KERNEL FUSION AND FEATURE SELECTION IN MACHINE LEARNING

Vadim Mottl, Olga Krasotkina, Oleg Seredin
Computing Center of the Russian Academy of Sciences,
119991, Vavilov Str., 40, Moskow, Russia,
vmottl@yahoo.com, ko180177@yandex.ru,
oseredin@yandex.ru

Ilya Muchnik
DIMACS, Rutgers University,
P.O. Box 8018, Piscataway, NJ 08855,USA
muchnik@dimacs.rutgers.edu

ABSTRACT
In machine learning, when the kernel-based approach is used for estimating dependences in a set of entities, in particular, for solving the problems of pattern recognition or regression estimation, only one kernel function is tacitly assumed to be defined on the set of entities. At the same time, it is typical for practice that there are several viewpoints at the numerical pairwise comparison of entities. In this work, we systematically exploit the fact that any kernel function on a set of entities of arbitrary kind embeds that set into a linear space in which it plays the role of inner product. To fuse the kernels heuristically suggested by different experts into an entire learning technique, we propose to consider Cartesian product of the respective number of linear spaces, each supplied with a specific kernel as specific inner product. The main requirement placed upon the fusion principle is avoidance of discrete selection in eliminating redundant kernels with the purpose of achieving acceptable computational complexity. A real-valued feature on the given set of entities defines, actually, a simplest kernel, therefore, the proposed kernel fusion principle is, at the same time, a principle of minimizing the feature space dimensionality in feature-based machine learning.

KEY WORDS
Kernel fusion, feature selection, regression estimation, pattern recognition.

1. Introduction

The problem of finding empirical dependences $y(\omega):\Omega\to Y$ in a set of entities of arbitrary kind $\omega\in\Omega$ is one of the glowing problems of modern data mining. Let a given data set be the set of experimentally measured values of a characteristic $y_j=y(\omega_j)\in Y$ within an accessible subset of entities $\Omega^*=\{\omega_1,...,\omega_N\}\subset\Omega$. It is required to continue this function onto the entire set Ω for it would be possible to estimate this characteristic $\hat{y}(\omega)$ for entities $\omega\in\Omega\setminus\Omega^*$ not represented in the original (training) data set [1,2]. In particular, if $y(\omega)$ takes values from a finite set, for instance, $Y=\{-1,1\}$, the problem is usually called the pattern recognition problem, and in the case of a real-valued characteristic $Y=\mathbb{R}$ it is referred to as the problem of regression estimation. It is clear that the problem of function continuation is meaningless until some assumptions are taken about the relations between the values $y(\omega)$ and other characteristics of entities $\omega\in\Omega$ that are more accessible to observation than the goal characteristic. The modern machine learning theory is based on the so-called compactness hypothesis, which consists in the assumption that if two entities are close to each other in the sense of an appropriate metric then so are also, in most cases, the respective values of the goal characteristic.

The mathematically most advanced methods of machine learning essentially exploit the assumption that the universe of entities can be represented as a linear space. As the simplest instrument of introducing linear operations in the set of entities $\omega\in\Omega$, the vector of some observable numerical features $\mathbf{x}(\omega)\in\mathbb{R}^n$ was traditionally considered, and the Euclidean metric produced by it $\rho(\omega',\omega'')=\|\mathbf{x}(\omega')-\mathbf{x}(\omega'')\|$ served as the basis of function continuation in respective machine learning techniques.

It became apparent soon that what immediately determines the result of training is the configuration of the training-set points, represented in $\mathbb{R}^n$ by their pair-wise inner products $(\omega'\cdot\omega'')=\mathbf{x}^T(\omega')\mathbf{x}(\omega'')$, rather than the values of features. This observation resulted in the potential function method of machine learning [2], which later was named the kernel method [1]. The idea of a kernel $K(\omega',\omega'')$ consists in understanding it as inner product of two entities $K(\omega',\omega'')=(\omega'\cdot\omega'')$ in a linear space, maybe, a hypothetical one. If a kernel function $K(\omega',\omega'')$ is defined on an arbitrary set of entities Ω, it produces a Euclidean metric in it

$$\rho(\omega',\omega'')=\left[K(\omega',\omega')+K(\omega'',\omega'')-2K(\omega',\omega'')\right]^{1/2} \quad (1)$$

which expresses a specific compactness hypothesis without the intervening notion of features. There is usually much freedom in measuring dissimilarity of entities, and, thus, several kernels may be heuristically suggested within the bounds of the same data analysis problem. It should be emphasized that the choice of features $x_i(\omega)\in\mathbb{R}$, each of which defines, actually, a simplest kernel $K_i(\omega',\omega'')=x_i(\omega')x_i(\omega'')$, is also ever heuristic. The aim of this work is to study the ways of fusing the given set of kernels and to organize, thereby, a concurrence of several compactness hypotheses in finding empirical regularities in the given set of entities. To fuse several kernels, we propose to consider Cartesian product of the different linear spaces into which the given kernels embed the set of entities, just like the feature space $\mathbf{x}(\omega)=[x_i(\omega),i=1,...,n]\in\mathbb{R}^n$ is

This work is supported by the Russian Foundation for Basic Research (Grants 02-01-00107 and 05-01-00679), Grant of the President of the Russian Federation for young scientists No. MK-3173.2004.09 (O. Seredin), INTAS Grant 04-77-7347, and NSF Grant CCR 0325398 (I. Muchnik).

