the instances of underlying three classes along the vector normal to the decision boundary. In particular, if any disagreement occurs then the model is penalized[7]. This is similar to the reasoning used in ranking methods. Clearly, ranking tools have much wider applicability and allow to learn from any ranking, not only from ranking generated by 3-class problem. However, they are usually slow because all possible pairwise relations are considered.

To compare our methods with SVM-rank [19] in preserving the order generated by 3-class problem, we assumed that any instance from negative class precedes examples of neutral class which in turn precede elements of positive class. We assumed that elements of the same class are not comparable. To measure the ranking performance, we count the number of comparable pairs, which lie in the correct order after classification and normalize it by the total number comparable pairs. This index which we call Rank-acc, can be seen as ranking accuracy.

The results presented in Table 4 show that the highest number of correctly ordered pairs was obtained by $SVM_{[-1,1]}$. As mentioned $SVM_{[-1,1]}$ tries to keep instances of every class within disjoint regions of the space. Therefore, every disagreement is automatically penalized by the model. On the other hand, the performance of $SVM_{\{0\}}$ was comparable to SVM-rank. Let us observe in Fig. 4 that SVM-rank tried to find such a vector (normal to decision

[6] One could also find an optimal threshold in a cross-validation procedure.

[7] Clearly, a penalty can be also given if the ordering along the normal subspace agrees, but instances are not localized within assumed margins.

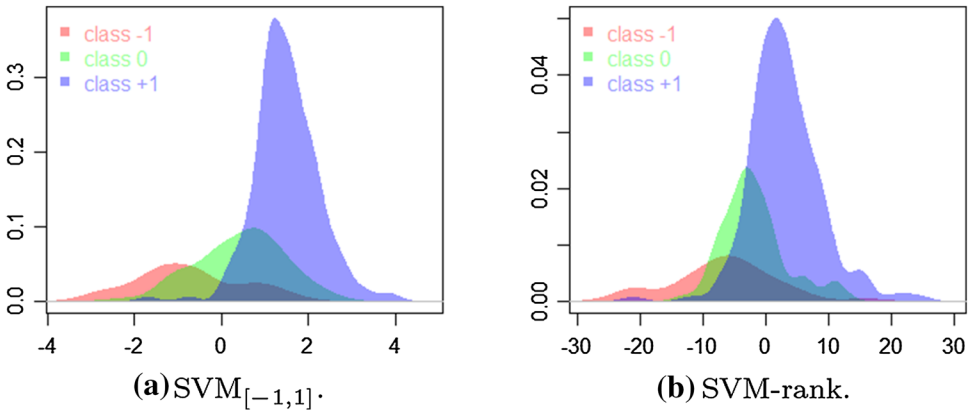


Fig. 4 Density plots of underlying three classes 4(**a**, **b**)

boundary) which allows to arrange (project) data in a wide range of one-dimensional subspace. This is characteristic to ranking methods. Although $SVM_{[-1,1]}$ projected data onto eight times lower range, its specialization to 3-class problems provided higher rate of ordering.

5 Conclusion

In this paper, we discussed two versions of Tri-Class SVM to take into account the information contained in additional neutral class. Although both methods add a penalty for an inappropriate classification of instances of neutral class, the difference lies in their understanding of miss-classification. $SVM_{\{0\}}$ uses more restrictive strategy and penalizes the model if an example of neutral class does not lie on a decision boundary, while in $SVM_{[-1,1]}$ we try to locate the elements of neutral class within the margin.

We examined proposed approaches in practical classification tasks. We showed that $SVM_{[-1,1]}$ can be useful in improving binary classification by including instances of the neutral class. The reasoning used in designing $SVM_{\{0\}}$ is different from a typical one used in most binary classifiers, as the neutral class can dominate the presence of positive and negative ones. The classifier is guided by the location of neutral class stronger than in the case of $SVM_{[-1,1]}$. This unusual strategy allows to explore less common regions of data and obtain surprising results. In particular, we demonstrated that a decision boundary created for one biological target of chemical compounds could be used to classify compounds characteristic for the other protein. Such behavior could be useful in detecting potential drug candidates.

Acknowledgements This work was partially supported by the National Science Centre (Poland) Grant Nos. 2016/21/D/ST6/00980 and 2015/19/B/ST6/01819 and 2015/19/D/ST6/01472.

