Neural Networks 22 (2009) 544–557

Contents lists available at ScienceDirect

Neural Networks

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

2009 Special Issue

A new learning paradigm: Learning using privileged information

Vladimir Vapnik*, Akshay Vashist

NEC Labs America, 4 Independence Way, Princeton, NJ 08540, United States

ARTICLE INFO

Article history:
Received 19 March 2009
Received in revised form 25 May 2009
Accepted 25 June 2009

Keywords:
Machine learning
SVM
SVM+
Hidden information
Privileged information
Learning with teacher
Oracle SVM

ABSTRACT

In the Afterword to the second edition of the book "Estimation of Dependences Based on Empirical Data" by V. Vapnik, an advanced learning paradigm called *Learning Using Hidden Information* (LUHI) was introduced. This Afterword also suggested an extension of the SVM method (the so called SVM$_\gamma$+ method) to implement algorithms which address the LUHI paradigm (Vapnik, 1982–2006, Sections 2.4.2 and 2.5.3 of the Afterword). See also (Vapnik, Vashist, & Pavlovitch, 2008, 2009) for further development of the algorithms.

In contrast to the existing machine learning paradigm where a teacher does not play an important role, the advanced learning paradigm considers some elements of human teaching. In the new paradigm along with examples, a teacher can provide students with hidden information that exists in explanations, comments, comparisons, and so on.

This paper discusses details of the new paradigm[1] and corresponding algorithms, introduces some new algorithms, considers several specific forms of privileged information, demonstrates superiority of the new learning paradigm over the classical learning paradigm when solving practical problems, and discusses general questions related to the new ideas.

1. Introduction: What does it mean "To Learn using privileged information" ?

The existing machine learning paradigm considers a simple scheme: given a set of training examples find in a given collection of functions the one that in the best possible way approximates the unknown decision rule. In such a paradigm a teacher does not play an important role.

In human learning, however, the role of a teacher is very important: along with examples a teacher provides students with explanations, comments, comparisons, and so on. In this paper we introduce elements of human teaching in machine learning. We consider an advanced learning paradigm called learning using privileged information (LUPI), where at the training stage a teacher gives some additional information x^* about training example x; *this privileged information will not be available at the test stage* (Vapnik, 1982–2006). We will develop the LUPI paradigm for support vector machine type of algorithms, and will demonstrate the superiority of the advanced learning paradigm over the classical one.

Formally, the classical paradigm of supervised machine learning is described as follows: given a set of pairs (training data)

$$(x_1, y_1), \ldots, (x_\ell, y_\ell), \quad x_i \in X,\ y_i \in \{-1, 1\},$$

generated according to a fixed but unknown probability measure $P(x, y)$, find among a given set of functions $f(x, \alpha), \alpha \in \Lambda$ the function $y = f(x, \alpha_*)$ that minimizes the probability of incorrect classifications (incorrect values of y). In this paradigm the vector $x_i \in X$ is description of the example and y_i is its classification. The goal is to find the function $y = f(x, \alpha_*)$ that guarantees the smallest probability of incorrect classifications.

The LUPI paradigm can be described as follows: given a set of triplets

$$(x_1, x_1^*, y_1), \ldots, (x_\ell, x_\ell^*, y_\ell), \quad x_i \in X,\ x_i^* \in X^*,\ y_i \in \{-1, 1\},$$

generated according to a fixed but unknown probability measure $P(x, x^*, y)$ find among a given set of functions $f(x, \alpha), \alpha \in \Lambda$ the function $y = f(x, \alpha_*)$ that guarantees the smallest probability of incorrect classification.

In the LUPI paradigm we have exactly the same goal as in the classical paradigm i.e., to find the best function in the admissible set of classification functions. However during the training stage we are given an additional privileged information (triplets (x, x^*, y) instead of pairs (x, y) as in the classical paradigm). The additional information $x^* \in X^*$ belongs (generally speaking) to the space X^* which is different from the space X.

Since the additional information is available at the training stage but it is not available for the test set we call it *privileged information* and the new machine learning paradigm *learning*

* Corresponding author. Tel.: +1 609 750 0170.
E-mail addresses: vlad@nec-labs.com, vapnik@att.net (V. Vapnik), vashist@nec-labs.com (A. Vashist).

[1] In this article we changed the terminology. We will call this paradigm Learning Using Privileged Information (LUPI) (instead of LUHI) since the word privilege better reflects the core idea of the new paradigm.

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.06.042

using privileged information or master-class learning[2] (Vapnik, 1982–2006).

Let us consider several examples where a teacher has an additional information during the training stage. More details will be presented in the latter part of this paper.

1. Suppose our goal is to find a rule that can predict outcome y of a treatment in a year given the current symptoms x of a patient. However at the training stage a teacher also can give additional information x^* about the development of symptoms in three months, in six months, and in nine months. Can this additional information about the development of symptoms improve a rule that predicts the outcome in a year?
2. Suppose that our goal is to find a rule to classify biopsy images into two categories: cancer and non-cancer. Here the problem is given images described in the pixel space find a classification rule in the pixel space. However, along with the picture the doctor has a report, written by a pathologist, which describes the pictures using a high level holistic language. The problem is to use pictures along with the pathologist's reports which will not be available at the test stage to find a good classification rule in the pixel space. In fact, the goal is to make an accurate diagnosis without consulting with a pathologist.
3. Suppose that our goal is to predict the exchange rate of a currency at the moment t in the money exchange problem. In this problem we have observations about the rate before the moment t and the goal is to predict if the rate will go up or down at the moment t. However in the historical data along with observations about the rates before moment t we also have observations about rates after moment t. This information is hidden for testing (but available for training). Can this privileged information (about future in the past) help one to construct a better predictive rule?

The situation with existence of privileged information is very common. In fact, for almost all machine learning problems there exists some sort of privileged information.

In the next section we will introduce a mechanism for SVM type of algorithms, which allows one to take advantage of privileged information. However first let us make the following remark:

> *It is known that well defined learning algorithms (say SVM with a universal kernel) converge, with increasing number of observations, to the Bayesian solution* (Steinwart, 2002; Vapnik, 1998). *The goal of the LUPI paradigm is to use privileged information to significantly increase the rate of convergence.*

2. How privileged information can be used in SVM type of algorithms

The basic idea of SVM is to find the optimal separating hyperplane, the one that makes a small number of training errors and possesses a large margin (Boser, Guyon, & Vapnik, 1992; Cortes & Vapnik, 1995).

There are two cases when constructing the optimal hyperplane: constructing the optimal hyperplane in the separable case (when the number of training errors is equal to zero) and constructing the optimal separating hyperplane in the non-separable case (when the number of training errors is not equal to zero).

To find the optimal hyperplane in the separable case one has to minimize the functional

$$R(w, b) = (w, w)$$

subject to the constraints

$$y_i[(w, x_i) + b] \geq 1, \quad i = 1, \ldots, \ell.$$

To find the optimal hyperplane in the non-separable case one introduces non-negative slack variables

$$\xi_i \geq 0, \quad i = 1, \ldots, \ell,$$

and minimizes the functional

$$R(w, b, \xi) = \frac{1}{2}(w, w) + C \sum_{i=1}^{\ell} \xi_i \tag{1}$$

subject to the constraints

$$y_i[(w, x_i) + b] \geq 1 - \xi_i, \quad i = 1, \ldots, \ell. \tag{2}$$

For both the problems there exist effective solutions (Platt, 1998; Vapnik, 1995, 1998).

From (1) and (2) one can see that there exists some small value of C in objective function (1) that in non-trivial cases makes any problem non-separable (not for all $i = 1, \ldots, \ell$ the equation $\xi_i = 0$ is true).

Note that in the separable case using ℓ observations one has to estimate n parameters of the vector w, while in the non-separable case one has to estimate $n + \ell$ parameters (n parameters of the vector w that defines a hyperplane and ℓ values of slacks ξ_i). Generally speaking, slacks are defined by the values of some function chosen from a wide set of functions (with high VC dimension).