Cartesian product of the one-dimensional linear spaces produced by single features. However, there are many ways of constructing linear functions in such a combined linear space. Some of these ways are experimentally tested in this paper on a set of simulated data.

The main requirement placed upon the fusion principle is to avoid discrete selection of kernels with the purpose of achieving acceptable computational complexity of the fusion algorithm. We use here the main idea of embedding the discrete problem of choosing a subset into a continuous problem of finding optimal nonnegative weights assigned to the elements of the initial set. This idea was originally proposed in [3] as a means of constructing Relevance Vector Machines (RVM).

2. The linear space produced by a kernel

A kernel $K(\omega',\omega'')$ on a set of entities of arbitrary kind $\omega \in \Omega$ can be defined as a real-valued function $\Omega \times \Omega \to \mathbb{R}$ possessing two properties – symmetry $K(\omega',\omega'') = K(\omega'',\omega')$ and positive semi-definiteness of the matrix $\left[K(\omega_i,\omega_j); i,j=1,...,m\right]$ for any finite collection of entities $\{\omega_1,...,\omega_m\} \subset \Omega$. The function $\rho(\omega',\omega'')$ (1) produced by a kernel is a metric [4], and, so, the set of entities Ω supplied with a kernel function becomes metric space. Any kernel function $K(\omega',\omega'')$ allows for mentally embedding the set Ω into a real linear space with inner product $\Omega \subseteq \tilde{\Omega}$. The null element $\phi \in \Omega$ and linear operations $\left(\omega'+\omega'': \tilde{\Omega}\times\tilde{\Omega}\to\tilde{\Omega}\right)$ and $\left(c\omega: \mathbb{R}\times\tilde{\Omega}\to\tilde{\Omega}\right)$ are defined in $\tilde{\Omega}$ in a special way, whereas the role of inner product is played by the kernel function itself $(\omega',\omega'') = K(\omega',\omega'')$. As the basis for introducing linear operations in the extended set $\tilde{\Omega}$, may serve the notion of coaxiality of elements in a metric space [4].

Let $<\omega',\omega''>$ be an ordered pair of elements $\omega',\omega'' \in \tilde{\Omega}$. We shall say that the element $\omega \in \tilde{\Omega}$ is coaxial to the pair $<\omega',\omega''>$ with coefficient $c \in \mathbb{R}$ if $\rho(\omega',\omega) = |c|\rho(\omega',\omega'')$ and $\rho(\omega'',\omega) = |1-c|\rho(\omega',\omega'')$. This fact will be denoted by the symbol $\omega = \mathrm{coax}\left(<\omega',\omega''>; c\right)$. The triangle inequality turns into equality for any three coaxial elements ω', ω'' and ω. A metric space will be said to be unboundedly convex if for any ordered pair $<\alpha',\alpha''>$ and any $c \in \mathbb{R}$ it contains at least one element coaxial to this pair with coefficient c. It is proved in [4] that the coaxial element is unique if the metric forming an unboundedly convex metric space is produced by a kernel function (1). Such metric spaces are called Euclidean metric spaces. It is assumed that the given set of entities Ω is embedded into a greater set $\Omega \subseteq \tilde{\Omega}$ in which a kernel function is defined and which is, so, a Euclidean metric space.

It is possible to define linear operations in the Euclidean metric space $\tilde{\Omega}$ in the following way (see [4] for details):

- the null element is a hypothetical element $\phi \in \tilde{\Omega}$ for which $K(\phi,\phi) = 0$;
- multiplying by real coeffcient $c\omega = \mathrm{coax}\left(<\phi,\omega>; c\right)$;
- summation $\omega'+\omega'' = 2\mathrm{coax}\left(<\omega',\omega''>; 1/2\right)$;
- inner product and norm $(\omega'\cdot\omega'') = K(\omega',\omega'')$,

$$\|\omega\| = \sqrt{K(\omega,\omega)}\,. \qquad (2)$$

It is just this system of linear operations which is produced in the extended set $\tilde{\Omega}$ by a kernel function defined in the original set of entities $\Omega \subseteq \tilde{\Omega}$. The dimensionality of the linear space $\tilde{\Omega}$ is the maximum number of elements $\{\omega_1,...,\omega_m\} \subset \tilde{\Omega}$ for which the matrix $\left[K(\omega_i,\omega_j); i,j=1,...,m\right]$ can be positive definite. We do not study here the question of the dimensionality of this space, which may be finite or infinite, but this issue is extremely important for the generalization performance of the decision rules inferred from a training set.

