Neurocomputing 149 (2015) 1109–1124

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Solving classification problems by knowledge sets

Marcin Orchel *

Department of Computer Science, AGH University of Science and Technology, Al. A. Mickiewicza 30, 30-059 Kraków, Poland

ARTICLE INFO

Article history:
Received 13 February 2014
Received in revised form
7 July 2014
Accepted 11 July 2014
Communicated by Yukun Bao
Available online 19 July 2014

Keywords:
Support vector machines
Classification
Density estimation
Prior knowledge
Prediction bands

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 with 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.

1. Introduction

One of the 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 knowledge about classification in regions of the input space, [2–5], knowledge about class invariance during transformation of the input. The second type includes knowledge about unlabeled examples, imbalance of classes, and 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 the 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.

* Tel.: +48 510519584.
E-mail address: marcin@orchel.pl

http://dx.doi.org/10.1016/j.neucom.2014.07.022
0925-2312/

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 intermediate step – constructing knowledge sets. Then, we find a classifier based on knowledge sets by using the maximal 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 classification. 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 density estimation is in Appendix B and Appendix C respectively.

2. Knowledge set model

At the beginning, we present some basic definitions and propositions. Notation for a knowledge set model is described in Appendix Appendix A. We will define some mathematical structures on a set, which will consist of the environment objects – common for proposed structures, and main objects. We propose the following set of environment objects, E: a universe X of possible elements of the set S, a set C of possible classes, and a set of mappings M, mapping some of $x \in X$ to some class $c \in C$, $x \to c$. We can define mappings as a function $m: X \to C$. The mappings can also be alternatively defined as an equivalence relation on X. The difference between such defined environment and the environment used for defining rough sets, [13] is that our environment has richer structure with some elements x which may not have mappings. However, we will use the equivalent environment with mappings for all $x \in X$ and with a special class c_0 for elements x which would not have mappings. The main goal of our structure will be to carry information about mapping of all elements of a set S to some class c. So we propose to define a structure which we call a *knowledge set* as follows.

Definition 2 (*knowledge set*). A knowledge set K is a tuple $K = (X, C, M; S, c)$, shortly, without environment objects it is a pair $K = (S, c)$, where $c \neq c_0$. It is a set $S \subset X$ of points with information that every $\vec{s} \in S$ maps to $c \in C$. The c is called *a class of a knowledge set*.

The complement of a knowledge set $K = (S, c)$ is defined as $K' = (S', c)$. We define a union of two knowledge sets $K_1 = (S_1, c)$ and $K_2 = (S_2, c)$ as $K_1 \cup K_2 = (S_1 \cup S_2, c)$, an intersection as $K_1 \cap K_2 = (S_1 \cap S_2, c)$, $K_1 \backslash K_2 = (S_1 \backslash S_2, c)$. We do not define a union and an intersection for two knowledge sets with different classes. We define an inclusion as $K_1 \subset K_2 \iff S_1 \subset S_2$, $s \in K \iff s \in S$.

Definition 3 (*perfect knowledge set*). A perfect knowledge set P is a knowledge set $K = (S, c)$ for which for all $s \in S$, $m(s) = c$.

Note that the knowledge set $(\varnothing, c)$ is perfect. The complement of a union of all perfect knowledge sets is a set of all elements with the c_0 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 $x \in X : m(x) = c$, we have $x \in S$.

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 c_0.

Definition 6 (*full perfect knowledge set*). A full perfect knowledge set is a knowledge set which is full and perfect.

A full perfect knowledge set for $c \in C$ is a union of all perfect knowledge sets for c. The complement of a union of all full perfect knowledge sets is a set of all elements with the c_0 class.

Now we define a pair of two knowledge sets for $C = \{c_1, c_2, c_0\}$, which we call a *knowledge setting*.

Definition 7 (*knowledge setting*). A knowledge setting is a pair of knowledge sets, (K_1, K_2), where $K_1 = (S_1, c_1)$, $K_2 = (S_2, c_2)$, $c_1, c_2 \in C$, $c_1 \neq c_2$.

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, (K_1, K_2), where $K_1 = (S_1, c_1)$, $K_2 = (S_2, c_2)$, $c_1 \neq c_2$.

Definition 9 (*full perfect knowledge setting*). A full perfect knowledge setting is a pair of full perfect knowledge sets, (K_1, K_2), where $K_1 = (S_1, c_1)$, $K_2 = (S_2, c_2)$, $c_1 \neq c_2$.

The full perfect knowledge setting fully describes the mappings M, so that we are able to construct the mappings M from it, for $x \in S_1$, $m(x) = c_1$, for $x \in S_2$, $m(x) = c_2$, otherwise $m(x) = c_0$.

Definition 10 (*uncertain set*). An uncertain set U is $U = (S_1 \cup S_2)' \cup (S_1 \cap S_2)$ for any knowledge setting (K_1, K_2).

For a perfect knowledge setting (K_1, K_2), $U = (S_1 \cup S_2)'$, because $S_1 \cap S_2 = \varnothing$. The intuition behind an uncertain set is that we cannot infer about classes of its elements based only on information coming from knowledge settings, without knowledge about mappings from the environment.

Definition 11 (*almost perfect knowledge setting*). An almost perfect knowledge setting is a knowledge setting (K_1, K_2), where $K_1 = (S_1, c_1)$, $K_2 = (S_2, c_2)$, $c_1 \neq c_2$ and $(S_1 \backslash (S_1 \cap S_2), c_1)$, $(S_2 \backslash (S_1 \cap S_2), c_2)$ are perfect knowledge sets.

A perfect knowledge setting is a special case of an almost perfect knowledge setting, when $S_1 \cap S_2 = \varnothing$. 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 $x_1, x_2, x_3 \mapsto c_1$, $x_4, x_5 \mapsto c_2$, $x_6 \mapsto c_3$, and $x_7 \mapsto c_0$. The examples of knowledge sets are $K_1 = (\{x_1, x_2\}, c_1)$, $K_2 = (\{x_4, x_5, x_6\}, c_2)$, $K_3 = (\{x_6\}, c_3)$, $K_4 = (\{x_2, x_7\}, c_2)$, and $K_5 = (\{x_1, x_6\}, c_3)$. We can notice that K_1 is a perfect knowledge set, K_2 is a full knowledge set, K_3 is a full perfect knowledge set. The (K_1, K_3) is a perfect knowledge setting. For (K_1, K_4), $U = \{x_2, x_3, x_4, x_5, x_6\}$. The (K_1, K_5) is an almost perfect knowledge setting.