This fact is reflected in the bounds for the rate of convergence: for the separable case one can guarantee a fast rate of convergence which has an order $O(h/\ell)$, where h is the VC dimension of the set of admissible hyperplanes while for the non-separable case one can guarantee only $O(\sqrt{h/\ell})$ rate of convergence (Vapnik, 1982–2006, 1998) (since choosing the slacks is equivalent to choosing a slack-function $\phi(x, \delta^*)$ from the set $\phi(x, \delta)$, $\delta \in \Delta$ which defines the values $\xi_i = \phi(x_i, \delta^*)$, $i = 1, \ldots, \ell$; the admissible set of slack-functions can have a high VC-dimension).

2.1. The key observation: Oracle SVM

Suppose now that any vector $x \in X$ belongs to one and only one of the two classes and that there exists the best (which minimizes the error rate) linear rule, defined by the pair w_0, b_0. Suppose that there also exists the so-called Oracle function

$$\xi(x) = [1 - y_i((w_0, x) + b_0)]_+$$

which satisfies the inequality

$$y_i((w_0, x_i) + b_0) \geq 1 - \xi_i^0, \quad \forall (x_i, y_i),$$

where

$$\xi_i^0 = \xi(x_i).$$

Note that $\xi_i^0 < 1$ if the classification of the vector x_i using hyperplane defined by the pair w_0, b_0 is correct and $\xi_i^0 > 1$ if the classification is incorrect.

Now let a teacher supply us with triplets

$$(x_1, \xi_1^0, y_1), \ldots, (x_\ell, \xi_\ell^0, y_\ell).$$

In this case instead of solving a non-separable type of problem (estimating n parameters of vector w of the hyperplane and ℓ values of slacks ξ_i) one is faced with solving a separable type of problem which leads to estimation of only n parameters of the hyperplane: one has to minimize the functional

$$R(w, b) = (w, w) \tag{3}$$

(same as in the separable case) subject to the following constraints (which are slightly different from the separable case constraints)

$$y_i[(w, x_i) + b] \geq r_i, \quad i = 1, \ldots, \ell, \tag{4}$$

where $r_i = 1 - \xi_i^0$ are known values. Let us call the problem defined by (3), (4) the *Oracle SVM problem*.

[2] In human master-class learning teacher's comments play the most important role. In fact, master-class learning uses the power of privileged information.

It is easy to prove the following fact.

Proposition 1. *If any vector $x \in X$ belongs to one and only one of the classes and there exists an Oracle function with respect to the best decision rule in the admissible set of hyperplanes, then with probability $1-\eta$ the following bound holds true*

$$P(y[(w_\ell, x)+b_\ell] < 0) \le P(1-\xi^0 < 0) + A\frac{h\ln\frac{\ell}{h} - \ln\eta}{\ell}, \quad (5)$$

where $P(y[(w_\ell, x)+b_\ell] < 0)$ is the probability of error for the Oracle SVM solution on the training set of size ℓ, $P(1-\xi^0 < 0)$ is the probability of error for the best solution in the admissible set of functions, h is the VC dimension of the admissible set of hyperplanes, and A is a constant.

That is the Oracle solution converges to the best possible solution in the admissible set of solutions with the rate $O(h/\ell)$.

Indeed, let a and b be random values. Consider three events $a < 0$, $b < 0$, and $a - b < 0$. Note that if event $a < 0$ holds true then at least one of the two following events $b < 0$ or $a - b < 0$ is valid. Therefore,

$$P\{a < 0\} \le P\{b < 0\} + P\{a - b < 0\}.$$

Now let w_ℓ, b_ℓ be the solution to the problem defined by (3), (4). Consider random values $a = y[(w_\ell, x)+b_\ell]$, and $b = 1-\xi^0(x)$. For these events the following inequality holds true

$$P\{y[(w_\ell, x)+b_\ell] < 0\} \le P\{y(w_\ell, x) + b_\ell < 1-\xi^0(x)\} + P\{1-\xi^0(x) < 0\}, \quad (6)$$

where probabilities of events in the inequality are generated by the training sets of size ℓ.

Following exactly as was done in Vapnik (1982–2006) (Chapter 6, Theorem 6.8), one can show, using the uniform convergence arguments, that for events whose values of empirical error are equal to zero (w_ℓ, b_ℓ satisfy (4)), with probability $1-\eta$ the following inequality holds true

$$P\{y[(w_\ell, x)+b_\ell] < 1-\xi^0(x)\} \le A\frac{h\ln\frac{2\ell}{h} - \ln\eta}{\ell}. \quad (7)$$

Note that

$$P\{y[(w_\ell, x)+b_\ell] < 0\} \quad (8)$$

is the probability of error of the estimated rule. Since $\xi^0 = \xi^0(x)$ are the values of the Oracle function for the best admissible rule, the probability of events $\{1-\xi^0(x) < 0\}$ defines the error rate for this rule

$$P\{1-\xi^0(x) < 0\}. \quad (9)$$

Combining (6), (7), (8) and (9) one obtains (5).

Fig. 1 shows the rate of convergence error rate of the SVM, and the Oracle SVM to the Bayesian rate for an artificial problem.

2.2. Privileged information and SVM

In reality, however, a teacher does not know either the values of slacks or the Oracle function. Instead, he can supply us with privileged information $x^* \in X^*$ and with the admissible set of the so called *correcting functions* $\phi(x^*, \delta), \delta \in \Delta$ that have a low VC dimension and contains the correcting function which defines the values of the Oracle function

$$\xi^0(x_i) = \phi(x_i^*, \delta_0), \quad \forall(x_i, x_i^*, y_i).$$

In this case our goal is to minimize (over w, b, δ) the functional

$$R(w, b, \delta) = \sum_{i=1}^{\ell} \theta[\phi(x_i^*, \delta) - 1] \quad (10)$$

(here $\theta(u) = 1$ if $u > 0$ and zero otherwise) subject to constraints

$$y_i[(w, x_i)+b] \ge 1-\phi(x_i^*, \delta), \quad i = 1, \ldots, \ell. \quad (11)$$

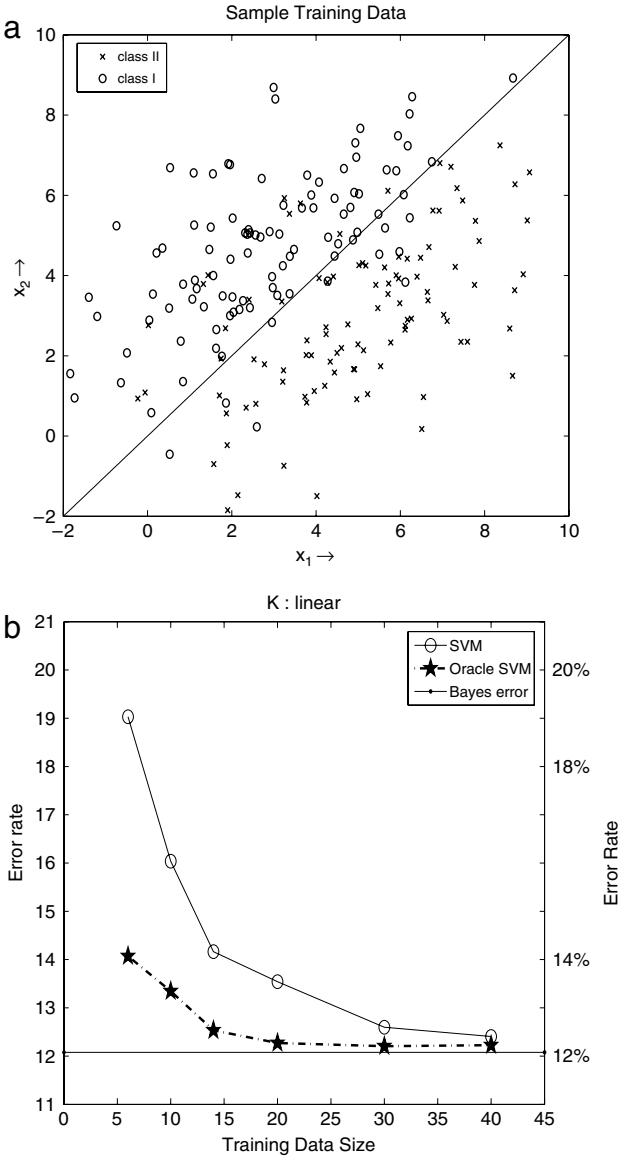

Fig. 1. An artificial problem. (a) Sample data with Bayes decision boundary (diagonal line). (b) Comparison of SVM, Oracle SVM, and Bayesian error rate.