References

1. Alm CO, Roth D, Sproat R (2005) Emotions from text: machine learning for text-based emotion prediction. In: Proceedings of the conference on human language technology and empirical methods in natural language processing. Association for Computational Linguistics, pp 579–586
2. Angulo C, Ruiz FJ, González L, Ortega JA (2006) Multi-classification by using tri-class SVM. Neural Process Lett 23(1):89–101
3. Asuncion A, Newman DJ (2007) UCI machine learning repository. http://www.ics.uci.edu/~mlearn/MLRepository.html
4. Bottou L, Chapelle O, DeCoste D, Weston J (2007) Large-scale kernel machines. MIT Press, Cambridge
5. Chapelle O, Schölkopf B, Zien A et al (2006) Semi-supervised learning. MIT Press, Cambridge
6. Chapelle O, Agarwal A, Sinz FH, Schölkopf B (2007) An analysis of inference with the universum. In: Advances in neural information processing systems, pp 1369–1376
7. Cherkassky V, Dhar S, Dai W (2011) Practical conditions for effectiveness of the universum learning. IEEE Trans Neural Netw 22(8):1241–1255
8. Cortes C, Vapnik V (1995) Mach Learn 20(3):273–297
9. Debnath R, Takahide N, Takahashi H (2004) A decision based one-against-one method for multi-class support vector machine. Pattern Anal Appl 7(2):164–175
10. Duh K (2008) Ranking vs. regression in machine translation evaluation. In: Proceedings of the third workshop on statistical machine translation. Association for Computational Linguistics, pp 191–194
11. Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874
12. Gabrielsen M, Kurczab R, Siwek A, Wolak M, Ravna AW, Kristiansen K, Kufareva I, Abagyan R, Nowak G, Chilmonczyk Z et al (2014) Identification of novel serotonin transporter compounds by virtual screening. J Chem Inf Modeling 54(3):933–943

13. Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, Light Y, McGlinchey S, Michalovich D, Al-Lazikani B et al (2012) ChEMBL: a large-scale bioactivity database for drug discovery. Nucl Acids Res 40(D1):D1100–D1107
14. Go A, Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford 1(2009):12
15. Hall LH, Kier LB (1995) Electrotopological state indices for atom types: a novel combination of electronic, topological, and valence state information. J Chem Inf Comput Sci 35(6):1039–1045
16. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
17. Joachims T (1999a) Making large scale svm learning practical. Tech. rep., Universität Dortmund
18. Joachims T (1999b) Transductive inference for text classification using support vector machines. In: Proceedings of the 16th international conference on machine learning, pp 200–209
19. Joachims T (2002) Optimizing search engines using clickthrough data. In: Proceedings of the 8th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 133–142
20. Koppel M, Schler J (2005) Using neutral examples for learning polarity. In: Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI), vol 2005, pp 1616–1616
21. Koppel M, Schler J (2006) The importance of neutral examples for learning sentiment. Comput Intell 22(2):100–109
22. McCorvy JD, Roth BL (2015) Structure and function of serotonin g protein-coupled receptors. Pharmacol Ther 150:129–142
23. Ogawa K, Suzuki Y, Takeuchi I (2013) Safe screening of non-support vectors in pathwise svm computation. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp 1382–1390
24. Sidorov G, Miranda-Jiménez S, Viveros-Jiménez F, Gelbukh A, Castro-Sánchez N, Velásquez F, Díaz-Rangel I, Suárez-Guerra S, Treviño A, Gordon J (2012) Empirical study of machine learning based approach for opinion mining in tweets. In: Mexican international conference on artificial intelligence. Springer, pp 1–14
25. Sindhwani V, Keerthi SS, Chapelle O (2006) Deterministic annealing for semi-supervised kernel machines. In: Proceedings of the 23rd international conference on machine learning. ACM, pp 841–848
26. Smieja M, Warszycki D (2016) Average information content maximization-a new approach for fingerprint hybridization and reduction. PloS ONE 11(1):e0146666
27. Snyder B, Barzilay R (2007) Multiple aspect ranking using the good grief algorithm. In: Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL), pp 300–307
28. Song Y, Zhang C, Lee J, Wang F, Xiang S, Zhang D (2009) Semi-supervised discriminative classification with application to tumorous tissues segmentation of MR brain images. Pattern Anal Appl 12(2):99–115
29. Vapnik V (2006) Estimation of dependences based on empirical data. Springer, Berlin
30. Vincent M, Winterstein G (2013) Argumentative insights from an opinion classification task on a french corpus. In: JSAI international symposium on artificial intelligence. Springer, pp 125–140
31. Warszycki D, Mordalski S, Kristiansen K, Kafel R, Sylte I, Chilmonczyk Z, Bojarski AJ (2013) A linear combination of pharmacophore hypotheses as a new tool in search of new active compounds-an application for 5-ht 1a receptor ligands. PloS ONE 8(12):e84510
32. Weston J, Collobert R, Sinz F, Bottou L, Vapnik V (2006) Inference with the universum. In: Proceedings of the 23rd international conference on machine learning. ACM, pp 1009–1016
33. Wilson T, Wiebe J, Hoffmann P (2009) Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. Comput. Linguist 35(3):399–433
34. Witek J, Smusz S, Rataj K, Mordalski S, Bojarski AJ (2014) An application of machine learning methods to structural interaction fingerprints a case study of kinase inhibitors. Bioorg Med Chem Lett 24(2):580–585
35. Xia R, Xu F, Zong C, Li Q, Qi Y, Li T (2015) Dual sentiment analysis: considering two sides of one review. IEEE Trans Knowl Data Eng 27(8):2120–2133
36. Yap CW (2011) PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 32(7):1466–1474