3. The class of linear decision rules in the linear space produced by a kernel function

The convenience of a kernel function as a means of measuring dissimilarity of any two entities by the respective Euclidean metric (1) consists in that it involves the notion of a linear function $y(\omega): \Omega \to \mathbb{R}$ in the set of entities of any kind. This circumstance makes it possible to develop very simple algorithms of estimating dependencies between, generally speaking, arbitrary entities by exploiting, in the featureless situation, practically all known methods which had been worked up for linear spaces. In this Section, we consider the commonly adopted class of kernel-based decision rules as a class of linear functions in the extended set of entities $\tilde{\Omega}$ supplied with linear operations and inner product produced by a continuation of the given kernel function. The class of linear functions in $\tilde{\Omega}$ is defined by two parameters $\vartheta \in \tilde{\Omega}$ and $b \in \mathbb{R}$.

$$y(\omega \mid \vartheta, b) = K(\vartheta,\omega) + b\,,\ \omega \in \Omega\,. \qquad (3)$$

We shall call parameter ϑ the direction element of the linear function. If the real value of the linear function is immediately treated as the goal characteristic of an entity, the choice of parameters $\vartheta \in \tilde{\Omega}$ and $b \in \mathbb{R}$ determines a regression dependence. If the sign of the linear function is understood as the goal characteristic, the parameters specify a classification of the set of entities into two classes:

$$y(\omega) = K(\vartheta,\omega) + b > 0 \to \text{class 1},\ y(\omega) \le 0 \to \text{class 2}. \qquad (4)$$

Such a way of specifying a linear function may appear nonconstructive because it involves a hypothetical element of a linear space $\vartheta \in \tilde{\Omega}$ as direction element in (3), which is nothing else than product of our imagination. But when solving the problem of inferring the regression dependence or decision rule of pattern recognition from a training set $\left\{(y_j,\omega_j); j=1,...,N\right\}$ by the principles of Support Vector Machines [1] or Relevance Vector Machines [3], the only reasonable choice of ϑ will be a linear combination of really existing objects $\hat{\vartheta} = \sum_{j=1}^{N} a_j \omega_j$ in accordance with the linear

operations induced in the extended set $\tilde{\Omega}$ by the kernel function $K(\omega',\omega'')$ [5]. As inner product in $\tilde{\Omega}$, the kernel function is linear with respect to its arguments, hence, the linear function resulting from training will include the values of the kernel function only for objects existing in reality $\hat{y}(\omega) = \sum_{j=1}^{N} a_j K(\omega_j,\omega)$.

4. Cartesian product of linear spaces produced by several kernel functions

It is natural to expect that different experts skilled in the specific knowledge area will propose different kernel function. The main idea of this work is to shift the burden of the final choice onto the training algorithm by concurrently fusing the given set of heuristically chosen kernels. Let $K_i(\omega',\omega'')$, $i = 1,...,n$, be the kernel functions defined on the same set of entities $\omega \in \Omega$ by different experts. These kernel functions embed the set Ω into different linear spaces $\Omega \subset \tilde{\Omega}_i$, $i = 1,...,n$, with different inner products and, respectively, different linear operations. It is convenient to treat the n linear spaces jointly as Cartesian product

$$\tilde{\boldsymbol{\Omega}} = \tilde{\Omega}_1 \times ... \times \tilde{\Omega}_n = \left\{ \boldsymbol{\omega} = (\omega_1,...,\omega_n) : \omega_i \in \tilde{\Omega}_i \right\} \qquad (5)$$

formed by ordered n-tuples of elements from $\tilde{\Omega}_1,...,\tilde{\Omega}_n$. The kernel function (i.e. inner product) in this linear space can be defined as the sum of the kernel functions (inner products) of the corresponding components in any two n-tuples $\boldsymbol{\omega}' = (\omega_1',...,\omega_n')$ and $\boldsymbol{\omega}'' = (\omega_1'',...,\omega_n'')$:

$$(\boldsymbol{\omega}' \cdot \boldsymbol{\omega}'') = K(\boldsymbol{\omega}',\boldsymbol{\omega}'') = \sum_{i=1}^{n} K_i(\omega_i',\omega_i''),\ \boldsymbol{\omega}',\boldsymbol{\omega}'' \in \tilde{\boldsymbol{\Omega}}, \quad (6)$$

$$\| \boldsymbol{\omega} \| = \sqrt{K(\boldsymbol{\omega},\boldsymbol{\omega})} = \sqrt{\sum_{i=1}^{n} K_i(\omega_i,\omega_i)}\ . \qquad (7)$$

The dimensionality of the combined linear space $\tilde{\boldsymbol{\Omega}}$ (5) will not exceed the sum of dimensionalities of the particular linear spaces.

A really existing entity $\omega \in \Omega$ will be represented by its n-fold repetition $\boldsymbol{\omega} = (\omega,...,\omega) \in \tilde{\boldsymbol{\Omega}}$. Then any real-valued linear function $\Omega \to \mathbb{R}$ is specified by the choice of parameters $\boldsymbol{\vartheta} \in \tilde{\boldsymbol{\Omega}}$ and $b \in \mathbb{R}$

$$y(\omega) = K(\boldsymbol{\vartheta},\omega) + b = \sum_{i=1}^{n} K_i(\vartheta_i,\omega) + b\ , \qquad (8)$$

where $\boldsymbol{\vartheta}$ is a combination of hypothetical elements of particular linear spaces $\boldsymbol{\vartheta} = (\vartheta_1,...,\vartheta_n)$, $\vartheta_i \in \tilde{\Omega}_i$, produced by particular kernel functions $K_i(\omega',\omega'')$ in $\tilde{\Omega}_i$. Thus, to define a numerical dependence over a set of entities of any kind by combining several kernel functions $K_i(\omega',\omega'')$, we have, first of all, to choose, as parameters, one element in each of linear spaces $\vartheta_i \in \tilde{\Omega}_i$ into which the kernel functions embed the original set $\Omega \subseteq \tilde{\Omega}_i$. It should be marked that the less the norm of the i th parameter in its linear space $\| \vartheta_i \|^2 = K_i(\vartheta_i,\vartheta_i)$, the less the influence of the respective summand on the value of the function (8). If $K(\vartheta_i,\vartheta_i) \to 0$, i.e. $\vartheta_i \cong \phi_i \in \tilde{\Omega}_i$, the i th kernel will practically not affect the goal function.