For this problem the following proposition analogous to Proposition 1, is valid:

Proposition 2. *Under the conditions of Proposition 1 with probability $1-\eta$ the following bound holds true*

$$P(y[(w_\ell, x)+b_\ell] < 0) \le P(1-\phi(x^*, \delta_\ell) < 0) + A\frac{(h+h^*)\ln\frac{2\ell}{(h+h^*)} - \ln\eta}{\ell},$$

where $P(y[(w_\ell, x)+b_\ell] < 0)$ is the probability of error for solution of the problem (10), (11) on the training set of size ℓ, $P(1-\phi(x^*, \delta_\ell) < 0)$ is the probability of event $\{\phi(x^*, \delta_\ell) > 1\}$, h is the VC dimension of the admissible set of hyperplanes, and h^* is the VC dimension of the admissible set of correcting functions.

The proof of this Proposition is analogous to the proof of Proposition 1. The only difference is that instead of using the uniform convergence argument over one parameter w that defines the set of linear admissible decision functions (as in (7)) we use the uniform convergence argument over two parameters w and δ that

define both the admissible sets: the set of linear decision functions and the set of correcting functions. We have

$$P\{y[(w_\ell, x) + b_\ell] < 1 - \phi(x^*, \delta_\ell)\} \le A\frac{(h + h^*)\ln\frac{2\ell}{h+h^*} - \ln\eta}{\ell},$$

where $(h + h^*)$ is the sum of the capacities of the two sets of admissible functions: the capacity h of the set of admissible decision functions and the capacity h^* of the set of admissible correcting functions.

To obtain the rate of convergence to the best possible rule one needs to estimate the rate of convergence $P\{\phi(x^*, \delta_\ell) > 1\}$ to $P\{\phi(x^*, \delta_0) > 1\}$. Note that this convergence is defined in the space suggested by a teacher (not in the decision space for the problem of interest).

In standard situation the uniform convergence arguments define an order $O(\sqrt{h^*/\ell})$ where h^* is the VC dimension of the admissible set of correcting functions. However for special constructions of the correcting space X^* (for example, that satisfies the conditions defined by Tsybakov (2004) or the conditions defined by Steinwart and Scovel for SVM (Steinwart & Scovel, 2004)) the convergence can be faster ($O([1/\ell]^\alpha)$, $\alpha > 1/2$).

A good correcting space is the one that allows a rate of convergence faster than the standard one.

The art of teacher is to specify such a space of privileged information and a set of admissible correcting functions that provide a fast rate of convergence

2.3. Two models of correcting functions

In this article we consider two models of the set of correcting functions: (A) the general X^*SVM+ model and (B) the particular dSVM+ model.

(A) In the X^*SVM+ model, an admissible set of non-negative correcting functions is defined in the multi-dimensional X^*-space.

(B) In the dSVM+ model, a set of admissible non-negative correcting functions is defined in a special one-dimensional d-space constructed as follows:

Step 1 Consider the conjugate problem of finding the decision rule in the space X^* by minimizing the functional

$$R(w^*, b^*, \xi^*) = \frac{1}{2}(w^*, w^*) + C\sum_{i=1}^{\ell}\xi_i^*$$

subject to the constraints

$$y_i[(w^*, x_i^*) + b^*] \ge 1 - \xi_i^*, \qquad \xi_i^* \ge 0,\ i = 1, \ldots, \ell$$

(using the classical SVM approach in X^* space). Let w_ℓ^* and b_ℓ^* be the solution to this problem.

Step 2 Using the solution to this problem define the so-called *deviation values*

$$d_i = 1 - y_i[(w_\ell^*, x_i^*) + b_\ell^*].$$

Step 3 Construct a new set of triplets of training data

$$(x_1, d_1, y_1), \ldots, (x_\ell, d_\ell, y_\ell)$$

(use deviation value d as privileged information instead of vector x^*).

Use this training data in SVM+ method (described in the next section) to learn a decision rule.

This idea stresses the main goal, to provide information about the slack variables in the simplest form (which allows one to choose the correcting functions from the set of one-dimensional functions with small VC dimension).

In sections devoted to experiments we will show that both methods of estimating slacks (the general X^*SVM+ and the particular dSVM+) lead to results that significantly outperform the classical SVM method. Also in almost all of our experiments the dSVM+ method outperforms the X^*SVM+ method.

3. Background: SVM and SVM+ methods

3.1. Background of SVM

To learn the decision rule $y = f(x)$ given training data, SVM first maps vectors x of space X into vectors z of space Z where it constructs the optimal separating hyperplane. (In the space X this hyperplane corresponds to some non-linear function (Cortes & Vapnik, 1995; Vapnik, 1998)). In other words, we consider the following problem: minimize the functional

$$R(w, b, \xi) = \frac{1}{2}(w, w) + C\sum_{i=1}^{\ell}\xi_i \tag{12}$$

subject to constraints

$$y_i[(w, z_i) + b] \ge 1 - \xi_i, \qquad \xi_i \ge 0,\ i = 1, \ldots, \ell. \tag{13}$$

The standard technique for solving this quadratic optimization problems is to construct Lagrangian

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2}(w, w) + C\sum_{i=1}^{\ell}\xi_i - \sum_{i=1}^{\ell}\alpha_i[y_i((w, z_i) + b) - 1 + \xi_i] - \sum_{i=1}^{\ell}\beta_i\xi_i,$$

where $\alpha_i \ge 0$ and $\beta_i \ge 0$ are the Lagrange multipliers, minimize this functional over w, b, and ξ and maximize it over multipliers α and β. The (dual space) solution of this problem requires to maximize the functional

$$R(\alpha) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j (z_i, z_j) \tag{14}$$

subject to constraints

$$\sum_{i=1}^{\ell} y_i\alpha_i = 0, \tag{15}$$

$$0 \le \alpha_i \le C, \tag{16}$$

where the vector w is defined by the equation

$$w = \sum_{i=1}^{\ell} y_i\alpha_i z_i,$$

and therefore the decision function $\mathrm{sgn}[(w, z) + b]$ is defined as

$$(w, z) + b = \sum_{i=1}^{\ell} y_i\alpha_i(z_i, z) + b. \tag{17}$$

Since according to Mercer's theorem (Vapnik, 1998) for any inner product in Z space there exists a positive definite function (kernel) $K(x_i, x_j)$ such that

$$(z_i, z_j) = K(x_i, x_j) \tag{18}$$

and vice-versa for any kernel there exists a space Z for which the equality (18) holds one can rewrite the Eqs. (14) and (17) as follows

$$R(\alpha) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \tag{19}$$

$$f(x) = \sum_{i=1}^{\ell} y_i\alpha_i K(x_i, x) + b.$$

Therefore, to find the decision rule one needs to maximize the functional (19) subject to constraints (15) and (16). For detail on the SVM method see Vapnik (1995, 1998).

3.2. Background of SVM+

In the SVM+ method we map vectors x of our training triplets (x, x^*, y) into space Z and vectors x^* into space Z^* where we define

our decision rule and correcting (slack) function as linear functions $(w, z)+b$ and $\xi = (w^*, z^*)+b^*$, respectively (Vapnik, 1982–2006).

To find these functions we minimize the functional

$$R(w, w^*, b, b^*) = \frac{1}{2}[(w, w) + \gamma(w^*, w^*)] + C\sum_{i=1}^{\ell}[(w^*, z_i^*) + b^*]$$

(here we define $\xi_i = [(w^*, z_i^*) + b^*]$) subject to constraints

$$y_i[(w, z_i) + b] \geq 1 - [(w^*, z_i^*) + b^*], \quad i = 1, \ldots, \ell,$$
$$[(w^*, z_i^*) + b^*] \geq 0, \quad i = 1, \ldots, \ell.$$

As in the previous section to solve this problem we construct the Lagrangian

$$L(w, b, w^*, b^*, \alpha, \beta) = \frac{1}{2}[(w, w) + \gamma(w^*, w^*)] + C\sum_{i=1}^{\ell}[(w^*, z_i^*) + b^*] - \sum_{i=1}^{\ell}\alpha_i\left[y_i[(w, z_i) + b] - 1 + [(w^*, z_i^*) + b^*]\right] - \sum_{i=1}^{\ell}\beta_i[(w^*, z_i^*) + b^*],$$

minimize it with respect to w, b, w^*, b^* and maximize with respect to Lagrange multipliers $\alpha \geq 0, \beta \geq 0$.

