Solving Classification Problems By Knowledge Sets

Marcin Orchel

Abstract
We propose a novel theoretical model and a method for solving binary classification problems. First, we find knowledge sets in the input space by using estimated density functions. Then, we find the final solution outside knowledge sets. We derived bounds for classification error based on knowledge sets. We estimate knowledge sets from examples and find the solution by using support vector machines (SVM). We performed tests on various real world data sets, and we achieved similar generalization performance compared to SVM with significantly smaller number of support vectors.

Keywords: support vector machines, classification, density estimation, prior knowledge, prediction bands

1. Introduction
One of possibilities to improve generalization performance for classification problems is to incorporate additional knowledge, sometimes called prior knowledge. Various types of prior knowledge have been already incorporated to SVM. In [1], the authors distinguish two types of prior knowledge: knowledge about class invariance, and knowledge about the data. The first type includes for example knowledge about classification in regions of the input space, [2, 3, 4, 5], knowledge about class invariance during transformation of the input. The second type includes for example knowledge about unlabeled examples, imbalance of classes, quality of the data. In [2, 3], the authors proposed informally a concept of knowledge sets: as for example cubes supposed to belong to one of two categories; they concentrated on incorporating prior knowledge in the form of polyhedral knowledge sets. In this paper, instead of incorporating prior knowledge, we use a concept of knowledge sets to model a standard classification problem, based only on training examples. We can interpret a knowledge set as information about classification for a set of data points in the input space. A decision boundary is supposed to lie outside knowledge sets (in an uncertain set). The similar concept of uncertainty is related to version spaces, which were used in Bayes point machines (BPS), [6]. A version space is a set of hypotheses that are consistent with a training sample. A soft version space is a version space where an error in classifying training data is allowed and is controlled by a parameter. The BPS method from each version space finds a representative candidate for a solution as a Bayes point, which is approximated by the center of mass of a polyhedron. In [7], the authors instead of a version space maintain a set of possible weight-vectors in the form of an axis-aligned box and they choose the candidate with the center of mass of a box. In BPS, a final version space is chosen according to the empirical test error, while in [7] the authors compare different boxes by using the principles from SVM: the principle of the empirical risk minimization (ERM) and the structural risk minimization (SRM) for the worst case hypothesis from the box. They also added the third principle of large volume. Large volume transductive principle was briefly treated in [8] for the case of hyperplanes and extended in [9]. In our approach, we deal with uncertainty in the input space instead of a hypothesis space. We propose a theoretical model of knowledge sets, where we define knowledge sets and an uncertain set. The knowledge sets are defined purely on sets, without assuming any particular space for elements or shapes, like boxes.

There are at least three models of a classification problem [10]: the risk minimization model, estimating the regression function of expected conditional probabilities of classification for given data, the Bayes approach of estimating density functions for conditional probabilities of data for particular classes. None of the above models is suitable for the concept of knowledge sets, so we propose a new classification model, called a knowledge set model.

Remark 1. In the knowledge set model, first, we generate knowledge sets. Then, we find a classifier based on the knowledge sets.

The most known method of classification based on predicting density functions from sample data is a Bayes classifier which is an intersection of density functions. Density functions are predicted by using for example Kernel Density Estimation (KDE), [11]. In this paper, we propose a classification method based on the knowledge set model, called knowledge set machines (KSM). In the proposed method instead of predicting directly the decision boundary from estimated density functions, we add an in-
termediate step – constructing knowledge sets. Then, we 
find a classifier based on knowledge sets by using the max-
imal margin principle used for example in SVM. Knowledge 
sets can be interpreted as partitioning an input space. The 
most known algorithm of partitioning for classification are 
decision trees which creates boxes with particular classifi-
cation. There were some attempts to improve partitioning 
by using tighter boxes, covering only part of the input 
space, and using another boxes for classifying the rest of 
the space, [12].

The outline of the paper is as follows. First, we will 
analyze a knowledge set model. Then, we will present the 
KSM method based on this model. Finally, we will show 
experiments and results. The introduction to SVM and 
KSM method based on this model. Finally, we will show 
analyze a knowledge set model. Then, we will present the 

2. Knowledge Set Model

At the beginning, we present some basic definitions and 
propositions. Notation for a knowledge set model is de-
scribed in Appendix A. We will define some mathematical 
structures on a set, which will consist of the environment 
objects - common for proposed structures, and main ob-
jects. We propose the following set of environment objects,
objects - common for proposed structures, and main ob-
structures on a set, which will consist of the environment 
described in Appendix A. We will define some mathematical 
respectively.

density estimation is in Appendix B and Appendix C 
experiments and results. The introduction to SVM and 
KSM method based on this model. Finally, we will show 

Note that the knowledge set (∅, c) is perfect. The comple-
ment of a union of all perfect knowledge sets is a set of 
all elements with the c₀ class. The difference between a 
knowledge set and a perfect knowledge set is that the first 
one is only information which is not necessary true.

Definition 4 (full knowledge set). A full knowledge set 
K is a knowledge set K = (S, c) such as for every x ∈ X : 
m(x) = c, we have x ∈ S.

Proposition 5. A complement of a full knowledge set is 
a subset of a union of some perfect knowledge set and a 
set of all x with a mapping c₀.

Definition 6 (full perfect knowledge set). A full per-
fekt knowledge set is a knowledge set which is full and 
perfect.

A full perfect knowledge set for c ∈ C is a union of all 
perfect knowledge sets for c.

Now we define a pair of two knowledge sets for 
C = {c₁, c₂, c₀}, which we call a knowledge setting.

Definition 7 (knowledge setting). A knowledge setting 
is a pair of knowledge sets, (K₁, K₂), where K₁ = (S₁, c₁), 
K₂ = (S₂, c₂), c₁, c₂ ∈ C, c₁ ≠ c₂.

We could also define a tuple of knowledge sets.

Definition 8 (perfect knowledge setting). A perfect 
knowledge setting is a pair of perfect knowledge sets, (K₁, 
K₂), where K₁ = (S₁, c₁), K₂ = (S₂, c₂), c₁ ≠ c₂.

Definition 9 (full perfect knowledge setting). A full 
perfect knowledge setting is a pair of full perfect knowl-
edge sets, (K₁, K₂), where K₁ = (S₁, c₁), K₂ = (S₂, c₂), 
c₁ ≠ c₂.

The full perfect knowledge setting fully describes the 
mappings M, so that we are able to construct the mappings 
M from it, for x ∈ S₁, m(x) = c₁, for x ∈ S₂, m(x) = c₂, 
otherwise m(x) = c₀.

Definition 10 (uncertain set). An uncertain set U is 
U = (S₁ ∪ S₂)' ∪ (S₁ ∩ S₂) for any knowledge setting (K₁, 
K₂).

For a perfect knowledge setting (K₁, K₂), U = (S₁ ∪ S₂)', 
because S₁∩S₂ = ∅. The intuition behind an uncertain set 
is that we cannot infer about classes of its elements based 
only on information coming from knowledge settings, with-
out knowledge about mappings from the environment.

Definition 11 (almost perfect knowledge setting). An 
almost perfect knowledge setting is a knowledge setting 
(K₁, K₂), where K₁ = (S₁, c₁), K₂ = (S₂, c₂), c₁ ≠ c₂ and 
(S₁ \ (S₁ ∩ S₂), c₁), (S₂ \ (S₁ ∩ S₂), c₂) are perfect know-
ledge sets.
A perfect knowledge setting is a special case of an almost perfect knowledge setting, when $S_1 \cap S_2 = \emptyset$. From any almost perfect knowledge setting, we can construct a perfect knowledge setting by removing a subset $S_1 \cap S_2$ from the knowledge sets.

**Proposition 12.** For an almost perfect knowledge setting, all $x$ such as $m(x) = c_0$ belongs to $U$.

In particular, for a perfect knowledge setting, all $x$ such as $m(x) = c_0$ belongs to $U$.

**Proposition 13.** Any subset of a perfect knowledge set is a perfect knowledge set.

**Example 1.** The environment consists of $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$, $C = \{c_0, c_1, c_2, c_3\}$, and the mappings $M$ are $x_1, x_2, x_3 \mapsto c_1$, $x_4, x_5 \mapsto c_2$, $x_6 \mapsto c_3$, $x_7 \mapsto c_0$. The examples of knowledge sets are $K_1 = \{x_1, x_2\}, c_1$, $K_2 = \{x_4, x_5, x_6\}, c_2$, $K_3 = \{x_6\}, c_3$. $K_4 = \{x_2, x_7\}, c_2$, $K_5 = \{x_1, x_6\}, c_3$. We can notice that $K_1$ is a perfect knowledge set, $K_2$ is a full knowledge set, $K_3$ is a perfect knowledge setting. While giving the next examples, we consider only $C = \{c_0, c_1, c_2\}$, the elements of the class $c_2$ becomes $c_0$. The $(K_1, K_3)$ is a perfect knowledge setting. The $(K_1, K_5)$ is an almost perfect knowledge setting. While giving the next example, we consider only $C = \{c_0, c_1, c_2\}$, the elements of the class $c_3$ becomes $c_0$. For $(K_1, K_4)$, $U = \{x_2, x_3, x_4, x_5, x_6\}$.

A process of mapping classes to elements of $X$ can be interpreted as classifying data in multi-class classification problems, when a set $C$ is finite. For this reason, a model of knowledge sets is suitable for analyzing classification problems. Knowledge settings are especially suitable for analyzing binary classification problems, when only two classes are possible. The mappings $M$ from the environment can be treated as a perfect classification and thus can be used for validation. We define a task of binary classification with limited knowledge as a task of mapping classes to elements of $X$ given only information about some knowledge setting without information about mappings from the environment. For example, we can classify elements from $S_1$ and $S_2$ according to the classes of knowledge sets, and for the uncertain set use any classification.

In order to apply our work to real world problems, we define a set of environments objects, $E_f$, with binary mappings defined not directly, but by using two functions $f_1 : X \to \mathbb{R}$ and $f_2 : X \to \mathbb{R}$. The $x$ element has the $c_1$ class, when $f_1(x) > f_2(x)$, the $x$ element has the $c_2$ class, when $f_2(x) > f_1(x)$, otherwise, when $f_1(x) = f_2(x)$, it has the $c_0$ class. Such definition has the following interpretation. We need to judge the certainty of all elements being in the class $c_1$ by giving them points from the highest which means the most certain. This is the function $f_1(x)$. Next, we judge the certainty of all elements being in the class $c_2$ by the similar procedure. This is the function $f_2(x)$. In order to find a perfect classification, it is enough to predict $\text{sgn}(f_1(x) - f_2(x))$. We could do it by finding two functions $f_1^*(x)$ and $f_2^*(x)$ such as

$$\text{sgn}(f_1^*(x) - f_2^*(x)) = \text{sgn}(f_1(x) - f_2(x)) \quad (1)$$

We want to solve a binary classification task with limited knowledge, so we need to find a knowledge setting. How such knowledge sets can look like? One of the source of ideas for knowledge sets is a definition of the mappings in the environment. We present two types of knowledge sets: margin and superlevel knowledge sets. The first presented type of knowledge sets are margin knowledge sets.

**Definition 14 (margin knowledge sets).** The margin knowledge sets are

$$\{(x : f_1(x) - f_2(x) + e_2(x) < 0), c_2\} \quad (2)$$

$$\{(x : f_1(x) - f_2(x) - e_1(x) > 0), c_1\} \quad (3)$$

where $e_1 : X \to \mathbb{R}$, $e_2 : X \to \mathbb{R}$ and $e_1(x) \geq 0$, $e_2(x) \geq 0$.

Note that margin knowledge sets could be potentially defined in a simpler environment with a function $h : X \to \mathbb{R}$, instead of two functions $f_1(x)$ and $f_2(x)$ with mappings based on $\text{sgn}(h(x))$.