This means that the parametric family of numerical functions (8) implies also an instrument of emphasizing "adequate" kernels with respect to the available observations and suppressing "inadequate" ones. Which kernels should be considered as adequate is the key question for providing a good generalization performance of the decision rule when it is applied to entities not represented in the training set.

5. The principle of kernel fusion

Let $\left\{ (\omega_j, y_j); y_j \in \mathbb{R}, j = 1,...,N \right\}$ or $\left\{ (\omega_j, g_j); g_j \in \{-1, 1\}, j = 1,...,N \right\}$ be the training set from which the linear goal function (8) is to be inferred for the problem of, respectively, regression estimation or pattern recognition. If the total dimensionality of the combined extended linear space $\tilde{\boldsymbol{\Omega}}$ (5) is greater than the number of entities in the training set, there always exist linear functions (8) that exactly reproduce the trainer's data $K(\boldsymbol{\vartheta},\omega_j) + b = y_j$ for all $j = 1,...,N$ in the problem of regression estimation or $K(\boldsymbol{\vartheta},\omega_j) + b \geq const$ in the problem of pattern recognition. Following the widely adopted principle [1], we shall prefer the function with the minimum norm of the direction element $\| \boldsymbol{\vartheta} \| \to \min$ under the constraints of the training set.

However, the norm in $\tilde{\boldsymbol{\Omega}}$ can be measured in several ways. The simplest version of norm follows from (7) $\| \boldsymbol{\vartheta} \|^2 = \sum_{i=1}^{n} K_i(\vartheta_i,\vartheta_i)$, but any linear combination of kernel functions with nonnegative coefficients also possesses all the properties of norm $\| \boldsymbol{\vartheta} \|^2 = \sum_{i=1}^{n} (1/r_i) K_i(\vartheta_i,\vartheta_i)$, $r_i \geq 0$. We come to the criterion

$$\begin{cases} \sum_{i=1}^{n} (1/r_i) K_i(\vartheta_i,\vartheta_i) \to \min(\vartheta_i \in \tilde{\Omega}_i), \\ \sum_{i=1}^{n} K_i(\vartheta_i,\omega_j) + b = y_j, \\ \text{or } g_j \left[\sum_{i=1}^{n} K_i(\vartheta_i,\omega_j) + b \right] \geq const,\ j = 1,...,N. \end{cases} \qquad (9)$$

If the penalty parameters $_i$ are equal to each other, for instance, $r_1 = ... = r_n = 1$, we have the "usual" training mode. In particular, criterion (9) with constraints $g_j \left[\sum_{i=1}^{n} K_i(\vartheta_i,\omega_j) + b \right] \geq const$ represents the original support vector method of pattern recognition [1]. With different penalty parameters for different kernels, the criterion (9) will try to avoid kernels with small $_i$. If $r_i \to 0$ then $(1/r_i) \to \infty$, the norm of the i th direction element is drastically penalized, and the respective kernel will not participate in forming the goal function.

The idea of adaptive training consists in jointly inferring the direction elements ϑ_i and penalty parameters $_i$ from the training set by additionally penalizing large weights [3]:

$$\begin{cases} \sum_{i=1}^{n} \left[(1/r_i) K_i(\vartheta_i,\vartheta_i) + \log r_i \right] \to \min(\vartheta_i, r_i), \\ \sum_{i=1}^{n} K_i(\vartheta_i,\omega_j) + b = y_j \\ \text{or } g_j \left[\sum_{i=1}^{n} K_i(\vartheta_i,\omega_j) + b \right] \geq \text{const},\ j = 1,...,N. \end{cases} \qquad (10)$$

As it will be reported in Section 0, these adaptive training criteria display a pronounced tendency to emphasize the kernel functions which are "adequate" to the trainer's data, and to suppress the "redundant" ones by large penalty weights $(1/r_i)$.