The (dual space) solution to this problem is defined by the decision function

$$f(x) = (w, z) + b = \sum_{i=1}^{\ell} y_i\alpha_i K(x_i, x) + b. \tag{20}$$

and the corresponding correcting function

$$\phi(x^*) = (w^*, z^*) + b^* = \frac{1}{\gamma}\sum_{i=1}^{\ell}(\alpha_i + \beta_i - C)K^*(x_i^*, x^*) + b^*. \tag{21}$$

Here $K(x_i, x_j)$ and $K^*(x_i^*, x_j^*)$ are kernels in X and X^* spaces that define inner products in Z and Z^* spaces and α, β are the solution of the following optimization problem: maximize the functional

$$R(\alpha, \beta) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \frac{1}{2\gamma}\sum_{i,j=1}^{\ell}(\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C)K^*(x_i^*, x_j^*) \tag{22}$$

subject to three types of constraints

$$\sum_{i=1}^{\ell}(\alpha_i + \beta_i - C) = 0,$$

$$\sum_{i=1}^{\ell} y_i\alpha_i = 0, \tag{23}$$

$$\alpha_i \geq 0, \quad \beta_i \geq 0.$$

3.3. Remarks on SVM+ algorithm

SVM+ algorithm has a simple interpretation. It has two kernels – which in different spaces define similarity measures between two objects. The decision function (20) depends on the kernel defined in the decision space. However, coefficients α generally speaking depend on similarity measures in both the spaces: decision and correcting spaces. Note that admissible SVM+ solutions contain the SVM solution. When the first constraint in (23) is valid as follows

$$\alpha_i + \beta_i - C = 0, \quad i = 1, \ldots, \ell, \tag{24}$$

(rather than overall sum being 0 as in (23)) the third term in (22) reduces to zero and constraints (23) become equivalent to constraints[3]

$$\sum_{i=1}^{\ell} y_i\alpha_i = 0$$
$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, \ell.$$

Also when γ tends to zero the equalities (24) hold true. In these situations we are back to SVM solution. That is when similarity measures in the correcting space are not appropriate the SVM+ algorithm can reject privileged information and construct the SVM solution. Otherwise the SVM+ takes privileged information into account.

From a mathematical point of view the SVM+ algorithm that takes into account both privileged and unprivileged information is very similar to SVM algorithms for finding solutions in the classical pattern recognition framework. It requires solving a quadratic optimization problem under constraints that are similar to constraints in the classical SVM. However the SVM+ algorithm is computationally costlier than SVM. It requires tuning of four hyper-parameters instead of two.

4. Some extensions of the SVM+ method

In this section we consider three extensions of the SVM+ method.

4.1. Mixture model of slacks

In the previous section we modeled slacks by values of some smooth function. This is not always the best choice. Let us model slacks by a mixture of values of some smooth function $\phi(x_i^*) = [(w^*, z_i^*) + b^*]$ and some values ξ_i^*

$$\xi_i = [(w^*, z_i^*) + b^*] + \xi_i^*, \quad i = 1, \ldots, \ell, \tag{25}$$

$$(w^*, z_i^*) + b^* \geq 0, \quad \xi_i^* \geq 0, \ i = 1, \ldots, \ell. \tag{26}$$

Our goal is to minimize the functional

$$R(w, w^*, b, b^*, \xi^*) = \frac{1}{2}[(w, w) + \gamma(w^*, w^*)] + C\sum_{i=1}^{\ell}[(w^*, z_i^*) + b^*] + \theta C\sum_{i=1}^{\ell}\xi_i^* \tag{27}$$

subject to constraints

$$y_i[(w, z_i) + b] \geq 1 - [(w^*, z_i^*) + b^*] - \xi_i^*,$$
$$[(w^*, z_i^*) + b^*] \geq 0,$$
$$\xi_i^* \geq 0.$$

In the Eq. (27) we choose value $\theta > 1$ to reinforce the smooth function part of the solution. (Note that for $0 < \theta \leq 1$ we are back to SVM solution while for sufficiently large value of θ we get the solution described in the previous section.)

The algorithm for finding the dual space solution for this extension almost coincides with the SVM+ algorithm described in the previous section. To define decision and correcting functions (20), (21) one has to maximize the same functional (22) subject to constraints (23) and the constraints

$$0 \leq \alpha_i \leq \theta C, \quad i = 1, \ldots, \ell.$$

4.2. Model where privileged information is available only for a part of the training data

Let us consider the situation when at the training stage a teacher supplies privileged information only for a fraction of examples. That is, the given training data has n pairs

$$(x_1, y_1), \ldots, (x_n, y_n),$$

[3] One can consider the SVM+ algorithm as a form of relaxation of the SVM algorithm.

and $\ell - n$ triplets

$(x_{n+1}, x^*_{n+1}, y_{n+1}), \ldots, (x_\ell, x^*_\ell, y_\ell)$.

In this situation one can introduce a model of slacks only for the cases where we are given the triplets, so we minimize the functional

$$R(w, w^*, b, b^*, \xi) = \frac{1}{2}[(w, w) + \gamma(w^*, w^*)] + C\sum_{i=1}^{n}\xi_i + C^*\sum_{i=n+1}^{\ell}[(w^*, z^*_i) + b^*]$$

subject to constraints

$$y_i[(w, z_i) + b] \geq 1 - \xi_i, \quad i = 1, \ldots, n,$$
$$\xi_i \geq 0, \quad i = 1, \ldots, n,$$
$$y_i[(w, z_i) + b] \geq 1 - [(w^*, z^*_i) + b^*], \quad i = n+1, \ldots, \ell,$$
$$[(w^*, z^*_i) + b^*] \geq 0, \quad i = n+1, \ldots, \ell.$$

The dual space solution for this case defines the decision function as (20) and the correcting function (for examples $n+1, \ldots, \ell$) as

$$\phi(x^*_i) = \frac{1}{\gamma}\sum_{j=n+1}^{\ell}(\alpha_j + \beta_j - C^*)K^*(x^*_j, x^*_i) + b^*, \quad i = n+1, \ldots, \ell,$$

where coefficients α_i, β_i are defined by the vector of maxima of the quadratic form

$$R(\alpha, \beta) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \frac{1}{2\gamma}\sum_{i,j=n+1}^{\ell}(\alpha_i + \beta_i - C^*)(\alpha_j + \beta_j - C^*)K^*(x^*_i, x^*_j)$$

subject to the constraint

$$\sum_{i=1}^{\ell} y_i\alpha_i = 0,$$

the constraint

$$\sum_{i=n+1}^{\ell}(\alpha_i + \beta_i - C^*) = 0,$$

the constraints

$$0 \leq \alpha_i \leq C, \quad i = 1, \ldots, n,$$
$$\alpha_i \geq 0, \quad i = n+1, \ldots, \ell,$$

(or the constraints

$$0 \leq \alpha_i \leq \theta C^*, \quad i = n+1, \ldots, \ell,$$

if one considers the mixture model (25), (26) of slacks $\xi_i, i = n+1, \ldots, \ell$) and the constraints

$$\beta_i \geq 0, \quad i = n+1, \ldots, \ell.$$

4.3. Multiple-space privileged information

Suppose we are given privileged information described in many different spaces. Without loss of generality let us consider two spaces: space X^* and space X^{**}.

Suppose we are given the triplets

$$(x_i, x^*_i, y_i), \quad i = 1, \ldots, n,$$

for one part of data and the triplets

$$(x_i, x^{**}_i, y_i), \quad i = n+1, \ldots, \ell,$$

for another part of data.