**Proposition 15.** The margin knowledge sets are perfect.

**Proposition 16.** For $e_1(x) \equiv 0$, $e_2(x) \equiv 0$ and the margin knowledge set $K_2$ defined as in (2) and the margin knowledge set $K_1$ defined as in (3), the knowledge setting $(K_1, K_2)$ is a full perfect knowledge setting.

The special case of margin knowledge sets are the simplified margin knowledge sets

$$\{(x : f_1(x) - f_2(x) + b_2 < 0), c_2\} \quad (4)$$

$$\{(x : f_1(x) - f_2(x) - b_1 > 0), c_1\} \quad (5)$$

where $b_1 \geq 0$, $b_2 \geq 0$. For $b_1 = b_2 = 0$, the simplified margin knowledge sets becomes full perfect knowledge sets.

The second presented type of knowledge sets are superlevel knowledge sets.

**Definition 17 (superlevel knowledge set).** A superlevel knowledge set is a knowledge set

$$\{(x : f_1(x) > a_1), c_1\} \quad (6)$$

or

$$\{(x : f_2(x) > a_2), c_2\} \quad (7)$$

for $a_1, a_2 \in \mathbb{R}$.

**Proposition 18.** For $(K_1, K_2)$, where $K_1$ and $K_2$ are superlevel knowledge sets, the knowledge set $K_3 = (S_k \setminus S_1 \cap S_2, c_k)$, where $k = \arg \max a_i$ for $i \in \{1, 2\}$ is a perfect knowledge set.
Proof. Assume that \( a_1 \geq a_2 \) and \( K_4 \) for \( k = 1 \) is not perfect, it means that there exists \( x_p \in S_3 \) that \( f_1 (x_p) \leq f_2 (x_p) \). Due to the fact that \( a_2 \leq a_1 < f_1 (x_p) \leq f_2 (x_p) \), \( x_p \) must belong to \( S_2 \), which contradicts the assumption for \( K_3 \).

We can notice due to Prop. 13 that all subsets of \( K_3 \) are also perfect. The special case of Prop. 18 is when \( a_1 = a_2 \).

Proposition 19. For \((K_1, K_2)\), where \( K_1 \) and \( K_2 \) are superlevel knowledge sets, when \( a_1 = a_2 \), then the knowledge sets \( K_4 = (S_1 \setminus S_2, c_1) \) and \( K_4 = (S_2 \setminus S_1, S_2, c_2) \) are perfect.

From the above proposition, we can conclude that

Proposition 20. For \((K_1, K_2)\), where \( K_1 \) and \( K_2 \) are superlevel knowledge sets and \( a_1 = a_2 \), then \((K_1, K_2)\) is an almost perfect knowledge setting.

The special case of Prop. 18 is when \( S_1 \cap S_2 = \emptyset \).

Proposition 21. For \((K_1, K_2)\), where \( K_1 \) and \( K_2 \) are superlevel knowledge sets and \( S_1 \cap S_2 = \emptyset \), then the knowledge set with \( \max \{a_1, a_2\} \) is a perfect knowledge set.

The special case of Prop. 18 is when \( S_1 \cap S_2 = \emptyset \) and \( a_1 = a_2 \).

Proposition 22. For \((K_1, K_2)\), where \( K_1 \) and \( K_2 \) are superlevel knowledge sets and \( S_1 \cap S_2 = \emptyset \) and \( a_1 = a_2 \), then \( K_1 \) and \( K_2 \) are perfect knowledge sets.

For \( a_1 \neq a_2 \) the above proposition would be untrue in general. For \((K_1, K_2)\), where \( K_1 \) and \( K_2 \) are superlevel knowledge sets and \( S_1 \cap S_2 \neq \emptyset \), the \( K_1 \) and \( K_2 \) are not perfect in general, but some of their subsets can be.

Example 2. The environment consists of \( X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\} \), \( C = \{c_0, c_1, c_2\} \), and the function \( f_1(x) \) is defined as \( f_1(x_1) = 9, f_1(x_2) = 1, f_1(x_3) = 6, f_1(x_4) = 2 \), \( f_1(x_5) = 6, f_1(x_6) = 6, f_1(x_7) = 3 \). The function \( f_2(x) \) is defined as \( f_2(x_1) = 1, f_2(x_2) = 9, f_2(x_3) = 4, f_2(x_4) = 7, f_2(x_5) = 5, f_2(x_6) = 6, f_2(x_7) = 5 \). Based on these functions, we can conclude about the final mappings: \( x_1, x_3, x_5 \rightarrow c_1, x_2, x_4, x_7 \rightarrow c_2, x_6 \rightarrow c_0 \). The examples of margin knowledge sets are \( K_1 = \{\{x : f_1(x) - f_2(x) - 1 > 0\}, c_1\} = \{\{x_1, x_3\}, c_1\} \), \( K_2 = \{\{x : f_1(x) - f_2(x) + 5 < 0\}, c_2\} = \{\{x_2, x_7\}, c_2\} \). The examples of superlevel knowledge sets are: \( K_3 = \{\{x : f_1 (x) > 3\}, c_1\} = \{\{x_1, x_3, x_5, x_6\}, c_1\} \), \( K_4 = \{\{x : f_2 (x) > 5\}, c_2\} = \{\{x_2, x_4, x_6\}, c_2\} \). We can notice that due to Prop. 18, the knowledge set \( K_5 = \{\{x_2, x_4\}, c_2\} \) is perfect.

The goal of our work is to apply a concept of knowledge sets for solving real world problems given a set of empirical data. Usually empirical data are represented by a set of data vectors in the Euclidean space. For this reason, we propose to define a set of environment objects, \( E_{R_i} \), with a universe \( X \) being a Cartesian product \( \mathbb{R}^n \). The solution of the equation

\[
\int f_1 (\bar{x}) - f_2 (\bar{x}) = 0
\]

is called a decision boundary. The decision boundary is a set of all elements with the \( c_0 \) class. The decision boundary of the Bayes solution (76) is a special case of (8), where \( f_1 \) and \( f_2 \) are probability density functions (PDF) for \( X|Y = 1 \), and \( X|Y = 1 \) multiplied by \( \pi_1 \) and \( \pi_2 \) respectively. The Bayes solution is a perfect classifier in the probabilistic model, so it is suitable for defining mappings in the model of knowledge sets.

When \( f_1 (\bar{x}) \) and \( f_2 (\bar{x}) \) are continuous functions, it is enough to find only a decision boundary, and a proper classification in at least one point outside the decision boundary. Due to Prop. 12, given an almost perfect knowledge setting, we can localize a decision boundary outside the knowledge sets and in the intersection.

2.1. Quality of Classification

We define some measures for comparing quality of classification in the model of knowledge sets. First, we present two measures for a candidate mapping function, then we apply them to knowledge sets and knowledge settings. For all environments, \( E, E_f, E_{R_i} \), we can measure quality of classification for a candidate mapping \( m_c(\cdot) \) by some measure \( \mu \) on a set of incorrectly classified elements,

\[
\epsilon [m_c] := \mu \{x : m_c (x) \neq m (x)\}.
\]

Alternatively, we could take the ratio of \( \epsilon [m_c] \) to \( \mu (X) \). The best \( \epsilon [m_c] \) is the minimal one and it is equal 0 for perfect classification. Note that it is possible that the error \( \epsilon \) is equal to 0 also for nonperfect classifications, only when the set of incorrectly classified examples has the measure \( \mu \) equal to 0. In a probabilistic model, the perfect classification is achieved for the Bayes classifier. Given data sample, we are interested in minimizing the number of incorrectly classified unseen data examples.

The motivation for our second measure is a probabilistic model, and a Bayes error rate, which is a probability of incorrect classification being a sum of some integrals of PDF multiplied by \( \pi_0 \) and \( \pi_1 \). [10].

Definition 23. In the environment \( E_f \), assuming that \( f_1 \) and \( f_2 \) are integrable, we can measure quality of classification for a candidate mapping function \( m_c \) by

\[
\epsilon_1 [m_c] := \int_{\{x : m_c (x) = c_2\}} f_1 (x) d\mu_1 + \int_{\{x : m_c (x) = c_1\}} f_2 (x) d\mu_1.
\]

The alternative formulation of \( \epsilon_1 [m_c] \) is

\[
\epsilon_2 [m_c] := \int_{\{x : m_c (x) = c_2\}} f_1 (x) d\mu_1 + \int_{\{x : m_c (x) = c_1\}} f_2 (x) d\mu_1.
\]

The best \( \epsilon_1 [m_c] \) is the minimal one and it is achieved for perfect classification due to the following theorem.
Theorem 24. For a set of candidate mappings \( H \), assuming that \( m(\cdot) \in H \),
\[
m(x) = \arg\min_{m(\cdot) \in H} \varepsilon_1[m_c].
\] (12)
It also holds for \( \varepsilon_a[m_c] \).

The proof is in Appendix D.1. Consider uniqueness of the minimization of the measure \( \varepsilon_1 \). It is possible that the error \( \varepsilon_1 \) is minimal also for nonperfect classification, when the set of incorrectly classified examples has the measure \( \mu_1 \) equal to 0. Moreover, it is also possible for the nonzero measure \( \mu_1 \) only when the set of incorrectly classified examples contains elements from the \( c_0 \) class or elements from the class \( c_1 \) mapped to the \( c_0 \) class (or elements from the class \( c_2 \) mapped to the \( c_0 \) class for \( \varepsilon_a \)).

Let’s analyze the relation of the \( \varepsilon_1[m_c] \) measure with the \( \varepsilon[m_c] \) one. Consider the candidate \( m_c \) in the environment \( E_f \) with the error points \( x_i \), for \( i = 1 \ldots e_1 \) with the \( c_1 \) or \( c_0 \) class, and with the error points \( e_1 \) for \( i = e_1 + 1 \ldots e_2 \) with the \( c_2 \) class. We have \( \varepsilon[m_c] = e_1 + e_2 \). Because the \( \varepsilon[m_c] \) is equal to 0 for the best classification, we subtract \( \varepsilon[m_c] \) from the \( \varepsilon_1[m_c] \) before comparison. And we have \( \varepsilon_1[m_c] = \sum_{i=1}^{e_1} f_1(x_i) - f_1(x_i) + \sum_{i=e_1+1}^{e_2} f_1(x_i) - f_2(x_i) \). We can see, that both type of errors could be equal, i.e. \( \varepsilon[m_c] = \varepsilon_1[m_c] - \varepsilon_1[m_c] \), when for example the difference between \( f_1(x) \) and \( f_2(x) \) is always 0 or 1.

Now, we apply the measures to knowledge sets. In order to apply the measure \( \varepsilon[m_c] \), we need to choose \( m_c \) for knowledge sets. We will use a candidate mapping \( m_c \) consistent with a knowledge set. The candidate mapping \( m_c \) is called consistent with a knowledge set \( K(S,c) \), when for \( x \in K \), we have \( m_c(x) = c \), and for \( x \notin K \), we have \( m_c(x) = c_0 \). Then, we get
\[
\varepsilon[K] := \mu(\{x \in K : c \neq m(x)\} \cup \{x \notin K : c_0 \neq m(x)\}).
\] (13)
The best \( \varepsilon[K] \) is achieved for a full perfect knowledge set. We define the second measure \( \varepsilon_1[m_c] \) for knowledge sets as follows.