The reasoning for the adaptive training criterion (10) is the view on the unknown direction elements $\vartheta_i \in \tilde{\Omega}_i$ in each of the linear spaces $\tilde{\Omega}_i$ as hidden independent random variables whose mathematical expectations coincide with the respective null elements $\mathrm{M}(\vartheta_i) = \phi_i \in \tilde{\Omega}_i$. The parameter $_i$ has the sense of the unknown mean-square distance of the random direction element from the null element in the sense of metric (1). Then (10) is equivalent to finding the joint maximum-likelihood estimate of the variables $\vartheta_1,...,\vartheta_n$ and their variances $r_1,...,\ r_n$ under the additional assumption that the dimensionality of each of linear spaces $\tilde{\Omega}_i$ is, maybe, very large but finite, and the norm $\| \vartheta_i \| = \sqrt{K_i(\vartheta_i, \vartheta_i)}$ of the respective direction element is *a priori* normally distributed in it with zero mathematical expectation:

$$p(\vartheta_i) = r_i^{-1/2} (2\pi)^{-1/2} \exp\left(-(1/2r_i) K_i(\vartheta_i, \vartheta_i)\right).$$

Such a density is analogous to the normal zero-mean density of a real-valued random variable:

$$p(x) = r^{-1/2} (2\pi)^{-1/2} \exp\left(-(1/2r) x^2\right).$$

6. Fusion algorithms for preset penalty parameters

We consider first the algorithms of solving the kernel fusion problems (9) for both regression estimation and pattern recognition with preset penalty parameters $_i$. The ranges of definition of variables $\vartheta_i \in \tilde{\Omega}_i$ are linear spaces, even if hypothetical ones, therefore, from the formal point of view, we have the problems of quadratic optimization in these spaces under, respectively, equality or inequality constrains, i.e. quadratic programming problems.

We start with the problem of pattern recognition which is defined by inequality constraints. It is out of significance which value is given to the constant, so, we shall put $const = 1$. The problem is equivalent to finding the saddle point of the Lagrangian:

$$L(\vartheta_1,...,\vartheta_n, b, \lambda_1,...,\lambda_N) = \frac{1}{2}\sum_{i=1}^{n} (1/r_i) K_i(\vartheta_i, \vartheta_i) - \sum_{j=1}^{N} \lambda_j \left[g_j \left(\sum_{i=1}^{n} K_i(\omega_j, \vartheta_i) + b \right) - 1 \right] \begin{cases} \to \min(\vartheta_i \in \tilde{\Omega}_i, b \in \mathbb{R}), \\ \to \max(\lambda_j \geq 0). \end{cases} \quad (11)$$

Let the Lagrange multipliers $\lambda_1,...,\lambda_N$ be fixed, then the Lagrangian is to be minimized by $\vartheta_1,...,\vartheta_n, b$. Since $\vartheta_i \in \tilde{\Omega}_i$ are elements of abstract linear spaces, for minimizing the Lagrangian we have to use the notion of Frechet differential instead of that of gradient [6]. The Frechet differential of a real-valued function over a linear space is element of this space: $\nabla_\vartheta K(\vartheta, \omega) = \omega$, $\nabla_\vartheta K(\vartheta, \vartheta) = 2\vartheta$. The minimum conditions for $\vartheta_1,...,\vartheta_n$ are the equations

$$\nabla_{\vartheta_i} L(\vartheta_1,...,\vartheta_n, b, \lambda_1,...,\lambda_N) = (1/2)(1/r_i) \nabla_{\vartheta_i} K_i(\vartheta_i, \vartheta_i) - \sum_{j=1}^{N} \lambda_j g_j \nabla_{\vartheta_i} K_i(\omega_j, \vartheta_i) = (1/r_i)\vartheta_i - \sum_{j=1}^{N} \lambda_j g_j \omega_j = \phi \in \tilde{\Omega}_i,$$

whence it follows that

$$\vartheta_i = r_i \sum_{j=1}^{N} \lambda_j g_j \omega_j \quad (12)$$

at the minimum point. As we see, the optimal values of abstract variables $\vartheta_i \in \tilde{\Omega}_i$ are linear combinations of the entities of the training set in the sense of linear operations induced by the kernel functions as inner products in the respective linear spaces. Differentiation by b does not offer any difficulty, and we obtain

$$(\partial/\partial b) L(\vartheta_1,...,\vartheta_n, b, \lambda_1,...,\lambda_N) = -\sum_{j=1}^{N} \lambda_j g_j = 0 \quad (13)$$

Substitution of (12)-(13) into the Lagrangian (11) eliminates $\vartheta_1,...,\vartheta_n, b$ and leads to the optimization problem of finding Lagrange multipliers $\lambda_1,...,\lambda_N$:

$$\begin{cases} W(\lambda_1,...,\lambda_N) = \sum_{j=1}^{N} \lambda_j - \sum_{j=1}^{N} \sum_{l=1}^{N} \left(g_j g_l \sum_{i=1}^{n} r_i K_i(\omega_j, \omega_l) \right) \lambda_j \lambda_l \to \max, \\ \sum_{j=1}^{N} g_j \lambda_j = 0, \ \lambda_j \geq 0, \ j = 1,...,N. \end{cases}$$

This quadratic programming problem is a version the central problem of the well-known support vector method in machine learning [1].