Let us map vectors $x \in X$ into space Z, vector $x^* \in X^*$ into space Z^* and vector $x^{**} \in X^{**}$ into space Z^{**} where we consider the linear functions

$$(w, z) + b, \ (w^*, z^*) + b^*, \ (w^{**}, z^{**}) + b^{**}.$$

Our goal is to minimize the functional

$$R(w, w^*, w^{**}, b, b^*, b^{**}) = \frac{1}{2}[(w, w) + \gamma((w^*, w^*) + (w^{**}, w^{**}))] + C\sum_{i=1}^{n}[(w^*, z^*_i) + b^*] + C\sum_{i=n+1}^{\ell}[(w^{**}, z^{**}_i) + b^{**}]$$

subject to the constraints

$$[(w^*, z^*_i) + b^*] \geq 0, \quad i = 1, \ldots, n,$$
$$[(w^{**}, z^{**}_i) + b^{**}] \geq 0, \quad i = n+1, \ldots, \ell,$$
$$y_i[(w, z_i) + b] \geq 1 - [(w^*, z^*_i) + b^*], \quad i = 1, \ldots, n,$$
$$y_i[(w, z_i) + b] \geq 1 - [(w^{**}, z^{**}_i) + b^{**}], \quad i = n+1, \ldots, \ell.$$

The dual space solution to this problem defines the decision function $y = \mathrm{sgn}[f(x)]$

$$f(x) = \sum_{i=1}^{\ell} y_i\alpha_i K(x_i, x) + b,$$

and the two correcting functions: the correcting function for the first set of examples

$$\phi_1(x^*_j) = \frac{1}{\gamma}\sum_{i=1}^{n}(\alpha_i + \beta_i - C)K^*(x^*_i, x^*_j) + b^*, \quad j = 1, \ldots, n,$$

and the correcting function for the second set of examples

$$\phi_2(x^{**}_j) = \frac{1}{\gamma}\sum_{i=n+1}^{\ell}(\alpha_i + \beta_i - C)K^{**}(x^{**}_i, x^{**}_j) + b^{**}, \quad j = n+1, \ldots, \ell.$$

To find the unknown parameters of these functions one has to maximize the functional

$$R(\alpha, \beta) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i\alpha_j K(x_i, x_j) - \frac{1}{2\gamma}\sum_{i,j=1}^{n}(\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C)K^*(x^*_i, x^*_j) - \frac{1}{2\gamma}\sum_{i,j=n+1}^{\ell}(\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C)K^{**}(x^{**}_i, x^{**}_j)$$

subject to constraints

$$\alpha_i \geq 0, \qquad \beta_i \geq 0,$$
$$\sum_{i=1}^{\ell} y_i\alpha_i = 0,$$
$$\sum_{i=1}^{n}(\alpha_i + \beta_i - C) = 0,$$
$$\sum_{i=n+1}^{\ell}(\alpha_i + \beta_i - C) = 0.$$

One can introduce more versions of the LUPI paradigm, however, we restrict ourself to these three.

5. Generalization for the regression estimation problem

The LUPI paradigm can be applied to the regression estimation problem also (see Vapnik (1982–2006), Section 2.5.3 of the Afterword).

5.1. Background of SVM regression

In the classical setting of SVM regression (RSVM), we are given a set of i.i.d. training data

$(x_1, y_1), \ldots, (x_\ell, y_\ell),$

where $x \in X$ is a vector and $y \in (-\infty, \infty)$ is a real value. Our goal is to estimate a real-valued regression function $y = f(x)$.

As before, to solve this problem using the kernel technique we map our vectors $x \in X$ into vectors $z \in Z$ and approximate the regression by a linear function in Z space

$y = (w, z) + b,$

where w and b have to be determined. In RSVM we consider the following setting: we minimize the functional

$$R(w, b) = \frac{1}{2}(w, w) + C \sum_{i=1}^{\ell} |y_i - (w, z_i) - b|_\varepsilon,$$

where u_ε is the so-called ε-insensitive function introduced in Vapnik (1995):

$u_\varepsilon = 0$ if $|u| \le \varepsilon$ and $u_\varepsilon = u$ if $|u| > \varepsilon$.

To minimize this functional we solve the following equivalent problem: minimize the functional

$$R(w, b, \xi, \xi^*) = \frac{1}{2}(w, w) + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*)$$

subject to constraints

$y_i - (w, z_i) - b \le \varepsilon + \xi_i, \quad i = 1, \ldots, \ell,$

$(w, z_i) + b - y_i \le \varepsilon + \xi_i^*, \quad i = 1, \ldots, \ell.$

The dual space solution of this problem has a form

$$y = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) K(x_i, x) + b,$$

where $K(\cdot, \cdot)$ is a positive definite kernel (that defines inner product of Z space in X space).

To find the parameters α, α^* one has to maximize the functional

$$R(\alpha, \alpha^*) = -\varepsilon \sum_{i=1}^{\ell} (\alpha_i + \alpha_j^*) + \sum_{i=1}^{\ell} y_i (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(x_i, x_j)$$

subject to constraints

$$\sum_{i=1}^{\ell} \alpha_i^* = \sum_{i=1}^{\ell} \alpha_i,$$

$$0 \le \alpha_i \le C, \ 0 \le \alpha_i^* \le C.$$

5.2. Background of SVM+ regression

In the situation with privileged information we are given the triplets

$(x_1, x_1^*, y_1), \ldots, (x_\ell, x_\ell^*, y_\ell)$

at the training stage rather than pairs (x_i, y_i)).

As in SVM+ let us map vector x into Z space, vector x^* into space Z^* where we consider three sets of linear functions:

(1) a set of linear functions in Z space $(w, z) + b$ in which we will look for approximating a decision function,

(2) a set of linear functions $(w_1^*, z^*) + b_1^*$ in which we will look for approximation of correcting functions for slacks ξ_i, and

(3) a set of linear functions $(w_2^*, z^*) + b_2^*$ in which we will look for approximation of the correcting functions for slacks ξ_i^*.

Therefore our problem (let us call it RSVM+) will be minimization of the functional

$$R(w, w_1^*, w_2^*, b, b_1^*, b_2^*) = \frac{1}{2}[(w, w) + \gamma[(w_1^*, w_1^*) + (w_2^*, w_2^*)]] + C \sum_{i=1}^{\ell} [(w_1^*, z_i^*) + b_1^*] + C \sum_{i=1}^{\ell} [(w_2^*, z_i^*) + b_2^*]$$

subject to constraints

$y_i - (w, z_i) - b \le \varepsilon + (w_1^*, z_i^*) + b_1^*, \quad i = 1, \ldots, \ell,$

$(w, z_i) + b - y_i \le \varepsilon + (w_2^*, z_i^*) + b_2^*, \quad i = 1, \ldots, \ell,$

$[(w_1^*, z_i^*) + b_1^*] \ge 0, \quad i = 1, \ldots, \ell,$

$[(w_2^*, z_i^*) + b_2^*] \ge 0, \quad i = 1, \ldots, \ell.$

The dual space solution to this problem defines the decision function

$$f(x) = \sum_{i=1}^{\ell} (\alpha_i^* - \alpha_i) K(x_i, x) + b$$

and the two correcting functions

$$\phi_1(x^*) = \frac{1}{\gamma} \sum_{i=1}^{\ell} (\alpha_i + \beta_i - C) K^*(x_i^*, x^*) + b_1^*,$$

$$\phi_2(x^*) = \frac{1}{\gamma} \sum_{i=1}^{\ell} (\alpha_i^* + \beta_i^* - C) K^*(x_i^*, x^*) + b_2^*,$$

where $K(\cdot, \cdot)$ and $K^*(\cdot, \cdot)$ are kernels that define inner products for spaces Z and Z^*, respectively. The parameters $\alpha, \alpha^*, \beta, \beta^*$ are solution to the following optimization problem: maximize the functional

$$R(\alpha, \alpha^*, \beta, \beta^*) = -\varepsilon \sum_{i=1}^{\ell} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{\ell} y_i (\alpha_i^* - \alpha_i) - \frac{1}{2} \sum_{i,j=1}^{\ell} (\alpha_i^* - \alpha)(\alpha_j^* - \alpha_j) K(x_i, x_j) - \frac{1}{2\gamma} \sum_{i,j=1}^{\ell} (\alpha_i^* + \beta_i^* - C)(\alpha_j^* + \beta_j^* - C) K^*(x_i^*, x_j^*) - \frac{1}{2\gamma} \sum_{i,j=1}^{\ell} (\alpha_i + \beta_i - C)(\alpha_j + \beta_j - C) K^*(x_i^*, x_j^*)$$

subject to constraints

$$\sum_{i=1}^{\ell} \alpha_i^* = \sum_{i=1}^{\ell} \alpha_i,$$

$$\sum_{i=1}^{\ell} (\alpha_i^* + \beta_i^* - C) = 0,$$

$$\sum_{i=1}^{\ell} (\alpha_i + \beta_i - C) = 0,$$

$$\alpha_i^* \ge 0, \ \alpha_i \ge 0, \ \beta_i^* \ge 0, \ \beta_i \ge 0, \ i = 1, \ldots, \ell.$$

From a computational point of view the RSVM+ algorithm that finds a solution using privileged information is similar to the classical RSVM algorithm for solving regression estimation problem.