Definition 25. We define the error, \( \varepsilon_1 \), for knowledge sets in the environment \( E_f \), assuming that \( f_1 \) and \( f_2 \) are integrable, for \( K_1 = (S_1, c_1) \) as
\[
\varepsilon_1[K_1] := \int_{K_1} f_1(x) \, d\mu_1 + \int_{K_1} f_2(x) \, d\mu_1,
\] (14)
and for \( K_2 = (S_2, c_2) \)
\[
\varepsilon_1[K_2] := \int_{K_2} f_1(x) \, d\mu_1 + \int_{K_2} f_2(x) \, d\mu_1.
\] (15)

Remark 26. It holds that \( \varepsilon_1[K_1] = \varepsilon_a[m_c] \), and \( \varepsilon_1[K_2] = \varepsilon_1[m_c] \), where \( m_c \) is a classification in which elements from the knowledge sets get the same class: for \( x \in K_1 \), \( m_c(x) = c_1 \), while for \( x \in K_2 \), \( m_c(x) = c_2 \). The elements from outside of the knowledge sets get the opposite class or unknown, for example for \( x \notin K_1 \), \( m_c(x) = c_0 \), while for \( x \notin K_2 \), \( m_c(x) = c_0 \).

The \( \varepsilon_1[K] \) becomes minimal when \( K \) is a full perfect knowledge set. Our goal is to create knowledge sets with small errors. We have the following theorem.

Theorem 27. In the environment \( E_f \), for two knowledge sets \( K_1 = (S_1, c_1) \) and \( L_1 = (S_2, c_1) \), assuming that \( K_1 \) is a perfect knowledge set and \( L_1 \subset K_1 \), then
\[
\varepsilon_1[K_1] \leq \varepsilon_1[L_1].
\] (16)
and for two knowledge sets \( K_2 = (S_3, c_2) \) and \( L_2 = (S_4, c_2) \), assuming that \( K_2 \) is a perfect knowledge set and \( L_2 \subset K_2 \), then
\[
\varepsilon_1[K_2] \leq \varepsilon_1[L_2].
\] (17)
Proof. The \( K_1 \cap L_1 \) is an empty set. For \( K_1 \cap L_1 \), the integrals are the same. For \( K_1 \cap L_1 \), we have \( f_1(x) > f_2(x) \), so
\[
\int_{K_1 \cap L_1} f_2(x) \, d\mu_1 \leq \int_{K_1 \cap L_1} f_1(x) \, d\mu_1.
\] (18)
For \( K_1 \cap L_1 \), the integrals are the same. The \( K_2 \cap L_2 \) is an empty set. For \( K_2 \cap L_2 \), the integrals are the same. For \( K_2 \cap L_2 \), we have \( f_2(x) > f_1(x) \), so
\[
\int_{K_2 \cap L_2} f_1(x) \, d\mu_1 \leq \int_{K_2 \cap L_2} f_2(x) \, d\mu_1.
\] (19)
For \( K_2 \cap L_2 \), the integrals are the same.

We can notice that both measures \( \varepsilon[K] \) and \( \varepsilon_1[K] \) are consistent with each other in the sense that for two perfect knowledge sets \( K \) and \( L \) when \( K.c = L.c \) and \( L \subset K \), then both errors are smaller or equal for \( K \), \( \varepsilon[K] \leq \varepsilon[L] \) and \( \varepsilon_1[K] \leq \varepsilon_1[L] \).

Next, we propose to define errors for knowledge settings. We need to define a mapping, \( m_c \) for a knowledge setting. We will use a candidate mapping \( m_c \) consistent with a knowledge setting. The candidate mapping \( m_c \) is called consistent with a knowledge setting \( (K_1, K_2) \), when for \( x \in K_1 \setminus K_2 \), we have \( m_c(x) = c_1 \), and for \( x \in K_2 \setminus K_1 \), we have \( m_c(x) = c_2 \), and for the rest \( x \), we have \( m_c(x) = c_0 \). So for a knowledge setting \( (K_1, K_2) \), we define
\[
\varepsilon[(K_1, K_2)] := \mu(\{x \in K_1 \setminus K_2 : c_1 \neq m(x)\})
\] (20)
\[
\cup \{x \in K_1 \setminus K_2 : c_2 \neq m(x)\}
\] (21)
\[
\cup \{x \notin K_1 \land x \notin K_2 : c_0 \neq m(x)\}
\] (22)
\[
\cup \{x \in K_1 \cap K_2 : c_0 \neq m(x)\}.
\] (23)

For a knowledge setting, we are able to achieve perfect classification, when \( K_1 \) and \( K_2 \) are full perfect knowledge sets. Then \( \varepsilon[(K_1, K_2)] \) would be the measure of an uncertain set \( U \). In order to use the second measure, \( \varepsilon_1[m_c] \)
for knowledge settings, we must redefine it, because it is independent of knowledge about the opposite class.

When we have additional assumption about our environment we can define the following error.

Definition 28. In the environment $E_f$, assuming that $f_1$ and $f_2$ are integrable, and that the set $\{x : m(x) = c_0\}$ has a measure 0, $\mu_1(\{x : m(x) = c_0\}) = 0$, and $f_1(x) \geq 0$, $f_2(x) \geq 0$, for all $x \in X$, we redefine $\varepsilon_c[m_c]$ as

$$
\varepsilon_c[m_c] := \int_{x : m_c(x) = c_2 \lor \epsilon} f_1(x) d\mu_1 + \int_{x : m_c(x) = c_0} f_2(x) d\mu_1.
$$

(24)

The best $\varepsilon_c[m_c]$ is the minimal one and it is achieved for perfect classification due to the following theorem.

Theorem 29. For a set of candidate mappings $H$, assuming that $m(\cdot) \in H$, $m(x) = \arg\min_{m_c(\cdot) \in H} \varepsilon_c[m_c]$. 

(26)

The proof is in Appendix D.2. Consider uniqueness of the minimization of the measure $\varepsilon_c$. It is possible that the error $\varepsilon_c$ is minimal also for nonperfect classification, when the set of incorrectly classified examples has the measure $\mu_1$ equal to 0. For a set with a measure $\mu_1$ greater than 0, it is not possible.

Definition 30. In the environment $E_f$, assuming that $f_1$ and $f_2$ are integrable, and that the set $\{x : m(x) = c_0\}$ has a measure 0, $\mu_1(\{x : m(x) = c_0\}) = 0$, and $f_1(x) \geq 0$, $f_2(x) \geq 0$, for all $x \in X$, we define $\varepsilon_b(K_1, K_2)$ as

$$
\varepsilon_b[[K_1, K_2]] := \int_{(K_1 \setminus K_2)'} f_1(x) d\mu_1 + \int_{(K_2 \setminus K_1)'} f_2(x) d\mu_1.
$$

(27)

Remark 31. It holds that $\varepsilon_b[[K_1, K_2]] = \varepsilon_c[m_c]$, where $m_c$ is a candidate mapping consistent with the knowledge setting $(K_1, K_2)$.

The $\varepsilon_b[[K_1, K_2]]$ becomes minimal when $(K_1, K_2)$ is a full perfect knowledge setting. Our goal is to create knowledge settings with small errors. We have the following theorem.

Theorem 32. In the environment $E_f$, for two knowledge settings $(K_1, K_2)$ and $(L_1, L_2)$, where $(K_1, K_2)$ is a perfect knowledge setting and $L_1 \subset K_1$ and $L_2 \subset K_2$, then

$$
\varepsilon_b[[K_1, K_2]] \leq \varepsilon_b[[L_1, L_2]].
$$

(28)

Proof. The $K_1' \cap L_1$ and $K_2' \cap L_2$ are empty sets. For $(K_1' \cap L_1') \cap (K_2' \cap L_2')$, the integrals are the same. For $K_1 \cap L_1'$, we have

$$
\int_{K_1 \cap L_1'} f_2(x) d\mu_1 \leq \int_{K_1 \cap L_1'} f_1(x) d\mu_1 + \int_{K_1 \cap L_1'} f_2(x) d\mu_1.
$$

(29)

For $K_1 \cap L_1'$, the integrals are the same. For $K_2 \cap L_2'$, we have

$$
\int_{K_2 \cap L_2'} f_1(x) d\mu_1 \leq \int_{K_2 \cap L_2'} f_2(x) d\mu_1 + \int_{K_2 \cap L_2'} f_1(x) d\mu_1.
$$

(30)

For $K_2 \cap L_2'$, the integrals are the same.

We can notice that both measures $\varepsilon_b[[K_1, K_2]]$ and $\varepsilon_b[[L_1, L_2]]$ are consistent with each other in the sense that for two perfect knowledge settings $(K_1, K_2)$ and $(L_1, L_2)$ when $L_1 \subset K_1$ and $L_2 \subset K_2$, then both errors are smaller or equal for $(K_1, K_2)$, $\varepsilon_b[[K_1, K_2]] \leq \varepsilon_b[[L_1, L_2]]$ and $\varepsilon_b[[K_1, K_2]] \leq \varepsilon_b[[L_1, L_2]]$.

Now, we will establish bounds for a binary classification task with limited knowledge. For any mapping candidate $m_c$ such as for $x \in K_1 \setminus K_2$ there is the $c_1$ class, and for $x \in K_2 \setminus K_1$ there is the $c_2$ class, and for the rest $x$, we have any classification, when $(K_1, K_2)$ is an almost perfect knowledge setting, we have a bound on the error $\varepsilon$.

$$
\varepsilon [m_c] \leq \mu(U).
$$

(31)

The similar holds for $\varepsilon_c[m_c]$.

$$
\varepsilon_c[m_c] \leq \varepsilon_b[[K_1, K_2]].
$$

(32)

Proof. For (31). We cannot have a measure $\mu$ greater than $\mu(U)$ for any set of incorrect mappings, because elements from $U'$ are always correctly classified.

For (32). If we map to $c_1$ or $c_2$ any set $S$ which maps to $c_0$, then we replace $\int_S f_1(x) d\mu_1 + \int_S f_2(x) d\mu_1$ with $\int_S f_1(x) d\mu_1$ or $\int_S f_2(x) d\mu_1$ which is obviously fewer.

These bounds are also minimal.

Proof. For (31). We can always create a candidate mapping which is incorrect for all elements from $U$ by for example classifying $c_0$ as $c_1$, $c_1$ as $c_2$, $c_2$ as $c_0$.

For (32). If we do not know that the set with the $c_0$ class has a measure 0, we can always construct a candidate mapping, where all elements from $U$ have a $c_0$ class, then we achieve the equality in the bound.

If we restrict ourselves to the candidate mappings when a $c_0$ class has a measure 0, then we can improve the bound (32) to

$$
\varepsilon_c[m_c] \leq \int_{K_2 \setminus K_1} f_1(x) d\mu_1 + \int_{K_1 \setminus K_2} f_2(x) d\mu_1
$$

(33)

$$
+ \int_U \max (f_1(x), f_2(x)) d\mu_1
$$

(34)

and this is the minimal possible bound. We achieve the equality when we incorrectly classify all elements from $U$ with a $c_1$ class as $c_2$, and opposite.

When we have a group of almost perfect knowledge settings, we might choose the one with the minimal bounds (31) and (32), so we minimize the maximal possible error.
\( \varepsilon[m_e] \) and \( \varepsilon[m_a] \). For computing the bound for \( \varepsilon[m_a] \), we need only information about almost perfect knowledge settings, for computing the bound for \( \varepsilon[m_e] \), we need more information, also about \( f_1(x) \) and \( f_2(x) \).

Let’s analyze whether for given set of knowledge sets, like margin or superlevel, we can always achieve perfect classification. We define a universal knowledge space, as a set of knowledge sets which is able to achieve perfect classification for any environment \( E_f \), in other words if a full perfect knowledge setting can be created with any two of them. Let’s consider margin and superlevel knowledge sets with the assumption that we use the functions from the environment with them. We can notice that margin knowledge sets are universal. When \( e_1(x) \equiv 0 \) and \( e_2(x) \equiv 0 \) in (2) and (3), then we can create a perfect knowledge setting with them. The superlevel knowledge sets are not universal. Because it exists the environment \( E_f \) in which they cannot be a part of a full perfect knowledge setting. Proof. Consider two points \( x_1 \) and \( x_2 \) for which we have \( f_1(x_1) = 4 \), \( f_2(x_2) = 3 \), \( f_2(x_1) = 5 \), \( f_2(x_1) = 1 \), so we have a classification \( m(x_1) = c_2 \), \( m(x_2) = c_1 \). We must choose \( a_1 < 3 \), and \( a_2 < 5 \), but for \( a_1 < 3 \) the point \( x_1 \) will lie in the uncertain set. So it gets incorrect class \( c_0 \) for a mapping consistent with a knowledge setting \( (K_1, K_2) \).