The entities ω_j whose Lagrange multipliers are positive $\lambda_j > 0$ correspond to active constraints $g_j \sum_{i=1}^{n} K_i(\vartheta_i, \omega_j) + b \geq 1$ and are called support elements of the training set. In accordance with (12), only support elements participate in forming the optimal estimates of direction elements $\vartheta_i \in \tilde{\Omega}_i$ in the combined discriminant hyperplane (8):

$$\hat{y}(\omega) = \sum_{j:\lambda_j > 0} \lambda_j g_j \sum_{i=1}^{n} r_i K_i(\omega_j, \omega) + b \begin{cases} > 0 \to \text{class 1}, \\ < 0 \to \text{class 2}. \end{cases} \quad (14)$$

The value of parameter b immediately follows from the active constraints $g_j \sum_{i=1}^{n} K_i(\vartheta_i, \omega_j) + b = 1$ at the support elements $\lambda_j > 0$. Multiplication of each active constraint by $\lambda_j g_j$ gives the equalities $\lambda_j \sum_{i=1}^{n} K_i(\vartheta_i, \omega_j) + b = \lambda_j g_j$, which may be summed up over the entire training set because $\lambda_j = 0$ for other elements. We obtain in accordance with (13)

$$\sum_{j=1}^{N} \left(\lambda_j \sum_{i=1}^{n} K_i(\vartheta_i, \omega_j) + b \right) = \sum_{j=1}^{N} \lambda_j g_j = 0,$$

whence it follows with respect to (12) that

$$b = -\left(\sum_{j:\lambda_j > 0} \lambda_j \sum_{l:\lambda_l > 0} \lambda_l g_l \sum_{i=1}^{n} r_i K_i(\omega_j, \omega_l) \Big/ \sum_{j:\lambda_j > 0} \lambda_j \right). \quad (15)$$

In the problem of regression estimation the Lagrangian differs from (11) by constraints in (9), which are equalities instead of inequalities:

$$L(\vartheta_1,...,\vartheta_n, b, \lambda_1,...,\lambda_N) = (1/2) \sum_{i=1}^{n} (1/r_i) K_i(\vartheta_i, \vartheta_i) - \sum_{j=1}^{N} \lambda_j \left(\sum_{i=1}^{n} K_i(\omega_j, \vartheta_i) + b - y_j \right) \begin{cases} \to \min(\vartheta_i \in \tilde{\Omega}_i, b \in \mathbb{R}), \\ \to \max(\lambda_j \in \mathbb{R}). \end{cases} \quad (16)$$

The minimum conditions for $\vartheta_1,...,\vartheta_n$

$$\nabla_{\vartheta_i} L(\vartheta_1,...,\vartheta_n, b, \lambda_1,...,\lambda_N) = \frac{1}{2}(1/r_i) \nabla_{\vartheta_i} K_i(\vartheta_i, \vartheta_i) - \sum_{j=1}^{N} \lambda_j \nabla_{\vartheta_i} K_i(\omega_j, \vartheta_i) = (1/r_i)\vartheta_i - \sum_{j=1}^{N} \lambda_j \omega_j = \phi \in \tilde{\Omega}_i$$

give the representation of optimal direction elements as linear combination of the training set elements:

$$\vartheta_i = r_i \sum_{j=1}^{N} \lambda_j \omega_j \,. \qquad (17)$$

The minimum condition for b:

$$(\partial/\partial b) L(\vartheta_1,...,\vartheta_n,b,\lambda_1,...,\lambda_N) = -\sum_{j=1}^{N} \lambda_j = 0 \,. \qquad (18)$$

Substitution of (17) and (18) into (16) leads to quadratic optimization problem

$$\begin{cases} W(\lambda_1,...,\lambda_N) = \sum_{j=1}^{N} y_j \lambda_j - \sum_{j=1}^{N}\sum_{l=1}^{N} \left(\sum_{i=1}^{n} r_i K_i(\omega_j,\omega_l) \right) \lambda_j \lambda_l \to \max, \\ \sum_{j=1}^{N} \lambda_j = 0 \end{cases}$$

with Lagrangian

$$M(\lambda_1,...,\lambda_N,\mu) = \sum_{j=1}^{N} y_j \lambda_j - (1/2) \sum_{j=1}^{N}\sum_{l=1}^{N} \left(\sum_{i=1}^{n} r_i K_i(\omega_j,\omega_l) \right) \lambda_j \lambda_l - \mu \sum_{j=1}^{N} \lambda_j .$$

Its stationary point $(\partial/\partial\lambda_j) M(\lambda_1,...,\lambda_N,\mu) = 0$ for all j and $(\partial/\partial\mu) M(\lambda_1,...,\lambda_N,\mu) = 0$ yields the system of $N+1$ linear equations with $N+1$ variables:

$$\begin{cases} \sum_{l=1}^{N} \left(\sum_{i=1}^{n} r_i K_i(\omega_j,\omega_l) \right) \lambda_l + \mu = y_j, \; j = 1,...,N, \\ \sum_{j=1}^{N} \lambda_j = 0. \end{cases} \qquad (19)$$

Solution of this system $\lambda_1,...,\lambda_N,\mu$ immediately gives the optimal direction elements (17). If we substitute (17) into equality constraints in (9), we obtain

$$\sum_{l=1}^{N} \left(\sum_{i=1}^{n} r_i K_i(\omega_l,\omega_j) \right) \lambda_l + b = y_j$$

Comparison of these equalities with (19) shows that

$$b = \mu \,. \qquad (20)$$

As a result, we obtain the optimal estimate of the regression function (8):

$$\hat{y}(\omega) = \sum_{j=1}^{N} \lambda_j \sum_{i=1}^{n} r_i K_i(\omega_j,\omega) + b \,. \qquad (21)$$

Thus, we have completely constructed the algorithms of solving the kernel fusion problems for both pattern recognition and regression estimation in the case of preset penalty weights $1/r_i$ in (9).