For the RSVM+ algorithm one can consider all the extensions described for the pattern recognition problem.

6. Extracting privileged information by adapting to teacher's concept of distance

Consider one more idea of using privileged information defined by the training vectors

$$(x_1, x_1^*, y_1), \ldots, (x_\ell, x_\ell^*, y_\ell).$$

Let vectors

$$x_i^* = (x_i^*(1), \ldots, x_i^*(m)) \in X^*, \quad i = 1, \ldots, \ell$$

have pair-wise distances in the m-dimensional Euclidean space X^* defined by the following $\ell \times \ell$ matrix

$$M^* = \|a_{i,j}^*\|, \quad i, j = 1, \ldots, \ell$$

$$a_{i,j}^* = \sqrt{\sum_{k=1}^{m} (x_i^*(k) - x_j^*(k))^2},$$

where $x_i^*(k)$ is value of coordinate k of vector x_i^*.

Consider the simplest (from computational point view) case: let us define the metric in X space as follows

$$a_{i,j} = \sqrt{\sum_{k=1}^{n} (x_i(k) - x_j(k))^2 \lambda_k} \tag{28}$$

where the metric of space X has n fixed parameters $\lambda_k \geq 0$ (scaling factors not necessarily equal to one).

Based on this metrics we define matrix of pairwise distances

$$M = \|a_{i,j}\|, \quad i, j = 1, \ldots, \ell.$$

Let us choose such scaling factors in the metric of X-space that specify the closest matrix M to the matrix M^* where closeness ρ of the matrixes we define by the expression

$$\rho = \sum_{i,j}^{\ell} (a_{i,j}^2 - (a_{i,j}^*)^2)^2. \tag{29}$$

The explicit form of this expression is

$$\rho(\lambda) = \sum_{i,j}^{\ell} \left(\sum_{k=1}^{n} \lambda_k (x_i(k) - x_j(k))^2 - (a_{i,j}^*)^2 \right)^2.$$

To find the optimal scaling factor one has to minimize the functional $\rho(\lambda)$ with respect to $\lambda_k \geq 0$. This leads to the solution of the following problem: maximize the quadratic form

$$\sum_{k=1}^{n} \lambda_k c_k - \frac{1}{2} \sum_{k,m=1}^{n} \lambda_k \lambda_m d_{k,m}$$

subject to the constraints $\lambda_k \geq 0$, where we have defined

$$c_k = \sum_{i,j}^{\ell} (x_i(k) - x_j(k))^2 (a_{i,j}^*)^2$$

$$d_{k,m} = \sum_{i,j}^{\ell} (x_i(k) - x_j(k))^2 (x_i(m) - x_j(m))^2.$$

In these equations $x_i(k)$ is the value of coordinate k of vector x_i.

Now one can use vectors $x_i^\lambda = (\sqrt{\lambda_1} x_i(1), \ldots, \sqrt{\lambda_n} x_i(n))$ instead of vector $x_i = (x_i(1), \ldots, x_i(n))$ which is constructed by taking into account the properties of privileged information. This leads to an adaptation of the kernel in solving our learning problems.

For example, using vectors x^λ (instead of x) in the Gaussian kernel one obtains the kernel

$$K(x_i, x_j) = \exp\left\{ -\frac{(x_i - x_j)^{\mathrm{T}} \Sigma (x_i - x_j)}{\sigma^2} \right\}$$

where Σ is a diagonal matrix with diagonal elements λ_k obtained as result of adaptation to the teacher's concept of metric and σ is parameter of the kernel that defines the best capacity factor for the SVM+ solution.

In the general case one can define the distance between two vectors as follows

$$\|x_i - x_j\|_A = \sqrt{(x_i - x_j)^{\mathrm{T}} A (a_i - x_j)}, \tag{30}$$

where A is positive semi-definite matrix. To find this matrix one has to solve the positive definite optimization problem: minimize the functional

$$R(A) = \sum_{i,j=1}^{\ell} [(x_i - x_j)^{\mathrm{T}} A (x_i - x_j) - (a_{i,j}^*)^2]^2 \tag{31}$$

in the set of positive semi-definite matrixes A. This, however, is computationally a very intensive problem (when ℓ is large).

One can consider the intermediate case, when matrix A is restricted to the set

$$A = UU^{\mathrm{T}}, \quad U = (u_1, \ldots, u_t),$$

where $u_1, \ldots, u_t$ are t linearly independent vectors. In this situation one has to minimize the functional

$$R(r_1, \ldots, r_t) = \sum_{i,j=1}^{\ell} \left[\sum_{d=1}^{t} (u_d^{\mathrm{T}} x_i - u^{\mathrm{T}} x_j)^2 - (a_{i,j}^*)^2 \right]^2$$

over $n \times t$ parameters of vectors $u_1, \ldots, u_t$.

The idea of using additional information for matrix learning was considered in the unsupervised learning framework (Chechik & Tishby, 2002; Xing, Ng, Jordan, & Russell, 2002). However in the LUPI paradigm it looks more direct.

7. Three examples of privileged information

In this section we present three examples where different types of privileged information are used for solving different pattern recognition problems. In all these examples we consider a very basic setting of the LUPI paradigm (we consider only one type of privileged information, privileged information is available for all the training data, the correcting values are defined only by the correcting function).

7.1. Advanced technical model as privileged information

One of the important problems in bioinformatics is classification of proteins: to define how they are evolutionarily related. To describe such a relationship human experts have created a hierarchical scheme of organization of proteins taking into account their 3D-structures. The determination of 3D-structures of proteins is very hard and time consuming problem (for many proteins it is not possible to obtain their 3D-structure using existing techniques). On the other hand one can easily obtain amino-acid sequences of proteins. The problem is to construct a rule for classification of proteins into families based on their amino-acid sequences. The main difficulty in this problem is that for some proteins for which the 3D-structure allows to strongly infer homology the amino-acid sequences contain only a weak signal (see Fig. 2(a)).

To obtain the classification rule based on amino-acid sequences the pattern recognition technique is used. There exist several databases that define the hierarchical organization of the proteins, contain their 3D structures and corresponding amino-acid sequences. From these databases one chooses a pair of classes of proteins of interest, uses specific examples of amino-acid sequence and corresponding classification (position in the hierarchy) as training data in the pattern recognition problem to construct the

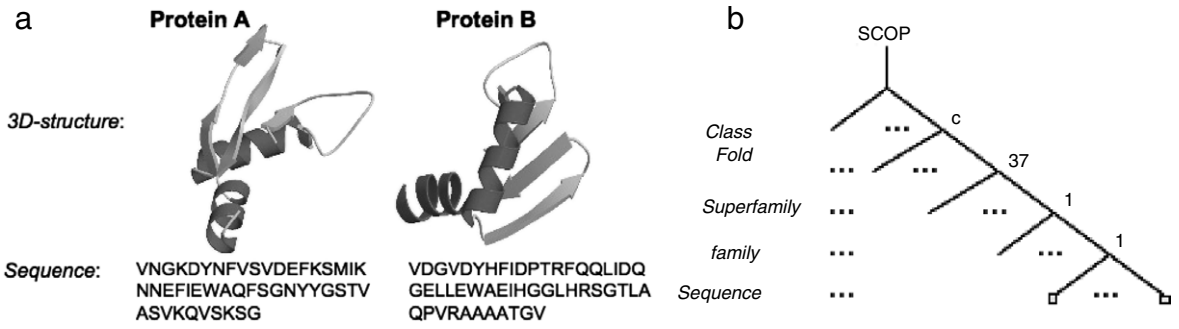

Fig. 2. (a) Two homologous proteins with low amino-acid sequence similarity but high 3D-structure similarity (the similarity is clear after aligning the two schematic 3D-structures) (b) Hierarchical organization of proteins in SCOP database into class, fold, superfamily, family and sequences.