2.2. Knowledge Sets for Uncertain Models

Given the functions \( f_1(x) \) and \( f_2(x) \) in the environment \( E_f \), we have full information about mappings. We can notice, that the best knowledge setting is a margin knowledge setting due to Prop. 16. Consider models with only limited information about the environment \( E_f \), particularly about functions \( f_1(x) \) and \( f_2(x) \). In such models, we have limited information about mappings. This is a source of additional uncertainty, besides the unknown mappings \( c_0 \) from the environment. We propose four such models: limited domain, prediction bands, limited prediction bands and limited measure on an uncertain set. The first three incorporate uncertainty to the functions \( f_1 \) and \( f_2 \). The last model incorporates uncertainty directly to the knowledge sets.

2.2.1. Limited domain model

The limited domain model limits information about functions \( f_1(x) \) and \( f_2(x) \) to the domains \( D_1 \subset X \) and \( D_2 \subset X \) respectively. We can note \( g_1(x) \) as a function \( f_1(x) \) with the limited domain \( D_1 \), and \( g_2(x) \) as a function \( f_2(x) \) with the limited domain \( D_2 \). Our knowledge about \( f_1(x) \) and \( f_2(x) \) is limited to the domains. So we might not be able to find a perfect classification. We can note that the knowledge is limited to two knowledge sets \( (D_1, c_1) \) and \( (D_2, c_2) \).

The question arises what will be the best knowledge setting? We are interested in finding the best possible almost perfect knowledge setting for which we have established the bounds. It means that we want to find the almost perfect knowledge setting with the best possible bounds (31) and (32). We can notice that we achieve the best bounds for the union of \( D_1 \) and \( D_2 \) and margin knowledge sets with \( e_1(x) \equiv 0 \) and \( e_2(x) \equiv 0 \). We can also notice that if \( (K_1, K_2) \) is an almost perfect knowledge setting, where \( K_1 = (D_1, c_1) \), \( K_2 = (D_2, c_2) \), then we achieve the best bounds for \( (K_1, K_2) \). The knowledge setting \( (K_1, K_2) \) can be any type of knowledge setting, particularly a superlevel knowledge setting.

2.2.2. Prediction bands model

The next model of incorporating uncertainty is based on prediction bands for functions \( f_1(x) \) and \( f_2(x) \). We are interested especially in how prediction bands are connected with knowledge sets. The general form of such bands is

\[
\begin{align*}
&f_1(x) + e_1(x), \quad (35) \\
&f_1(x) - e_2(x), \quad (36) \\
&f_2(x) + e_3(x), \quad (37) \\
&f_2(x) - e_4(x). \quad (38)
\end{align*}
\]

where \( e_1(x) \geq 0, e_2(x) \geq 0, e_3(x) \geq 0, e_4(x) \geq 0 \) are functions \( X \to \mathbb{R} \). We have four intersections of these bands

\[
\begin{align*}
&c_1 : f_1(x) + e_1(x) - f_2(x) - e_3(x) = 0, \quad (39) \\
&c_2 : f_1(x) - e_2(x) - f_2(x) - e_3(x) = 0, \quad (40) \\
&c_3 : f_1(x) + e_1(x) - f_2(x) + e_4(x) = 0, \quad (41) \\
&c_4 : f_1(x) - e_2(x) - f_2(x) + e_4(x) = 0. \quad (42)
\end{align*}
\]

We can create perfect knowledge sets from \( c_2 \) and \( c_3 \).

**Proposition 33.** The following knowledge sets are perfect

\[
\{x : f_1(x) - e_2(x) - f_2(x) - e_3(x) > 0\}, c_1 \quad (43)
\]

and

\[
\{x : f_1(x) + e_1(x) - f_2(x) + e_4(x) < 0\}, c_2 \quad (44)
\]

**Proof.** For (43), we get the perfect knowledge set (3), because \( e_2(x) + e_3(x) \geq 0 \). For (44), we get the perfect knowledge set (2), because \( e_1(x) + e_4(x) \geq 0 \).

**Proposition 34.** The following inequalities hold

\[
c_2 \leq c_4, c_2 \leq c_1 \quad (45)
\]

and

\[
c_3 \geq c_1, c_3 \geq c_1 \quad (46)
\]

and

\[
c_2 \leq c_3. \quad (47)
\]

The proof is in Appendix D.3.

**Proposition 35.** In the model with prediction bands, we achieve the best bounds (31) and (32) for the margin knowledge sets (43) and (44).

**Proof.** This is because (43) and (44) are perfect and due to Prop. 34, the uncertain set is the minimal possible (decreasing this set would lead to a nonperfect knowledge setting).
2.2.3. Limited prediction bands

The next model of incorporating uncertainty is the model of limited prediction bands. Given prediction bands, we additionally limit their domains. We limit domains in the form of superlevel knowledge sets

\[f_1(x) + e_1(x), D_1 = \{x : f_1(x) + e_1(x) > a_1\}, \]
\[f_1(x) - e_2(x), D_2 = \{x : f_1(x) - e_2(x) > a_1\}, \]
\[f_2(x) + e_3(x), D_3 = \{x : f_2(x) + e_3(x) > a_2\}, \]
\[f_2(x) - e_4(x), D_4 = \{x : f_2(x) - e_4(x) > a_2\}. \]

Proposition 36. In the model with limited prediction bands, 
when \(a_1 = a_2\), we achieve the best bounds (31) and (32) for the combined knowledge setting - the common part of margin knowledge sets (43), (44) and the superlevel knowledge sets \((K_1, K_2), (K_3, K_4)\) respectively, where \(S_1 = D_2, S_2 = D_3, S_3 = D_1, S_4 = D_4\).

The best combined knowledge setting in the model with cut prediction bands depends on error functions \(e_i(x)\) for \(i = 1, 4\). We consider any type of additional constraints on error functions, such that we will be able to derive general forms of knowledge sets used in the best combined knowledge setting. We will treat separately margin and superlevel knowledge sets. First, we present the result for equivalence of superlevel knowledge sets from the combined knowledge setting and superlevel knowledge sets without error functions by introducing the assumption about the error depending only on the function value and monotonicity as follows.

Proposition 37. In the environment \(E_R\), when the function \(f_1(\bar{x})\) is continuous and the prediction band \(f_1(\bar{x}) - e_2(f_1(\bar{x}))\) has the same monotonicity as \(f_1(\bar{x})\), then the perfect knowledge set

\[\{\bar{x} : f_1(\bar{x}) > a_1\}, c_1\]

is equivalent to

\[\{\bar{x} : f_1(\bar{x}) - e_2(f_1(\bar{x})) > a_2\}, c_1\]

for

\[a_2 = a_1 - e_2(a_1).\]

The proof is in Appendix D.4. This proposition means that the form of superlevel knowledge sets from the model with cut prediction bands is the same as the form of superlevel knowledge sets for the functions \(f_1\) and \(f_2\) without errors.

The second result is to limit errors to constants, when for all \(x, e_i(x) = d_i\), where \(d_i \geq 0\) are some constants for \(i = 1, 2, 3, 4\). In this case, the prediction bands are the shifted original functions. We get standard superlevel knowledge sets for \(a_1 = a_1 - d_1, a_1 = a_1 + d_2\) and \(a_2 = a_2 - d_3, a_2 = a_2 + d_4\) respectively. We also get margin knowledge sets in the form (4) and (5) for \(b_1 = d_2 + d_3, b_2 = d_1 + d_4\).

There is also possibility to define more specific knowledge sets, with specific error functions, but it requires the analysis of particular methods used for estimating functions \(f_1(x)\) and \(f_2(x)\).

Summarizing, we proposed three models for incorporating additional uncertainty for the task of classification. We showed that we can minimize the bounds using margin and superlevel knowledge sets. The next step is to find the final mapping.

2.3. A Solution for Almost Perfect Knowledge Settings

To solve a classification problem, we need to provide mappings for all elements of \(X\). When we have a knowledge setting, we need additionally provide mappings for elements of an uncertain set. In our environment \(E_f\), we have only bounds for all such mappings for an almost perfect knowledge setting. Is it possible to choose any particular candidate mapping, for given almost perfect knowledge setting, in order to minimize the bounds (31) and (32)? For the bound (31), it is not possible to choose any particular \(m_c\), because we can always have a mapping which has different classification in all elements from \(U\) than the chosen \(m_c\) and then we get the bound. So all are equally good. Regarding the bound (32), because we assumed that the measure \(\mu\) for the \(c_0\) class is 0 while defining \(\varepsilon_c\), so it is better to choose the candidate mapping with the same assumption. Any candidate mapping fulfilling this assumption is equally good, because we can always have a mapping which has different classification in all elements from \(U\) than the chosen \(m_c\) and then we get the bound (33). So without any additional assumptions about our environment \(E_f\) we can choose any function.

In practice, we will use a more richer environment \(E_R\), and so we propose to use the principle of maximal margin for finding the candidate mapping in a knowledge set model. This principle were successfully used and justified for the SVM method. For a knowledge setting, we find two parallel hyperplanes between knowledge sets, such as the distance between them is maximal, the decision boundary will be the hyperplane in the half between the two hyperplanes.

2.4. Relation to Support Vector Machines

We propose the model of limited measure on an uncertain set. Consider now combined errors in the following form.

Definition 38. In the environment \(E_f\), assuming that \(f_1\) and \(f_2\) are integrable, and that the set \(\{x : m(x) = c_0\}\) has a measure 0, \(\mu_1(\{x : m(x) = c_0\}) = 0\), and \(f_1(x) \geq 0, f_2(x) \geq 0\), for all \(x \in X\), and \(\mu_2(\{x : m(x) = c_0\}) = 0\), \(a > 0\) is a parameter, we define \(\varepsilon_d[m_c]\) as

\[\varepsilon_d[m_c] := \int_{\{x : m(x) = c_0\}} f_1(x) d\mu_1\]
For \( m() \), the value of the \( \varepsilon_d[m_c] \) cannot be computed, because we get division by zero. Consider now some limitations on possible mapping candidates \( m_c \). We propose the constraint

\[
\mu_2(\{x : m_c(x) = c_0\}) \geq a .
\]  

For a hypothesis space \( H \) with hypotheses fulfilling the constraint (58), we get the best hypothesis in terms of \( \varepsilon_c[m_c] \) for the \( m_c \) for which \( \mu_2(\{x : m_c(x) = c_0\}) = a \).

**Definition 39.** In the environment \( E_I \), assuming that \( f_1 \) and \( f_2 \) are integrable, and that the set \( \{x : m(x) = c_0\} \) has a measure 0, \( \mu_1(\{x : m(x) = c_0\}) = 0 \), and \( f_1(x) \geq 0 \), \( f_2(x) \geq 0 \), for all \( x \in X \), and \( \mu_2(\{x : m(x) = c_0\}) = 0 \), \( a > 0 \) is a parameter, we define \( \varepsilon_c(K_1, K_2) \) as

\[
\varepsilon_c(K_1, K_2) := \frac{1}{(K_1 \setminus K_2)^{\prime}} \int_{(K_1 \setminus K_2)^{\prime}} f_1(x) \, dp_1 + \frac{1}{(K_2 \setminus K_1)^{\prime}} \int_{(K_2 \setminus K_1)^{\prime}} f_2(x) \, dp_1 + \frac{a}{\mu_2(U)} .
\]

**Remark 40.** It holds that \( \varepsilon_c(K_1, K_2) = \varepsilon_d[m_c] \), where \( m_c \) is a candidate mapping consistent with the knowledge setting \( (K_1, K_2) \).