7. Adaptive fusion algorithms

In the criterion of adaptive kernel fusion (10), the penalty weights $1/r_1,...,1/r_n$ are considered as unknown and should be inferred from the training set along with parameters $\vartheta_1,...,\vartheta_n,b$ of the combined regression function (8). We shall use the Gauss-Seidel iteration for jointly finding the optimal values of both groups of variables:

$$\begin{cases} (\vartheta_1^{(k+1)},...,\vartheta_n^{(k+1)}) = \arg\min_{\vartheta_1,...,\vartheta_n} \sum_{i=1}^{n} (1/r_i^{(k)}) K_i(\vartheta_i,\vartheta_i) \\ \sum_{i=1}^{n} K_i(\vartheta_i,\omega_j) + b = y_j \\ \text{or } g_j \left[\sum_{i=1}^{n} K_i(\vartheta_i,\omega_j) + b \right] \ge \text{const}, \; j = 1,...,N, \end{cases} \qquad (22)$$

$$r_i^{(k+1)} = \arg\min_{r} \left[(1/r) K_i(\vartheta_i^{(k+1)},\vartheta_i^{(k+1)}) + \log r \right], \qquad (23)$$

starting with equal penalty parameters $r_1 = ... = r_n = 1$. The algorithms of solving both versions of problem (22) are presented in the previous Section. At each step of the iteration process, the algorithms give the Lagrange multipliers $(\lambda_1^{(k+1)},...,\lambda_N^{(k+1)})$ or $(\lambda_1^{(k+1)},...,\lambda_N^{(k+1)},\mu^{(k+1)})$ which completely define the representation of direction elements in the discriminant hyperplane (12) $\vartheta_i^{(k+1)} = r_i^{(k)} \sum_{j=1}^{N} \lambda_j^{(k+1)} g_j \omega_j$ or regression function (17) $\vartheta_i^{(k+1)} = r_i^{(k)} \sum_{j=1}^{N} \lambda_j^{(k+1)} \omega_j$. It remains to use these representations for solving the optimization problem (23). Differentiation of the objective function gives the same equation for both pattern recognition and regression estimation

$$(\partial/\partial r) \left[(1/r) K_i(\vartheta_i^{(k+1)},\vartheta_i^{(k+1)}) + \log r \right] = (1/r) \left[-(1/r) K_i(\vartheta_i^{(k+1)},\vartheta_i^{(k+1)}) + 1 \right] = 0,$$

whence it follow that $r_i^{(k+1)} = K_i(\vartheta_i^{(k+1)},\vartheta_i^{(k+1)})$, i.e.

$$r_i^{(k+1)} = (r_i^{(k)})^2 \sum_{j:\lambda_j^{(k+1)}>0} \sum_{l:\lambda_l^{(k+1)}>0} g_j g_i \lambda_j^{(k+1)} \lambda_l^{(k+1)} K_i(\omega_j,\omega_l)$$

in the problem of pattern recognition, and

$$r_i^{(k+1)} = (r_i^{(k)})^2 \sum_{j=1}^{N} \sum_{l=1}^{N} K_i(\omega_j,\omega_l) \lambda_j^{(k+1)} \lambda_l^{(k+1)}$$

in the problem of regression estimation.

As a whole, the mutually analogous iteration processes give a succession of approximations to the optimal discriminant hyperplane

$$\hat{y}^{(k+1)}(\omega) = \sum_{j=1}^{N} \lambda_j^{(k+1)} g_j \sum_{i=1}^{n} r_i^{(k+1)} K_i(\omega_j,\omega) + b^{(k+1)} \begin{cases} >0, \\ <0, \end{cases}$$

or the optimal regression function

$$\hat{y}^{(k+1)}(\omega) = \sum_{j=1}^{N} \lambda_j^{(k+1)} \sum_{i=1}^{n} r_i^{(k+1)} K_i(\omega_j,\omega) + b^{(k+1)} \,.$$

As a rule, the process converges in 10-15 steps and displays a pronounced tendency to suppressing the weights at "redundant" kernel functions $r_i \to 0$ along with emphasizing $r_i >> 0$ the kernel functions which are "adequate" to the trainer's data. This fact provides a computationally effective selection of kernel functions without straightforward discrete choice of their subsets.

8. A particular case: Feature selection as kernel fusion

There is no insurmountable barrier between the featureless kernel-based way of forming parametric families of numerical functions on a set of entities of any kind and the usual parametric family of linear functions on the set of entities represented by vectors of their numerical features. The latter way is particular case of the former one. Indeed, a numerical feature $x(\omega): \Omega \to \mathbb{R}$ is equivalent to the simplest kernel function in the form of product $K(\omega',\omega'') = x(\omega')x(\omega'')$ that embeds the set of entities into a one-dimensional linear space $\Omega \subseteq \tilde{\Omega}$. Respectively, a vector of features $\mathbf{x}(\omega) = \left[x_1(\omega) \cdots x_n(\omega) \right]$ gives n kernel functions at once $K_i(\omega',\omega'') = x_i(\omega')x_i(\omega'')$ and n versions of such an embedding $\Omega \subseteq \tilde{\Omega}_i$. The choice of one entity in each of these spaces $\vartheta_i \in \tilde{\Omega}_i$, $i = 1,...,n$, namely, n real numbers $\left(x_1(\vartheta_1) \cdots x_n(\vartheta_n) \right) \in \mathbb{R}^n$, along with a numerical con-