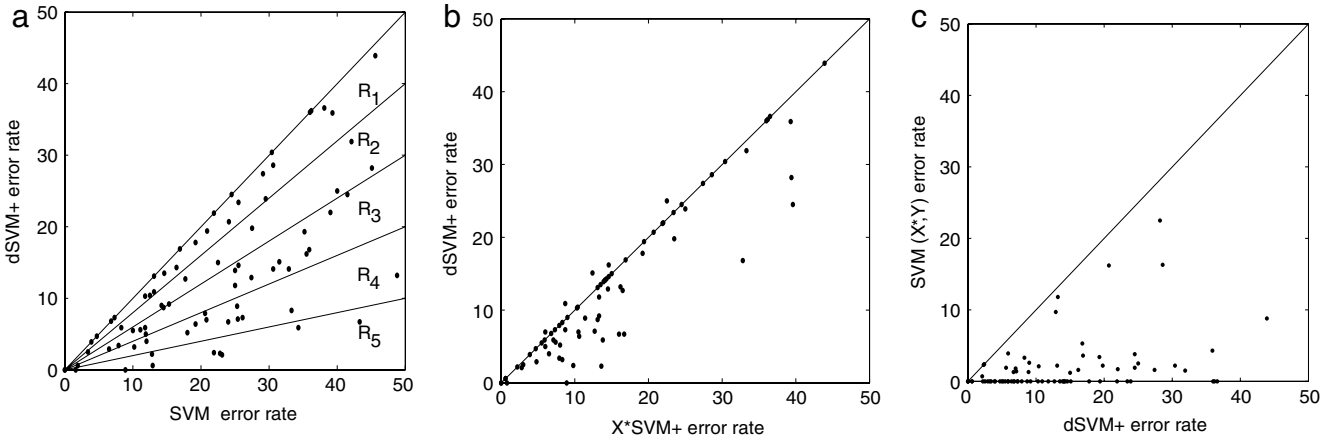

Fig. 3. Comparison of error rates obtained using different algorithms on 80 protein homology problems. (a) dSVM+ vs. SVM, (b) dSVM+ vs. X*SVM+, and (c) dSVM+ vs. SVM based on 3D structures.

desired decision rule. SVM is considered to be one of the best techniques in constructing such decision rules (Kuang et al., 2005; Liao & Noble, 2003).

Below we compare the results of solving protein classification problems in the classical paradigm (SVM algorithm applied to amino-acid sequences and their classifications as training data) with the results of solving the same problems in the LUPI paradigm (using SVM+ algorithm, applied to amino-acid sequences and their classifications, and 3D-structures as privileged information).

In our experiments we used SCOP (Structural Classification of Proteins) database (Murzin, Brenner, Hubbard, & Chothia, 1995), which provides protein sequences and their hierarchical organization (see Fig. 2(b)) defined by human experts according to the 3D-structures of proteins. The 3D-structures for SCOP sequences are available at PDB (Protein Data Bank) (Berman et al., 2000) We focused on determining homology based on protein amino-acid sequences from different superfamilies (third level of hierarchy). We considered 80 binary classification problems, shown in Table 2 which contain 80 different superfamilies with the largest number of sequences. For results to be representative the superfamilies were chosen so that they span diverse protein classes and folds. The similarity between amino-acid sequences was computed using the *profile-kernel* (Kuang et al., 2005) and that between 3D-structures was computed using *MAMMOTH* a program to compare 3D-structures (Ortiz, Strauss, & Olmea, 2002).

For every problem we divided our data (amino-acid sequences) into three parts: about 1/3 of data for training set, about 1/3 of data for validation set and about 1/3 of data for testing set. From these amino-acid sequences we constructed two sets of data X and X^* as follows.

Using two different similarity measures we created two sets of n-dimensional vectors: set of vectors $x \in X$ and set of vectors $x^* \in X^*$. In set X coordinate k of vectors x_i was defined by the value of closeness (in the *profile kernel* measure) of amino-acid i to the amino-acid k while in the set X^* vectors x^* was defined by the value of closeness (in the *MAMMOTH* measure) of the 3D-structure of amino-acid i to the 3D-structure of the amino-acid k. To obtain a solution in the classical paradigm we used the RBF kernel in SVM method. To obtain advanced learning paradigm solutions, we used two RBF kernels in the SVM+ method. For the LUPI paradigm we considered both the dSVM+ and X^*SVM+ methods.

Results of our experiments which compare the classical paradigm with the LUPI paradigm which uses dSVM+ method are summarized in Fig. 3. For details see Table 2.

Fig. 3(a) shows that:

Among the 80 problems considered there was none in which the classical paradigm outperformed the LUPI paradigm.
In 3 cases both the SVM and the LUPI made no test error.
In 11 cases the LUPI scheme was not able to improve the results of the classical one (points lying on diagonal, Fig. 3(a)).
In the remaining cases the LUPI scheme outperformed the classical scheme.
In 15 cases the improvement was small (the number of errors was reduced by less than 1.2 times, points in region R_1).
In 12 cases the improvement was significant (the number of errors was reduced between 1.2 and 1.5 times, points in region R_2).
In 17 cases the improvement was big (the number of errors was reduced between 1.5 and 2 times, points in region R_3).
In 13 cases the improvement was major (the number of errors was reduced between 2.5 and 5 times, points in region R_4).

Table 1
Error rates of SVM, X*SVM+, dSVM+, and Oracle SVM on qualitatively predicting the Mackey–Glass series.

Steps ahead, T = Training size		1	5	8	Steps ahead, T = Training size	1	5	8
SVM	100	2.7	5.2	8.2	400	2.2	3.5	5.2
X*SVM+	100	2.4	5.0	7.8	400	1.8	3.1	4.7
dSVM+	100	2.0	4.8	7.2	400	1.7	2.9	4.3
Oracle SVM	100	1.6	2.9	5.3	400	1.2	1.8	2.8
SVM	200	2.5	4.9	7.3	500	2.1	3.2	5.0
X*SVM+	200	2.1	4.6	6.8	500	1.7	3.1	4.5
dSVM+	200	1.9	3.8	6.5	500	1.7	2.7	4.2
Oracle SVM	200	1.2	2.2	4.6	500	1.1	1.5	2.7
≈Bayes (SVM with 10,000 training examples)		0.3	0.5	0.6		0.3	0.5	0.6

In 9 cases the improvement was dramatic (the number of errors was reduced by more than 5 times, points in region R_5).

Results obtained using the X^*SVM+ method of the LUPI paradigm also outperform the classical SVM paradigm. However, as shown in Fig. 3(b), in almost all cases the dSVM+ method outperformed the X^*SVM+ method. Fig. 3(c) shows that classification based on information about 3D-structure of proteins is much more accurate than classification based on information about amino-acid sequences.

7.2. Future events as privileged information

In many problems (for example, in finance market prediction) it is important to predict values of time series. There are two settings of the time series prediction problem:

1. The quantitative prediction problem: given historical information about the values of time series up to moment t predict the value of the time series at the moment $t + \Delta$.
2. The qualitative prediction problem: given historical information about the values of time series up to moment t predict if the value of the time series at the moment $t + \Delta$ will be larger (the first class) or smaller (the second class) than the value at the moment t (roughly speaking to make a decision to sell or buy[4]).

For both the settings one can use the LUPI paradigm: for the quantitative setting one uses it in the regression framework and for the qualitative setting one uses it in the pattern recognition framework. We experimented with the qualitative problem (using the pattern recognition technique).

Many researchers consider the model chaotic time series introduced by Mackey and Glass which is a solution of the equation

$$\frac{dx(t)}{dt} = -ax(t) + \frac{bx(t-\tau)}{1 + x^{10}(t-\tau)},$$

where a, b, and τ (delay) are parameters of the equation. Using different initial conditions $x(\tau) = x_\tau$ one obtains different realizations this quasi-chaotic series.