We propose a limitation on knowledge sets, as

\[
\mu_2(U) \geq a .
\]

For a set of knowledge sets fulfilling the constraint (58), we get the best knowledge set in terms of \( \varepsilon_1(K_1, K_2) \) for the \( (K_1, K_2) \) for which \( \mu_2(U) = a \).

The SVM follows the model of uncertainty of limited measure on an uncertain set. In order to notice that, consider the environment \( E_R \) for \( X \) being a closed domain \( D \), with two functions \( f_1(\vec{x}) \) and \( f_2(\vec{x}) \) being density functions in a kernel space. Let the \( h(\vec{x}) = 0 \) be any hyperplane in the kernel space, \( \vec{a} \cdot \vec{x} + b = 0 \). Consider knowledge sets in the form

\[
(\{\vec{x} : h(\vec{x}) > 1\}, c_2)
\]

for class 1, and

\[
(\{\vec{x} : h(\vec{x}) < -1\}, c_1)
\]

for class -1. We will create a method for finding a candidate mapping in our environment, which we will call SVMb. For all possible knowledge settings \( (K_1, K_2) \), where \( K_2 \) is defined as in (62) and \( K_1 \) as in (63), we propose to minimize \( \varepsilon_c(K_1, K_2) \) defined in (59) for a measure \( \mu_2 \) being a squared distance between the margin hyperplanes \( h(\vec{x}) = -1 \) and \( h(\vec{x}) = 1 \). When we have a knowledge setting, we choose a hyperplane \( h(\vec{x}) = 0 \) as a decision boundary. The measure \( \mu_2 \) for an uncertain set for the knowledge setting \( (K_1, K_2) \) is equal to \( 4/||\vec{a}||^2 \). So from (61), we get the constraint

\[
\frac{4}{||\vec{a}||^2} \geq a .
\]

In the \( \varepsilon_c(K_1, K_2) \), we get the last term equal to

\[
\frac{a ||\vec{a}||^2}{4} .
\]

Now, we can notice that SVM is very similar to SVMb. First, we can see that the optimization problem OP 1 can be reformulated as minimization of

\[
\frac{1}{2nC} \mu_2 ||\vec{w}||^2 + \frac{1}{n} \sum_{i=1}^{n} \max (0, 1 - y_i h(\vec{x}_i)) .
\]

We can notice that

\[
\frac{1}{2nC} \mu_2 ||\vec{w}||^2 \leq \frac{1}{2nC} \mu_2 ||\vec{w}||^2 + \frac{1}{n} \sum_{i=1}^{n} \max (0, 1 - y_i h(\vec{x}_i)) .
\]

Because for \( h(\vec{x}) \equiv 0 \), we have \( ||\vec{w}|| = 0 \) and \( b_c = 0 \), so the (66) is equal to 1, and we have

\[
\frac{1}{2nC} \mu_2 ||\vec{w}||^2 + \frac{1}{n} \sum_{i=1}^{n} \max (0, 1 - y_i h(\vec{x}_i)) \leq 1 .
\]

Due to (67), we get

\[
\frac{1}{2nC} \mu_2 ||\vec{w}||^2 \leq 1 .
\]

So

\[
\frac{4}{||\vec{a}||^2} \geq \frac{2}{nC} .
\]

We can notice that the inequality (70) is equivalent to (64) for \( a = 2/(nC) \), then notice also that the parameter in the first term in (68) multiplied by 4 is equal to the \( a \).

In SVM instead of the \( \varepsilon_b[(K_1, K_2)] \), the sum of slack variables is used, (B.1). We can notice that for a data sample generated according to the given probability distribution, we can estimate the two terms in \( \varepsilon_b[(K_1, K_2)] \) by dividing the number of examples for knowledge sets \( K_2^i \) and \( K_1^i \) by all examples. As the number of examples approaches to infinity, we get better estimation. In SVM, instead of the number of examples, we deal with the slack variables. When we divide a slack variable \( \xi_c^{(i)} \) by \( ||\vec{w}|| \), we get a distance from the hyperplane \( h(\vec{x}) + 1 = 0 \) or \( h(\vec{x}) - 1 = 0 \) to the point. Slack variables has been introduced instead of the number of errors to make the optimization problem computationally feasible, [14].

2.5. Applying knowledge sets to the probabilistic model

In practice, we have sample data in the Euclidean space as the input for a classification problem. When we have
a probabilistic model, where sample data are generated according to some unknown conditional probability distributions for data for two classes, we can treat the densities of these distributions multiplied with prior class probabilities as the functions \( f_1 \) and \( f_2 \) in the knowledge set model. The best solution in a probabilistic model is the same as the best solution in the knowledge set model. We can notice that density functions fulfill additional requirements, they must be greater or equal 0 and the integral must be equal to 1.

The source of uncertainty in the probabilistic model is the limited number of data. Let’s analyze how this uncertainty can be converted to the uncertainty about unknown density functions. When we consider data samples from some unknown distribution, we can notice that in the area of input space with small probability of data (small density values), there are no point or only a few. The data samples can be interpreted as knowledge about a density function. So we can notice that we have limited knowledge about the area of input space without points (tails of distributions). It suggests that we should take into account knowledge in the form of superlevel knowledge sets, and to consider prediction for the tails separately. Another reason of considering superlevel knowledge sets could be the limited precision of computation.

The uncertainty in sample data can be also converted to a special type of prediction bands, like confidence bands and intervals for density estimation. For example, there are confidence bands for KDE. Moreover, the assumptions from the Prop. 37 about monotonicity and dependency on the function value are fulfilled for some of these bands, [15].

2.6. Comparison with Statistical Learning

The model of knowledge sets consists of two parts: finding knowledge sets, then finding the final mapping based on knowledge sets. In this paper, we mainly investigated the first part, for the second, we choose the final mapping lying in an uncertain set according to the maximal margin principle. The final decision boundary lies in an uncertain set, so this is a limitation for a set of possible mappings that comes from analysis of knowledge sets. In existing models like statistical learning the main focus is on the last part of the knowledge set model, how to choose the final mapping among given possible mappings. In the knowledge set model the key point is to find the limitations for a set of possible mappings in the form of knowledge sets.

In statistical learning, the risk minimization model has been proposed. As presented by Vapnik in [16, 17], the goal is to minimize the expected loss, called a risk, given distribution of data, (71). The model assumes that data are independent and identically distributed. We minimize the risk functional

\[
R(f) = \int L(\vec{x}, y, f(\vec{x})) dP(\vec{x}, y),
\]  

(71)

where \( L : X \times Y \times R \rightarrow [0, \infty) \) is a loss function, \( f : X \rightarrow Y \). For classification, the loss function is 1, when \( y \neq f(\vec{x}) \), otherwise it is 0, and \( Y = \{-1,1\} \). The distribution function \( P \) on \( X \times Y \) is unknown, but i.i.d. data \((y_1, x_1), \ldots, (y_n, x_n)\) are given, where \( y_i \in \{-1,1\} \) for \( i \in \{1,\ldots,n\} \). The main question is how to find a function \( f \) from a given space of functions, called hypotheses. In the knowledge set model, we address the question how to limit the set of hypotheses by knowledge sets. Because the main part of the knowledge set model is a construction of knowledge sets, so we are able potentially to plug any source of data not only empirical data in order to construct knowledge sets. The risk minimization model is bounded to the empirical data. It tries to find directly the best hypothesis given data points.

In the knowledge set model, the input for the model are functions \( f_1 \) and \( f_2 \) or directly knowledge sets. However, the problem of converting data samples to the functions or knowledge sets must also be addressed, because the most popular source of data are data examples.

The risk minimization model is closely related to the probability setting. In the knowledge set model, the environments \( E_f \) and \( E_R \) for classification problems are more general, we do not assume any probability context. The functions \( f_1 \) and \( f_2 \) in the environment \( E_f \) can be any functions, not necessarily density functions.

One of the goal of statistical learning is to find a hypothesis which is universally consistent, so that for more data points the candidate mapping tends to the perfect mapping. In our approach this problem is only related to constructing knowledge sets. If we construct knowledge sets from predicted density functions, the problem of universal consistency regards only predicting density functions from data points. If we construct knowledge sets from other types of knowledge, not in the form of data points, we do not have this problem.

The generalization bounds based on a Vapnik-Chervonenkis (VC) dimension are as following, [16]. With the probability at least \( 1 - \eta \) the inequality holds true

\[
R(\alpha) \leq R_{\text{emp}}(\alpha) + \varepsilon(n) \left(1 + \frac{4R_{\text{emp}}(\alpha)}{\varepsilon(n)}\right),
\]

(73)

where

\[
\varepsilon(n) = 4\frac{\ln 2\tau + 1}{\tau} - \frac{\ln \eta/4}{\tau},
\]

(74)

\[
\tau = \frac{n}{h},
\]

(75)

\( h \) is a VC dimension. The bound on the risk is a trade-off between the empirical risk and VC dimension. This is a reason of an idea of minimizing the empirical risk and simultaneously the VC dimension. In our model, we do not use the risk, because our model is more general – not necessarily based on distributed data. Based on this bound, Vapnik developed the SVM method which tries to minimize this bound by minimizing an empirical error and the
VC dimension for given data. We also derived SVM with the help of the model of limited measure on an uncertain set. Comparing \( \varepsilon_b([K_1, K_2]) \) with (73), we can notice that the \( \varepsilon_b([K_1, K_2]) \) only tests the knowledge sets, while the bound (73) regards the final mapping. The empirical error \( R_{emp} \) is loosely related to the first two terms in (59). The VC dimension is related to two things in the knowledge set model. Since the VC dimension is related to the maximal margin principle, so finding the final mapping in a knowledge set model is related to the VC dimension. Moreover, the VC dimension is related to the third term in (59) in a way that the third term in (59) can limit the VC dimension. In other words, the final mapping is related more to how the final mapping should look like, knowledge sets are the constraints on where the decision boundary should lie. Both aspects are closely related to each other, for example the extreme for choosing the place for the decision boundary is exactly the definition of the final mapping. Note, that in the derivation of the bounds for the VC dimension based on the maximal margin principle, [18], the domain for random points is used, which is a hypersphere. We can think of about knowledge sets, especially the uncertain set \( U \), that they are the similar limitation to the VC dimension as the domain. The more formal comparison will be done in the future.

3. The Classification Method Based on Knowledge Sets

We propose a classification method based on knowledge sets, which we call KSM. Consider the task of binary classification problem, with predicted \( f_1(x) \) and \( f_2(x) \). In the analysis of uncertain models, we have found that we should look for margin and possibly superlevel knowledge sets. Because we do not know the prediction bands and the limited domain, we should test margin and superlevel knowledge sets with different values of the parameters. If we have limited number of possible tests, we should choose the most promising, for example based on bounds for knowledge sets.

In the KSM method we generate different knowledge sets in the following forms.

**Remark 41.** We propose to investigate practically the following types of knowledge sets:

1. The simplified margin knowledge sets in the form (4) and (5).
2. The superlevel knowledge sets in the form from (6) and (7).

The knowledge sets are generated from the predicted density functions, Fig. 1. We generate only those knowledge sets which are perfect, but according to the predicted density functions, not real density functions which are unknown. We will call them predicted perfect knowledge sets. So the knowledge sets do not intersect with each other. The best knowledge set is found by minimizing the predicted bound (32), \( \varepsilon_b([K_1, K_2]) \).