stant $b \in \mathbb{R}$ specifies a linear function on the set of entities: $y(\omega) = \sum_{i=1}^{n} K_i(\vartheta_i, \omega) + b = \sum_{i=1}^{n} a_i x_i(\omega) + b$ where $a_i = x_i(\vartheta_i)$. The smaller the i th coefficient, i.e. the norm of the i th imaginary entity $\| \vartheta_i \| = x_i(\vartheta_i)$, the smaller is the contribution of this feature $x_i(\omega)$ to the value of the function.

9. Experimental results

As the essence of feature selection is shown to be the same as that of kernel fusion, we tested the proposed approach, for obviousness sake, on a set $\{(\mathbf{x}_j, y_j);\ j = 1,...,N\}$ of $N = 300$ pairs consisting of randomly chosen feature vectors $\mathbf{x}_j \in \mathbb{R}^n$, $n = 100$, and scalars obtained by the rule $y_j = a_1 x_{j,1} + a_2 x_{j,2} + \xi_j$ with $a_1 = a_2 = 1$ and ξ_j as normal white noise with zero mean value and some variance σ^2. So, only $n' = 2$ features of $n = 100$ were rational in the simulated data. In the experiment with regression estimation this set was taken immediately, whereas for the experiment with pattern recognition we took the set $\{(\mathbf{x}_j, g_j);\ j = 1,...,N\}$ where $g_j = -1$ if $y_j < 0$ and $g_j = 1$ if $y_j \geq 0$. In both experiments, we randomly chose $N_{tr} = 20$ pairs for training. So, the size of the training set was ten times greater than the number of rational features, but five times less than the full dimensionality of the feature vector. The remaining $N_{test} = 280$ pairs we used as the test set. The comparative results of training with equal weights at features $r_1 = ... = r_n = 1$ (9) and with adaptive weights (10) are presented in the following two tables:

Regression estimation
Error rate: ratio of the root-mean-square error in the test set to the actual root variance of the observation noise σ

Feature set	Training procedure	
	equal weights	adaptive weights
2 rational features	1.03	inapplicable
all 100 features	7.46	1.13

Pattern recognition
Error rate: misclassification percentage in the test set

Feature set	Training procedure	
	equal weights	adaptive weights
2 rational features	1.98%	inapplicable
all 100 features	28.0%	7.53%

As was expected, the classical training criterion with equal weights shows a drastic increase in the error rate in both cases when confusing features (i.e. confusing kernel functions) participate in training. At the same time, the error rate with weights adaptation is little sensitive to the presence of purely noisy features. In both experiments, the weights at redundant features turned practically into computer zeros after 10 iterations.

10. Conclusions

A numerical feature, when assigned to entities of a certain kind, embeds the set of these entities into a one-dimensional linear space. The essence of assigning a kernel function in a set of entities is also embedding it into a hypothetical linear space through the notion of coaxiality of elements of a Euclidean metric space.

The important difference is that the dimensionality of the space induced by a kernel function will be, as a rule, greater than one, if not infinite at all. The main point of the way of fusing several kernels is the idea to consider the Cartesian product of the respective linear spaces, just as the multidimensional feature space formed by a vector of features is the Cartesian product of the respective one-dimensional ones.

Thus, treating the universal set of "all feasible" entities as a linear space practically wipes out the difference between a set of kernels and a set of features and, so, between the featureless and feature-based approach to data analysis. The featureless multi-kernel approach replaces the problem of choosing the features by that of choosing the kernels. According to which of these two problems is easier, the feature-based or the featureless approach should be preferred.

However, fusing too many kernels, just as training with too many features, will inevitably worsen the generalization performance of the decision rule inferred from a small training set unless some regularization measures are taken. The technique of kernel selection proposed here is only one of possible principles of kernel fusion and has its shortcomings. In particular, such a technique should involve elements of testing on a separate set immediately in the course of training, for instance, on the basis of the leave-one-out principle.

References:

[1] V. Vapnik. *Statistical Learning Theory*. John-Wiley & Sons, Inc. 1998.

[2] M.A. Aizerman, E.M. Braverman, L.I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 1964, Vol. 25, pp. 821-837.

[3] Bishop C.M., Tipping M.E. Variational relevance vector machines. In: *C. Boutilier and M. Goldszmidt (Eds.), Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 2000, pp. 46–53.

[4] V.V. Mottl. Metric spaces admitting linear operations and inner product. *Doklady Mathematics*, Vol. 67, No. 1, 2003, pp. 140–143.

[5] V. Mottl, O. Seredin, S. Dvoenko, C. Kulikowski, I. Muchnik. Featureless pattern recognition in an imaginary Hilbert space. *Proceedings of the 15th International Conference on Pattern Recognition*. Quebec City, Canada, August 11-15, 2002.

[6] A.N. Kolmogorov, S.V. Fomin. *Introductory Real Analysis*. Prentice-Hall, Englewood Cliffs, 1970.