We used the Mackey–Glass series with parameters $a = 0.1$, $b = 0.2$, $\tau = 17$ (these are the usual parameters for experimental studies of algorithms for chaotic time series prediction) and initial condition $x(\tau) = 0.9$.

There exist many articles devoted to the prediction of the Mackey-Glass time series using different algorithms (Casdagli, 1989; Mukherjee, Osuna, & Girosi, 1997). Below to compare the LUPI paradigm with the SVM, we use the same parameters as used in the article (Mukherjee et al., 1997) where it has been demonstrated that SVM outperforms many classical algorithms for time series prediction.

In contrast to Mukherjee et al. (1997), we used the pattern recognition setting rather than the regression setting. This setting better reflects finance market problems (as it relates to buy or sell decisions). Article (Mukherjee et al., 1997) considered one step ahead prediction problem ($T = 1$). It turn out, however, to be an easy problem. Therefore along with one step prediction we also considered five and eight step prediction problems ($T = 1, T = 5, T = 8$).

To predict if $x(t + T) > x(t)$ we use (as in Mukherjee et al. (1997)) a four dimensional vector of observations on time series

$$x_t = (x(t-3), x(t-2), x(t-1), x(t)).$$

Our goal is to compare the SVM method for time series prediction with SVM+ that uses future events (observations) of series as privileged information. As privileged vectors x_t^* we consider

$$x_t^* = (x(t+T-2), x(t+T-1), x(t+T+1), x(t+T+2)).$$

In Table 1 we report on error rates of SVM and SVM+ (for both dSVM+ and X^*SVM+ methods) for three different problems (one step five steps and eight steps ahead predictions ($T = 1, 5, 8$), and four sizes of training sets: $\ell = 100, 200, 400, 500$. For model selection, we used a validation set of size 500. It also shows the Oracle SVM error rates and approximation to the Bayesian error rate (≈Bayes).

To evaluate closeness of the obtained error rate to Oracle SVM and the Bayesian, we first found an approximation to the Bayesian rule

$$f_0(z) = (w_0, z) + b_0 = \sum_{i=1}^{\ell} \alpha_i^0 y_i K(x_i, x) + b_0,$$

using the SVM solution for a large data set (10,000 examples). Since the SVM error rate converges to the Bayesian error rate we consider the obtained rule as an approximation to the Bayesian rule and its error rate as the Bayesian error rate. Then using $f_0(z)$ we constructed the Oracle slacks

$$r_i = y_i f_0(z_i)$$

and using the technique described in the Section 2.1 calculated the Oracle SVM error rate.

7.3. Holistic description as privileged information

In this example we consider the digit recognition problem of classifying images of digits 5 and 8 in the MNIST database. This database describes digits as vectors in the 28×28 pixel images and contains 5.522 and 5.652 images of 5 and 8, respectively. Distinguishing between these two digits in 28×28 pixel space is an easy problem. To make it more difficult we resized the digits to 10×10 pixel images. A sample of 28×28 images and corresponding 10×10 images are shown in Fig. 4. We used 100 examples of 10×10 images as a training set, 4000 as a validation set (for tuning the parameters in SVM and SVM+) and the rest 1866 as the test set (Vapnik et al., 2008).

For every training image we created its holistic (poetic) description (Vapnik et al., 2008). A poetic description for the first

[4] There also exists a third decision "hold" which we do not consider here.

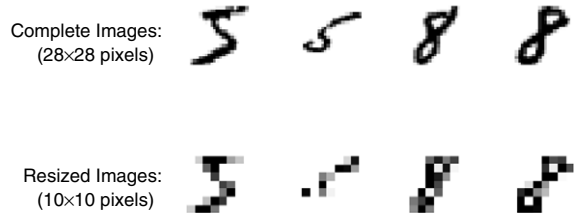

Fig. 4. Sample digits along with their resized images.

image of 5 (see Fig. 4) is as follows:

> *Not absolute two-part creature. Looks more like one impulse. As for two-partness the head is a sharp tool and the bottom is round and flexible. As for tools it is a man with a spear ready to throw it. Or a man is shooting an arrow. He is firing the bazooka. He swung his arm, he drew back his arm and is ready to strike. He is running. He is flying. He is looking ahead. He is swift. He is throwing a spear ahead. He is dangerous. It is slanted to the right. Good snaked-ness. The snake is attacking. It is going to jump and bite. It is free and absolutely open to anything. It shows itself, no kidding. Its bottom only slightly (one point!) is on earth. He is a sportsman and in the process of training. The straight arrow and the smooth flexible body. This creature is contradictory - angular part and slightly roundish part. The lashing whip (the rope with a handle). A toe with a handle. It is an outside creature, not inside. Everything is finite and open. Two open pockets, two available holes, two containers. A piece of rope with a handle. Rather thick. No loops, no saltire. No hill at all. Asymmetrical. No curlings.*

A poetic description for the first image of 8 (see Fig. 4) is as follows:

> *Two-part creature. Not very perfect infinite way. It has a deadlock, a blind alley. There is a small right-hand head appendix, a small shoot. The right-hand appendix. Two parts. A bit disproportionate. Almost equal. The upper one should be a bit smaller. The starboard list is quite right. It is normal like it should be. The lower part is not very steady. This creature has a big head and too small bottom for this head. It is nice in general but not very self-assured. A rope with two loops which do not meet well. There is a small upper right-hand tail. It does not look very neat. The rope is rather good - not very old, not very thin, not very thick. It is rather like it should be. The sleeping snake which did not hide the end of its tail. The rings are not very round - oblong - rather thin oblong. It is calm. Standing. Criss-cross. The criss-cross upper angle is rather sharp. Two criss-cross angles are equal. If a tool it is a lasso. Closed absolutely. Not quite symmetrical (due to the horn).*

Poetic descriptions were translated into 21-dimensional feature vectors. A subset of these features (with range of possible values) is: two-part-ness (0 - 5); tilting to the right (0 - 3); aggressiveness (0 - 2); stability (0 - 3); uniformity (0 - 3), and so on. The values of these features (in the order they appear above) for the first 5 and 8 are [2, 1, 2, 0, 1], and [4, 1, 1, 0, 2], respectively. Poetic descriptions and their translations were created prior to the learning process by an independent expert. The data for the digits, their poetic descriptions with corresponding feature vectors, and their ying-yang style descriptions with corresponding feature vectors are publicly available at www.nec-labs.com/research/machine/ml_website/department/software/learning-with-teacher.

Our goal was to construct a decision rule for classifying 10x10 pixel images using the 100 dimensional pixel space X and the corresponding 21-dimensional vectors in the space X^*. This idea was realized using the SVM+ algorithm in the two forms X^*SVM+ form and in dSVM+ form, described in the Section 2.3. For every training data size, 12 different random samples selected from the training data were used and we report the average of test errors.

Fig. 5. (a)Error rates of SVM+ on the digit recognition task. (b) Plot between deviation values from the decision rule in poetic space and corresponding correcting function values. This representative plot was generated for a training sample of size 70.

Results of using different correction spaces (21-dimensional poetic space and 1-dimensional space of deviation values) in SVM+ are shown in Fig. 5(a). The error rate of SVM trained and tested on 10×10 digits is shown by the line marked with circles. Error rate of using 21-dimensional poetic space as correction space (dSVM+ form) is shown by the line marked with crosses. Error rate using deviation values in the poetic space as correction space (X^*SVM+ form) is shown by the line with stars. In both cases the use of privileged information improves performance.

Fig. 5(b) shows the functional relationship between the deviation values defined in the poetic space and the values of the correcting function.

To understand how much information is contained in poetic descriptions, we conducted the following experiment. We used 28×28 pixel digits (784 dimensional space) instead of the 21-dimensional poetic descriptions in dSVM+ and X^*SVM+ methods in SVM+ (results shown in Fig. 6). In both the settings, using the 28×28 pixel description of digits SVM+ performs worse than SVM+ using poetic descriptions.

7.4. Analysis of the experimental results

1. *Classification of protein families.* In experiments on prediction of homology between protein sequences we used 3D structures of proteins as privileged information. This is a very strong