Consider the prediction of density functions. We estimate density functions \( g_1(\xbar) \) and \( g_2(\xbar) \), \( g_1 \) for \( X|Y = -1 \), \( g_2 \) for \( X|Y = 1 \). The Bayes classifier is

\[
\hat{d}(\xbar) = 1, \text{ when } \hat{\pi}_2 \hat{g}_2(\xbar) > \hat{\pi}_1 \hat{g}_1(\xbar),
\]

\[
\hat{d}(\xbar) = -1 \text{ otherwise },
\]

where \( \hat{\pi}_1 \) is the estimation of \( P(Y = -1) \), \( \hat{\pi}_2 \) is the estimation of \( P(Y = 1) \). To generate knowledge sets, we will use the functions \( \hat{f}_1(\xbar) = \hat{\pi}_1 \hat{g}_1(\xbar) \) and \( \hat{f}_2(\xbar) = \hat{\pi}_2 \hat{g}_2(\xbar) \). We can notice that in addition to density functions, we need also to predict \( \hat{\pi}_0 \) and \( \hat{\pi}_1 \). The first option is to set \( \hat{\pi}_0 = 0.5 \), and \( \hat{\pi}_1 = 0.5 \). The second, which we choose, is to test different values. For margin knowledge sets, it is enough to test only one of them, testing the second one is related to setting different value of the \( b_1 \) or \( b_2 \) parameter. We have the margin knowledge set \( \{ \xbar : \hat{\pi}_1 \hat{g}_1(\xbar) - \hat{\pi}_2 \hat{g}_2(\xbar) > b_1 \} \), \( c_1 \), after dividing by \( \hat{\pi}_1 \), we get

\[
\left( \left\{ \xbar : \hat{g}_1(\xbar) - \frac{\hat{\pi}_2}{\hat{\pi}_1} \hat{g}_2(\xbar) > \frac{b_1}{\hat{\pi}_1} \right\}, c_1 \right).
\]

So we can see that we have only two parameters, \( \hat{\pi}_2/\hat{\pi}_1 \) and \( b_1/\hat{\pi}_1 \). Choosing the best value of \( \hat{\pi}_2/\hat{\pi}_1 \) is realized by cross validation, choosing the best value of \( b_1/\hat{\pi}_1 \) is realized by minimizing the predicted bound (32), \( \varepsilon_b([K_1, K_2]) \). For superlevel knowledge sets, setting \( \hat{\pi}_0 \) and \( \hat{\pi}_1 \) is related to choosing different levels, we have a superlevel knowledge set

\[
\left( \left\{ \xbar : \hat{g}_1(\xbar) > \frac{\hat{\pi}_1}{\hat{\pi}_2} \hat{g}_2(\xbar) \right\}, c_1 \right).
\]

So a level \( a_1/\hat{\pi}_1 \) is responsible for choosing proper knowledge sets and simultaneously for choosing proper prior knowledge about classes. Since, we decided that we use only predicted perfect knowledge sets, so we must test the condition if the superlevel knowledge set is inside the predicted full perfect knowledge set,

\[
\left( \left\{ \xbar : \hat{g}_1(\xbar) - \frac{\hat{\pi}_2}{\hat{\pi}_1} \hat{g}_2(\xbar) > 0 \right\}, c_1 \right).
\]

So, for superlevel knowledge sets, we have also two parameters, the best value of the \( \hat{\pi}_2/\hat{\pi}_1 \) is found by cross validation, and the best value of the \( a_1/\hat{\pi}_1 \) is found by minimizing the predicted bound (32), \( \varepsilon_b([K_1, K_2]) \). The knowledge sets are found based on data examples, so this is the second type of estimation which is used for knowledge sets, in addition to the estimation of the density functions.

After finding knowledge sets, we find a decision boundary by using the SVM method with a fixed big value of \( C_C \) (almost hard margin SVM) run on the data points from inside knowledge sets. The rest of data points are disregarded. The almost hard margin SVM realizes the principle of maximal margin for knowledge sets. It is only the approximated realization, because the knowledge sets are
represented approximately by inner points. The advantage of such representation is that we have the finite number of knowledge sets to check.

The requirement for the KSM is not to introduce new parameters for cross validation due to computational reasons. Because we have the fixed big value of the \( C_c \), we can introduce one new parameter for cross validation, which we will call \( D_c \). We mentioned earlier about this parameter, and it represents \( \hat{\pi}_2/\hat{\pi}_1 \) for both types of knowledge sets. See the pseudocode of the KSM, Alg. 1.

The parameters \( b_1/\hat{\pi}_1 \) and \( b_2/\hat{\pi}_1 \) for margin knowledge sets could be potentially set to 0 and this would be the best choice for predicted density functions according to the predicted bound (32), \( \varepsilon_b([K_1, K_2]) \), but we decided to compute the predicted bound directly from the sample data, not from the predicted density functions. The similar for superlevel knowledge sets, we could potentially find the predicted bound (32) based on predicted density functions, but we decided to compute the predicted bound from the sample data. See the pseudocode for finding margin knowledge sets, Alg. 3, and superlevel knowledge sets, Alg. 2.

The complexity of computing the predicted bound \( \varepsilon_b([K_1, K_2]) \) for given predicted perfect knowledge sets is \( O(n^2) \). For margin knowledge sets, we can find the optimal knowledge sets separately for each class, and the values of functions for training data can be computed before, so the final complexity of choosing the best knowledge sets is \( O(n^2) \). For superlevel knowledge sets we can also find the optimal knowledge sets separately for each class, so the complexity is the same.

We propose to choose the same kernels for density prediction and a solution. We choose radial basis function (RBF) kernel functions with the same values of the \( \sigma \) parameter.

Our method can lead to some exceptional cases, when we cannot use the SVM for computing the final solution. We have the following exceptional cases:

1. the list of data vectors for both knowledge sets is empty,
2. the list of data vectors for one of knowledge sets is empty.

The second case happens for example, when all data examples fall on the same side of the Bayes decision boundary. Since, there are no data examples on the other side, so there is no possibility to create a knowledge set with nonempty set of data vectors. This case can happen for margin and superlevel knowledge sets. The first case can happen only for superlevel knowledge sets, when all data examples fall on the same side of the Bayes decision boundary and additionally there is no points in any of predicted perfect superlevel knowledge sets. There could be many reasons of this state, for example too small data sets. Knowledge sets are generated based on the Bayes classifier, so the inaccurate prediction of density functions and finally the Bayes decision boundary can also cause this effect. Moreover, the parameter \( \hat{\pi}_2/\hat{\pi}_1 \) set to some values differ than 1 by a cross validator can increase this effect. Because of these problems, we decided to combine knowledge sets with SVM. They could be potentially combined with any other classifier. The hybrid method works in a way that for exceptional cases, we use SVM with \( C_c = D_c \). We expect that for data sets with more exceptional cases, we get the solution closer to the SVM solution.

We can think about the KSM that it automatically checks when we can use SVM with a very big value of the \( C_c \) (almost hard margin SVM) and we use it with the reduced data (from inside knowledge sets). The KSM method is a hybrid between the almost hard margin SVM and the soft margin SVM. We expect that KSM could have potentially smaller number of support vectors than SVM due to reduced data and almost hard margin type. However, for multiple exceptional cases, the effect might diminish.

The effect of imbalanced data (more points from the first class, than from the other) is that the class with more points will have more accurate density prediction. Moreover, we multiply density functions with predictions of probabilities of classes, like in the Bayes classifier.

4. Experiments

We compare performance of SVM with KSM for various real world data sets. We chose all real world data sets for binary classification from the LibSVM site [19] which originally come from UCI Machine Learning Repository and Statlog (for the aia data sets, we chose only a1a; the covtype data set is reduced to the first 25000 data vectors); see the details about the data sets in Table 1. We use LIBSVM [20] for running internally SVM in all methods. For all data sets, every feature is scaled linearly to [0, 1]. We performed all tests with the RBF kernel. For variable parameters like the \( D_c, C_c, \sigma \) for the RBF kernel, we use a double grid search method for finding the best values - first a coarse grid search is performed, then a finer grid search as described in [21]. More advanced methods for tuning parameters can also be used to improve results of both methods SVM and KSM, for example [22, 23]. The number of values searched by the grid method is a trade-off between accuracy and speed of tests. We use the procedure similar to repeated double cross validation for performance comparison, [24]. For the outer loop, we run a modified k-fold cross validation for \( k = 20 \) - the training set size is fixed instead of the testing set size, and when it is not possible to create the next fold, we shuffle data and start from the beginning. The standard 5 fold cross validation is used for the inner loop for finding optimal values of the parameters. After finding optimal values, we run the method on training data, and we report results for a test set. We report the ratio of exceptional cases, \( r_e \), for the KSM, which is the ratio of the number of iterations of the outer loop for which KSM chooses to perform
SVM without knowledge sets to all iterations (currently 20). The double cross validation is used only for comparing classifiers, for choosing the final model the standard cross validation is used on all data. We can interpret the ratio \( r_e \), as the chance that the SVM without knowledge sets will be chosen by KSM in the standard cross validation during model selection. For example, the ratio 0.7 means that we expect that there is 70% chance that the KSM will end with the SVM without knowledge sets, and 30% chance that the KSM will end with the SVM with knowledge sets. Only in the latter, we can expect for example decreased number of support vectors due to reduced data. Note that even when KSM chooses SVM without knowledge sets, the parameters of SVM, such as \( C \) and \( \sigma \) might have different values than while performing standard SVM. This happens because for the values of parameters \( C \) and \( \sigma \) chosen by the standard SVM, the KSM might perform SVM with knowledge sets. Due to the above, even when the ratio is 1.0, we might expect that the final model will be different from the standard SVM.

We use the Friedman test with the two tailed Nemenyi post hoc test for checking statistical significance of the classification error and the number of support vectors, as suggested in [25, 26, 27]. First, we average the results over the folds of the outer cross validation, then we perform statistical tests on averaged results, for example we compute ranks for classification error based on averaged classification errors over the folds. Note however that alternative procedures are possible [27, 28], for example computing ranks based on all results, which can lead to less significant results for the proposed methods due to the exceptional cases. The statistical procedure is performed for the level of significance equal to 0.05. The critical values for the Friedman test are taken from the statistical table design specifically for the case with smaller number of tests or methods as suggested in [27], page 249. The critical values for the Nemenyi test are taken from the statistical table for the Tukey test which are divided by \( \sqrt{2} \).

We compare SVM with KSM for various training size. We report the training size per each iteration in the outer loop in Table 1. We report all results in Table 2 and Table 3. The general conclusion is that the KSM with margin and superlevel knowledge sets achieve similar generalization performance as SVM with significantly smaller number of support vectors. The proposed methods are slower than SVM, however we expect improvements in the execution time after some optimizations. The detail observations are

- SVM is the best in average classification error in 25 out of 45 tests, KSM with margin knowledge sets in 14 out of 45 tests, KSM with superlevel knowledge sets in 7 out of 45 tests (columns te1, te2, te3 in Table 2),
- SVM has the best rank in classification error in 23 out of 45 tests, KSM with margin knowledge sets in 12 out of 45 tests, KSM with superlevel knowledge sets in 10 out of 45 tests (columns r1, r2, r3 in Table 2),
- SVM has the smallest number of support vectors in 2 out of 45 tests, KSM with margin knowledge sets in 12 out of 45 tests, KSM with superlevel knowledge sets in 15 out of 45 tests, (columns s1, s2, s3 in Table 2),
- SVM has the best rank in the number of support vectors in 2 out of 45 tests, KSM with margin knowledge sets in 12 out of 45 tests, KSM with superlevel knowledge sets in 15 out of 45 tests, (columns s1, s2, s3 in Table 2),
vectors in 3 out of 45 tests, KSM with margin knowledge sets in 29 out of 45 tests, KSM with superlevel knowledge sets in 13 out of 45 tests (columns sr1, sr2, sr3 in Table 2),

- SVM is the fastest (see columns sp1, sp2, sp3 in Table 2),
- the ratio of exceptional cases is higher for KSM with superlevel knowledge sets in 38 out of 45 tests (columns ex2, ex3 in Table 2). The ratio of exceptional cases is generally rather big and often exceeds 0.5. It means, that in more than half tests (out of 20), the SVM hypotheses were selected by the inner cross validation,
- SVM has the best rank in classification error for 100, 200, 400, 500 data vectors in the training set, for 300 the KSM with margin knowledge sets is the best (r1, r2, r3 in Table 3),
- there is no statistically significant differences in classification error between all methods (column tf in Table 3),
- the KSM with margin knowledge sets has the best average rank in the number of support vectors for all sizes of a training set, the SVM is the worst (columns sr1, sr2, sr3 in Table 3),
- both methods have smaller number of support vectors than SVM which is statistically significant for 100, 200, 300 training vectors (columns sn12, sn13, sn23 in Table 3). For 400 and 500 training vectors we did not achieve significantly better results (we have a small number of tests, for 100 training set size we did not achieve significantly better results (we have a small number of tests, for 100 training set size we had 11 tests, for 200 we have 11 tests, for 300 we have 9 tests, for 400 we have 7 tests, for 500 we have 7 tests).

Summarizing, the proposed method is competitive to SVM in terms of the number of support vectors with similar generalization performance.

4.1. Discussion
4.1.1. What is the relation to rough sets?

The environment E with a special class c0 used for knowledge sets is the same as for rough sets. So we can show similarities between rough sets and knowledge sets. Rough sets is a concept of sets with additional information about their approximations in the form of P-lower and P-upper approximations, [13]. The definition of a rough set is as follows. For a set of all elements X, the R-lower approximation of a set S is

\[ \text{R}_* (S) = \bigcup \{ R(x) : R(x) \subset S \} \],

(81)

the R-upper approximation of S is

\[ \text{R}^* (S) = \bigcup \{ R(x) : R(x) \cap S \neq \emptyset \} \],

(82)

the R-boundary region of S is

\[ R_N_R (S) = R^* (S) - R_* (S) \],

(83)

where the R is an equivalence relation, the R(x) is the equivalence class determined by an element x. The set S is rough, if the boundary region of the S is nonempty, otherwise it is crisp. In order to compare rough sets to our model, we can notice that instead of the relation R, we can use a notion of mappings M by mapping a class from the set C to all elements from an equivalence class. We can notice the following facts:

\[ R_* , R^* \in P (\{ \emptyset , K_1.S_1, K_2.S_2, \{ x : m(x) = c_0 \} \}) \],

(84)

Table 1: Performance of KSM for real world data per training set size, part 1. Column descriptions: id – id of a test, dn – the name of a data set, size – the number of all examples, dim – dimension of a problem, tr – a training set size.

<table>
<thead>
<tr>
<th>id</th>
<th>dn</th>
<th>size</th>
<th>dim</th>
<th>tr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>australian</td>
<td>690</td>
<td>14</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>australian</td>
<td>690</td>
<td>14</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>australian</td>
<td>690</td>
<td>14</td>
<td>300</td>
</tr>
<tr>
<td>3</td>
<td>australian</td>
<td>690</td>
<td>14</td>
<td>400</td>
</tr>
<tr>
<td>4</td>
<td>australian</td>
<td>690</td>
<td>14</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>breast-cancer</td>
<td>675</td>
<td>8</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>breast-cancer</td>
<td>675</td>
<td>8</td>
<td>200</td>
</tr>
<tr>
<td>7</td>
<td>breast-cancer</td>
<td>675</td>
<td>8</td>
<td>300</td>
</tr>
<tr>
<td>8</td>
<td>breast-cancer</td>
<td>675</td>
<td>8</td>
<td>400</td>
</tr>
<tr>
<td>9</td>
<td>breast-cancer</td>
<td>675</td>
<td>8</td>
<td>500</td>
</tr>
<tr>
<td>10</td>
<td>alaAll</td>
<td>26008</td>
<td>123</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>alaAll</td>
<td>26008</td>
<td>123</td>
<td>200</td>
</tr>
<tr>
<td>12</td>
<td>alaAll</td>
<td>26008</td>
<td>123</td>
<td>300</td>
</tr>
<tr>
<td>13</td>
<td>alaAll</td>
<td>26008</td>
<td>123</td>
<td>400</td>
</tr>
<tr>
<td>14</td>
<td>alaAll</td>
<td>26008</td>
<td>123</td>
<td>500</td>
</tr>
<tr>
<td>15</td>
<td>covtypeSmall</td>
<td>25000</td>
<td>54</td>
<td>100</td>
</tr>
<tr>
<td>16</td>
<td>covtypeSmall</td>
<td>25000</td>
<td>54</td>
<td>200</td>
</tr>
<tr>
<td>17</td>
<td>covtypeSmall</td>
<td>25000</td>
<td>54</td>
<td>300</td>
</tr>
<tr>
<td>18</td>
<td>covtypeSmall</td>
<td>25000</td>
<td>54</td>
<td>400</td>
</tr>
<tr>
<td>19</td>
<td>covtypeSmall</td>
<td>25000</td>
<td>54</td>
<td>500</td>
</tr>
<tr>
<td>20</td>
<td>diabetes</td>
<td>768</td>
<td>8</td>
<td>100</td>
</tr>
<tr>
<td>21</td>
<td>diabetes</td>
<td>768</td>
<td>8</td>
<td>200</td>
</tr>
<tr>
<td>22</td>
<td>diabetes</td>
<td>768</td>
<td>8</td>
<td>300</td>
</tr>
<tr>
<td>23</td>
<td>diabetes</td>
<td>768</td>
<td>8</td>
<td>400</td>
</tr>
<tr>
<td>24</td>
<td>diabetes</td>
<td>768</td>
<td>8</td>
<td>500</td>
</tr>
<tr>
<td>25</td>
<td>german</td>
<td>1000</td>
<td>24</td>
<td>100</td>
</tr>
<tr>
<td>26</td>
<td>german</td>
<td>1000</td>
<td>24</td>
<td>200</td>
</tr>
<tr>
<td>27</td>
<td>german</td>
<td>1000</td>
<td>24</td>
<td>300</td>
</tr>
<tr>
<td>28</td>
<td>german</td>
<td>1000</td>
<td>24</td>
<td>400</td>
</tr>
<tr>
<td>29</td>
<td>german</td>
<td>1000</td>
<td>24</td>
<td>500</td>
</tr>
<tr>
<td>30</td>
<td>heart</td>
<td>270</td>
<td>13</td>
<td>100</td>
</tr>
<tr>
<td>31</td>
<td>heart</td>
<td>270</td>
<td>13</td>
<td>200</td>
</tr>
<tr>
<td>32</td>
<td>ionosphere_scale</td>
<td>350</td>
<td>33</td>
<td>100</td>
</tr>
<tr>
<td>33</td>
<td>ionosphere_scale</td>
<td>350</td>
<td>33</td>
<td>200</td>
</tr>
<tr>
<td>34</td>
<td>ionosphere_scale</td>
<td>350</td>
<td>33</td>
<td>300</td>
</tr>
<tr>
<td>35</td>
<td>liver-disorders</td>
<td>341</td>
<td>6</td>
<td>100</td>
</tr>
<tr>
<td>36</td>
<td>liver-disorders</td>
<td>341</td>
<td>6</td>
<td>200</td>
</tr>
<tr>
<td>37</td>
<td>liver-disorders</td>
<td>341</td>
<td>6</td>
<td>300</td>
</tr>
<tr>
<td>38</td>
<td>mushrooms</td>
<td>8124</td>
<td>111</td>
<td>100</td>
</tr>
<tr>
<td>39</td>
<td>mushrooms</td>
<td>8124</td>
<td>111</td>
<td>200</td>
</tr>
<tr>
<td>40</td>
<td>mushrooms</td>
<td>8124</td>
<td>111</td>
<td>300</td>
</tr>
<tr>
<td>41</td>
<td>mushrooms</td>
<td>8124</td>
<td>111</td>
<td>400</td>
</tr>
<tr>
<td>42</td>
<td>mushrooms</td>
<td>8124</td>
<td>111</td>
<td>500</td>
</tr>
<tr>
<td>43</td>
<td>sonar_scale</td>
<td>208</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>44</td>
<td>sonar_scale</td>
<td>208</td>
<td>60</td>
<td>200</td>
</tr>
</tbody>
</table>
5. Summary

In this paper, we proposed a novel theoretical classification method of knowledge sets. We derived some bounds for a classification error. We proposed incorporation of four models of uncertainty to a knowledge set model: limited domain, prediction bands, limited prediction bands and limited measure of an uncertain set. We derived the SVM method from the proposed model. The advantage of a knowledge set model is greater understanding of classification problems and the source of ideas for designing new classification methods.

We proposed also a novel classification method, KSM, based on a knowledge set model. We provided details on implementation and performed experiments. The advantage of the proposed method is an improvement in the number of support vectors compared to SVM, so we achieve simpler results, and we can improve speed of testing new examples.

In the future, we plan to extend the analysis of knowl-
edge sets, for example to give more details about the relation to the VC dimension. For KSM, we plan to decrease the number of exceptional cases, so we would not need to combine the KSM with the SVM.

Acknowledgments

I would like to express my sincere gratitude to Professor Witold Dzwinel and Marcin Kurdziel (AGH University of Science and Technology, Department of Computer Science) and Professor Stan Matwin (Dalhousie University) and Roman Staszyk for discussion and useful suggestions. The research is financed by the National Science Centre, project id 217859, UMO-2013/09/B/ST6/01549, and Professor Stan Matwin (Dalhousie University) and Professor Witold Dzwinel and Marcin Kurdziel (AGH University of Science and Technology, Department of Computer Science) for financial support.

Appendix A. Notation and Symbols

Knowledge set model

c an element of a set of all possible classes C.
C a set of all possible classes; for a binary classification this is C = {c0, c1, c2}, c0 means unknown class.
E a symbol for an environment.
e1, e2, e3, e4 e1 and e2 are used for defining margin knowledge sets, besides, e1, e2, e3, e4 are used for defining prediction bands.
f1, f2 symbols for functions defining classification in the environment E.
K a symbol for a knowledge set, shorty it is a pair K = (S, c), where S denotes a set, c, a class; when not stated differently, it is assumed that K1 = (S1, c1) and K2 = (S2, c2); alternatively the L symbol is used for knowledge sets; when K1 and K2 are superlevel knowledge sets, then the levels are denoted as a1 and a2 respectively.
m a function of mappings.
M a set of mappings.
U a symbol for an uncertain set.

X a universe of possible elements of a set; we denote an element of X as x.

Appendix B. Support Vector Classification Basics

For a classification problem, we consider a set of n training vectors x̄i for i ∈ {1, . . . , n}, where x̄i = (x̄ i1, . . . , x̄ in). The i-th training vector is mapped to y i ∈ {−1, 1}. The m is a dimension of the problem. The support vector classification (SVC) soft margin case optimization problem with ∥∥1 norm is

Optimization problem (OP) 1.

\[ \min_{w_c, b, \xi_c} \frac{1}{2} \|w_c\|^2 + C_c \sum_{i=1}^{n} \xi_i \] (B.1)
subject to
\[ y_i h(x_i) \geq 1 - \xi_i \] , (B.2)
\[ \xi_i \geq 0 \] (B.3)

for i ∈ {1, . . . , n}, where
\[ h(x_i) = w_c \cdot x_i + b_c \] , (B.4)
\[ C_c > 0 \] . (B.5)

The \( h^*(\overline{x}) = w_c^* \cdot \overline{x} + b_c^* = 0 \) is a decision curve of the classification problem. Some of training points can be incorrectly classified.

The \( w_c^* \cdot \overline{x} \) can be computed as
\[ w_c^* \cdot \overline{x} = \sum_{i=1}^{n} y_i \alpha_i^* K(x_i, \overline{x}) \] . (B.6)

Therefore, the decision curve is
\[ h^*(\overline{x}) = \sum_{i=1}^{n} y_i \alpha_i^* K(x_i, \overline{x}) + b_c^* = 0 \] , (B.7)

where \( \alpha_i \) are Lagrange multipliers of the dual problem, \( K(\cdot, \cdot) \) is a kernel function, which appears only in the dual
Appendix C. Density Estimation Basics

We introduce the KDE, also known as the Parzen window method—the basic method for density estimation. We estimate density given sample data by

$$f_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{x - x_i}{h} \right),$$  \hspace{1cm} (C.1)

where $K$ is the kernel, $h$ is the bandwidth. The example of a kernel function is the Gaussian kernel.

Appendix D. Proofs

Appendix D.1. The proof of Thm. 24

Proof. For $m(x) = c_2$, we have $f_1(x) < f_2(x)$, so

$$\int_{x: m(x) = c_2 \cap m(x) = c_2} f_1(x) \, d\mu_1 \leq \int_{x: m(x) = c_2 \cap m(x) = c_0} f_2(x) \, d\mu_1.$$  \hspace{1cm} (D.1)

For $m(x) = c_1$, we have $f_1(x) > f_2(x)$, so

$$\int_{x: m(x) = c_1 \cap m(x) = c_2} f_2(x) \, d\mu_1 \leq \int_{x: m(x) = c_1 \cap m(x) = c_0} f_1(x) \, d\mu_1.$$  \hspace{1cm} (D.2)

For $m(x) = c_1$ and $m_c(x) = c_0$, the integrals are the same. For $m(x) = c_0$, we have $f_1(x) = f_2(x)$ and we get

$$\int_{x: m(x) = c_0 \cap m(x) = c_2} f_2(x) \, d\mu_1 = \int_{x: m(x) = c_0 \cap m(x) = c_0} f_1(x) \, d\mu_1.$$  \hspace{1cm} (D.3)

For $m(x) = c_0$ and $m_c(x) = c_0 \cap m(x) = c_0$, the integrals are the same. For $m(x) = c_1$ and $m_c(x) = c_1$ or $m(x) = c_2$ and $m_c(x) = c_2$, the integrals are the same.

Appendix D.2. The proof of Thm. 29

Proof. For $m(x) = c_2$, we have $f_1(x) < f_2(x)$, so

$$\int_{x: m(x) = c_2 \cap m(x) = c_2} f_1(x) \, d\mu_1 \leq \int_{x: m(x) = c_2 \cap m(x) = c_0} f_2(x) \, d\mu_1,$$  \hspace{1cm} (D.4)

For $m(x) = c_1$, we have $f_1(x) > f_2(x)$, so

$$\int_{x: m(x) = c_1 \cap m(x) = c_2} f_2(x) \, d\mu_1 \leq \int_{x: m(x) = c_1 \cap m(x) = c_0} f_1(x) \, d\mu_1.$$  \hspace{1cm} (D.5)

For $m(x) = c_0$, we have $f_1(x) = f_2(x)$ and we get

$$\int_{x: m(x) = c_0 \cap m(x) = c_2} f_1(x) \, d\mu_1 = \int_{x: m(x) = c_0 \cap m(x) = c_0} f_2(x) \, d\mu_1.$$  \hspace{1cm} (D.6)

For $m(x) = c_0$ and $m_c(x) = c_2 \cap m_c(x) = c_0$, the integrals are the same. For $m(x) = c_1$ and $m_c(x) = c_1$ or $m(x) = c_2$ and $m_c(x) = c_2$, the integrals are the same.
Appendix D.3. The proof of Prop. 34

Proof. First, we will prove that \( c_2 \leq c_4 \). We have

\[
\begin{align*}
    f_1(x) - e_2(x) - f_2(x) - e_3(x) & \leq (D.17) \\
    f_1(x) - e_2(x) - f_2(x) + e_4(x) & \leq (D.18)
\end{align*}
\]

because

\[
- f_2(x) - e_3(x) \leq - f_2(x) + e_4(x) .
\]

It is also true that \( c_2 \leq c_1 \),

\[
\begin{align*}
    f_1(x) - e_2(x) - f_2(x) - e_3(x) & \leq (D.20) \\
    f_1(x) + e_1(x) - f_2(x) - e_3(x) & \geq (D.21)
\end{align*}
\]

because we have

\[
\begin{align*}
    f_1(x) - e_2(x) & \leq f_1(x) + e_1(x) . (D.22)
\end{align*}
\]

It is also true that \( c_3 \geq c_4 \),

\[
\begin{align*}
    f_1(x) + e_1(x) - f_2(x) + e_4(x) & \geq (D.23) \\
    f_1(x) - e_2(x) - f_2(x) + e_4(x) & \geq (D.24)
\end{align*}
\]

because

\[
\begin{align*}
    f_1(x) + e_1(x) & \geq f_1(x) - e_2(x) . (D.25)
\end{align*}
\]

It also true that \( c_3 \geq c_1 \),

\[
\begin{align*}
    f_1(x) + e_1(x) - f_2(x) + e_4(x) & \geq (D.26) \\
    f_1(x) + e_1(x) - f_2(x) - e_3(x) & \geq (D.27)
\end{align*}
\]

because we have

\[
\begin{align*}
    - f_2(x) + e_4(x) \geq - f_2(x) - e_3(x) . (D.28)
\end{align*}
\]

We can also notice that \( c_2 \leq c_4 \leq c_3 \), so \( c_2 \leq c_3 \).

\( \Box \)

Appendix D.4. The proof of Prop. 37

Proof. The knowledge set

\[
\{(\vec{x} : f_1(\vec{x}) > a_1) , c_1) \}
\]

is equivalent to

\[
\{(\vec{x} : g_1(\vec{x}) > a_2) , c_1) \},
\]

if and only if for all \( \vec{x} \) such as \( f_1(\vec{x}) > a_1 \), it holds that \( g_1(\vec{x}) > a_2 \), for all \( \vec{x} \) such as \( f_1(\vec{x}) \leq a_1 \), it holds that \( g_1(\vec{x}) \leq a_2 \). In our case, \( g_1(\vec{x}) \leq f_1(\vec{x}) \) for all \( \vec{x} \), so we can write that \( g_1(\vec{x}) + e_2(\vec{x}) = f_1(\vec{x}) \), where \( e_2(\vec{x}) \geq 0 \), so we can write the second knowledge set as

\[
\{(\vec{x} : f_1(\vec{x}) - e_2(\vec{x}) > a_2) , c_1) \}.
\]

We can notice that \( a_2 \leq a_1 \), because \( f_1(\vec{x}) \) is a continuous function and the set for which \( f_1(\vec{x}) = a_1 \) is nonempty, so for \( a_2 > a_1 \) some points would be omitted. We can also notice that for \( \vec{x}_1 \) and \( \vec{x}_2 \) such as \( f_1(\vec{x}_1) = f_1(\vec{x}_2) \), we must have \( e_1(\vec{x}_1) = e_1(\vec{x}_2) \). Consider \( a_1 \) so that \( f_1(\vec{x}_1) = a_1 \) and \( f_1(\vec{x}_2) = a_1 \), it holds \( a_1 - e_2(\vec{x}_1) = a_2 \), because \( f_1 \) is continuous and it could not be \( a_1 - e_2(\vec{x}_1) < a_2 \) because there would be additional points in the second set, it could not be \( a_1 - e_2(\vec{x}_1) > a_2 \), because there would be fewer points in the second set. We can see that it also holds for \( \vec{x}_2 \), \( a_1 - e_2(\vec{x}_2) = a_2 \), so \( e_2(\vec{x}_1) = e_2(\vec{x}_2) \). So we can write (D.31) as

\[
\{(\vec{x} : f_1(\vec{x}) - e_2(f_1(\vec{x})) > a_2) , c_1) \}.
\]

From the above analysis we can conclude that \( a_2 = a_1 - e_2(a_1) \). So we can write (D.32) as

\[
\{(\vec{x} : f_1(\vec{x}) - e_2(f_1(\vec{x})) > a_1 - e_2(a_1)) , c_1) \}.
\]

So we must prove that (D.29) is equivalent to (D.33). When we notate the prediction band as

\[
b(\vec{x}) = f_1(\vec{x}) - e_2(f_1(\vec{x})) \]

we get: for \( f_1(\vec{x}) = a_1 \), the both inequalities are not fulfilled, for \( f_1(\vec{x}) > a_1 \), we must have

\[
b(\vec{x}) > a_1 - e_2(a_1) .
\]

For \( f_1(\vec{x}) < a_1 \), we must have

\[
b(\vec{x}) < a_1 - e_2(a_1) .
\]

It means that the prediction band must have the same monotonicity as \( f_1(\vec{x}) \).

\( \Box \)

Appendix E. Pseudocode for knowledge set machines

The pseudocode is available in Alg. 1, Alg. 2 and Alg. 3.

References


Alg. 1 Knowledge set machines

**Input:** dataMatrix, $D_c$, $\sigma$

1. $f_1, f_2 = \text{findDensityFunctions}(\text{dataMatrix})$ [by KDE]
2. $f_1 = f_1/\text{divisor}$ [divisor is a weight for basis functions in KDE for $f_1$]
3. $f_2 = f_2/\text{divisor}$
4. if areSuperlevelKnowledgeSets() then
5. $ks_1, ks_2 = \text{findSuperlevelKnowledgeSets}(f_1, f_2, \text{dataMatrix})$
6. else
7. $ks_1, ks_2 = \text{findMarginKnowledgeSets}(f_1, f_2, \text{dataMatrix})$
8. end if
9. knowledgeSetVectors = getAllDataVectorsFromKnowledgeSets(ks1, ks2)
10. if knowledgeSetVectors.isEmpty() or knowledgeSetVectors.isOnlyOneClass() then
11. $\text{svm(dataMatrix, } D_c, \sigma) \{\text{svm(data, C, } \sigma)\}$
12. else
13. $\text{svm(knowledgeSetVectors, 1000000, } D_c, \sigma)$
14. end if

Alg. 2 Finding superlevel knowledge sets

**Input:** dataMatrix, $f_1, f_2, D_c$

1. $f_2 = D_c*f_2$
2. bestErrors1 = unset
3. bestLevel1 = unset
4. for all dataVector1 from dataMatrix where dataVector1.class() == -1 do
5. level1 = $f_1(\text{dataVector1})$
6. errors = 0
7. for all dataVector2 from dataMatrix do
8. if dataVector2.class() == -1 and $f_1(\text{dataVector2}) \leq \text{level1}$ then
9. errors++
10. else if dataVector2.class() == -1 and $f_1(\text{dataVector2}) - f_2(\text{dataVector2}) \leq 0$ then
11. isCorrect=false
12. break
13. end if
14. end for
15. if isCorrect and (bestErrors1 == unset or errors < bestErrors1) then
16. bestLevel1 = level1
17. bestErrors1 = errors
18. end if
19. end for
20. return bestLevel1, bestLevel2

Alg. 3 Finding margin knowledge sets

**Input:** dataMatrix, $f_1, f_2, D_c$

1. $f_2 = D_c*f_2$
2. bestErrors1 = unset
3. bestValue1 = unset
4. for all dataVector1 from dataMatrix where dataVector1.class() == -1 do
5. value = $f_1(\text{dataVector1}) - f_2(\text{dataVector1})$
6. if value $\geq 0.0$ then
7. errors = 0
8. for all dataVector2 from dataMatrix where dataVector2.class() == -1 do
9. if $f_1(\text{dataVector2}) - f_2(\text{dataVector2}) \leq \text{value}$ then
10. errors++
11. end if
12. end for
13. if (bestErrors1 == unset or errors < bestErrors1) then
14. bestValue1 = value
15. bestErrors1 = errors
16. end if
17. end if
18. end for
19. if bestValue1 == unset or bestErrors1 == size(dataMatrix with class -1) then
20. bestValue1 = 0
21. end if
22. return bestValue1, bestValue2

References: