Graph Kernels – a Synthesis Note on Positive Definiteness

Cornelia Metzig[1], Gilles Bisson[1], Cécile Amblard[1], Mirta Gordon[1]

Université Joseph Fourier / Grenoble 1 / CNRS
Laboratoire LIG - Bâtiment CE4
Allée de la Palestine
38610 Gières FRANCE
cornelia.metzig@imag.fr, gilles.bisson@imag.fr, cecile.amblard@imag.fr,
mirta.gordon@imag.fr

Abstract :

We review the problem of extending the applicability of support vector machines (SVM) to graph data. Many similarity measures, generally called kernels, on graph data have been proposed in the last decade. Yet some of them, like the optimum assignment kernel (**?**), are not positive semidefinite, which limits their application in SVM. In this paper we recall the necessary conditions for using SVM. While the Mercer theorem gives necessary and sufficient conditions for vectorial data, we show that for graph data an embedding in a Hilbert space has to be defined explicitly, and that weaker conditions do not suffice. For several kernels proposed in the literature we demonstrate that an underlying Hilbert space does exist by specifying the corresponding basis. Our findings are illustrated with small examples from the graph kernel literature.

1. Introduction

Comparing graphs is a problem that arises in many fields of research, such as bio- or chemoinformatics, where large molecules with several thousands of atoms are represented as graphs. Data mining algorithms are being applied on databases of such graphs for purposes like drug discovery. Another important field is the structure analysis of social networks.

In the past decade many similarity measures for graphs have been proposed and studied, among many others (14)(18)(20). These similarity measures have been interpreted as kernels and have been used in Support Vector Machines (SVM). Due to what is called the "kernel trick", positive semidefinite kernels

defined on the input space can be used within the SVM algorithm, since they are scalar products in some feature space. Graphs however are non-vectorial data, so it is not evident how support vector machines can be used on them, and whether the kernel trick can be applied.

The aim of the present paper is to clarify under which necessary und sufficient condition similarity measures on graphs represent indeed scalar products in the feature space, i.e. are positive semidefinite kernels. We show why some kernels on graphs from the literature fulfill positive definiteness, and others do not. Unlike the related work of Shin et al. (11), our reasoning does not require the introduction of any new formalism. Having the special case of non-vectorial data in mind, a thorough look at the Mercer Theorem and the definition of a Reproducing Kernel Hilbert Space is enough to deduce a simple rule for checking positive semidefiniteness. The difficulty lies in the extension of a formalism from metric data to structured objects.

The paper is organized as follows. In section 2 we briefly recall support vector machines. In section 3, we present the two possible ways of extending SVM to graph data, by proving positive semidefiniteness of the corresponding kernel on graph data: either via Mercer's theorem or through a Reproducing Kernel Hilbert Space (RKHS) defined directly on the set of graphs. In section 4, we discuss examples from the literature of RKHS defined on graphs. Finally in section 5 we point out possible consequences of our results, then we conclude.

2. SVM algorithm and Mercer's theorem

Support vector machines are learning algorithms that determine a linear decision function $f : X \to Y$ from a given set of m labelled training data points $\{\mathbf{x}_i, y_i\}_{i=1}^m$, where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{\pm 1\}$. The predicted label of a test data point $\mathbf{x}$ depends on its position with respect to the separating hyperplane, according to

$$f(\mathbf{x}) = sgn\left[\mathbf{w} \cdot \mathbf{x} + b\right], \tag{1}$$

where $\cdot$ denotes the scalar product, $\mathbf{w}$ is a vector perpendicular to the hyperplane and b is its (conveniently normalized) distance from the origin. The function f maximizes the margin between the datapoints and the hyperplane and is found by solving a quadratic optimization problem under constraints. The vector $\mathbf{w}$ is a linear combination of vectors from the training data which are closest to the hyperplane. They are called support vectors. Thus,

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i, \tag{2}$$

summed over all support vectors, with $\alpha_i > 0$ for the support vectors and $\alpha_i = 0$ for other training vectors. The linear decision function f can be written as

$$f(\mathbf{x}) = sgn \left[\sum_i \alpha_i y_i \, \mathbf{x_i} \cdot \mathbf{x} + b \right]. \tag{3}$$

Since for equation 3, only scalar products between pairs of input vectors need to be defined, the SVM algorithm can be extended to vectorial data that is not linearly separable in the input space by means of Mercer's theorem, also known as "kernel trick".

2.1. The Mercer theorem

The kernel trick is based on the following theorem, originally stated by J. Mercer in 1909 (4). It states(1):

Let C be a compact subsect of $\mathbb{R}^n$. To guarantee that the continuous symmetric kernel function (or kernel) $K(\mathbf{x_1}, \mathbf{x_2})$ in $L_2(C)$ has an expansion (i.e. represents a scalar product)

$$K(\mathbf{x_1}, \mathbf{x_2}) = \sum_{k=1}^{\infty} a_k \Phi_k(\mathbf{x_1}) \Phi_k(\mathbf{x_2}), \tag{4}$$

(with positive coefficients a_k >0), it is necessary and sufficient that K be positive semidefinite, i.e. it fulfills the condition

$$\int_C \int_C g(\mathbf{x_1}) g(\mathbf{x_2}) K(\mathbf{x_1}, \mathbf{x_2}) d\mathbf{x_1} \, d\mathbf{x_2} \geq 0 \tag{5}$$

for all g of integrable square, i.e. $\in L_2(C)$.

The Φ_k in equation 4 are intuitively the "features" (1), i.e. the implicit mapping from the input space C to the feature space.

The great usefulness of this theorem lies in the fact that the mapping to the feature space does not need to be known explicitly, only an appropriate positive semidefinite kernel has to be defined on the entire input space[1]. For a more complete introduction to kernels and the kernel trick, see (2) and (3).

[1]Later, Dunford and Schwartz showed that the theorem holds on a compact metric space (5). In this article we do not consider further generalizations such as by (**?**) as they are

In practice, if the data is non-separable in the input space, we may either define explicitly a mapping into a feature space, or a kernel. In the first case, one needs to prove that the feature space is a vector space endowed with a scalar product. In the second case, one must merely prove that the kernel be positive semidefinite.

2.2. How to show positive definiteness

Showing that a function is positive semidefinite is the central difficulty when applying the Mercer theorem. Often, the integral in equation (5) cannot be evaluated explicitly. In that case, the proof of positive semidefiniteness often recurs to closure properties of positive definite functions (see (7), (8)). The following properties allow for the construction of new positive semidefinite kernels starting from existing ones. Let $\mathcal{X}$ be a nonempty set. It holds (among other properties)

– Closure under sum: For two positive-semidefinite symmetric kernels K_A, K_B: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the sum

$$K = K_A + K_B : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{6}$$

is a positive semidefinite symmetric kernel.

– Closure under product: For two positive-semidefinite symmetric kernels K_A, K_B: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$, the product

$$K = K_A \cdot K_B : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{7}$$

is a positive semidefinite symmetric kernel.

– Closure under tensor product: Let $K_A : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ and $K_B : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$ be positive-semidefinite symmetric kernels, $x_A \in \mathcal{A}$, $x_B \in \mathcal{B}$. Then their tensor product $K_A K_B : (\mathcal{A} \times \mathcal{B}) \times (\mathcal{A} \times \mathcal{B}) \to \mathbb{R}$, where

$$\begin{aligned} K_A K_B((x_{A1}, x_{B1}), (x_{A2}, x_{B2})) = \\ K_A(x_{A1}, x_{A2}) \Delta K_B(x_{B1}, x_{B2}) \end{aligned} \tag{8}$$

is a positive semidefinite symmetric kernel.

---

not relevant for our particular problem. In order to understand the reasoning of this paper, an unfamiliar reader can imagine an Euclidean space instead of a compact metric space as domain for the kernel.

– Closure under concatenation of functions: For two positive-semidefinite symmetric kernels K_A: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$, K_B: $\mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the concatenation

$$K = K_B \circ K_A : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{9}$$

is a positive semidefinite symmetric kernel.

These closure properties are convenient tools to show positive semidefiniteness of a kernel K if the positive semidefiniteness of the constituting functions K_A, K_B is known. This is often the case for typical kernels defined on $\mathbb{R}^n$, since the functions K_A, K_B can be expressed as linear combinations of the canonic scalar product on $\mathbb{R}^n$, which obviously is positive definite. For instance, this proof applies for kernel functions that can be written as a series expansion like the Gaussian kernel,

$$K(\mathbf{x_1}, \mathbf{x_2}) = exp(-(\|\mathbf{x_1} - \mathbf{x_2}\|^2/2\sigma), \tag{10}$$

or a product of kernels such as the exponential kernel

$$K(\mathbf{x_1}, \mathbf{x_2}) = (\alpha \mathbf{x_1} \cdot \mathbf{x_2} + c)^d. \tag{11}$$

Let us now address the case of learning algorithms. The calculation of a kernel for a specific sample of datapoints yields its Gram matrix. Recall that if a positive semidefinite kernel function is evaluated at a finite number of points m, it gives rise to a symmetric and positive semidefinite Gram matrix K_{ij} satisfying

$$\sum_{1 \le i \le m, 1 \le j \le m}^{m} c_i c_j K_{ij} \ge 0 \tag{12}$$

for all $c_i \in \mathbb{R}$, implying that its eigenvalues are non-negative. Positive definiteness of one particular Gram matrix does not imply positive definiteness of the kernel function sampled at other points. (A positive semidefinite Gram matrix calculated from all training points is only a valid proof for positive semidefinitess of a kernel if in addition two assumptions hold: the kernel function K be continuous in the input space, and the subset of m training points be dense in the test set.)

3. Extension of SVM to graph data

For data in a metric space, we have the tools to prove positive semidefiniteness of a kernel and thus to apply SVM. Since both the training data and the

(unknown) test data are in a metric space (the input space), for any chosen *continuous* kernel function we can prove via closure properties its positive semidefinitess *on the entire input space* and then apply the Mercer theorem. The kernel evaluated at *any* set of test points yields a positive semidefinite Gram matrix. Now the challenge is to prove positive semidefiniteness for graph kernels. Is the Mercer theorem a useful tool for kernel functions defined on the set of graphs? A priori no, since the set of graphs is not a metric space. Indeed, without any underlying metric, continuity of a kernel function is not defined, and thus cannot serve as a means to extend positive semidefiniteness from training data onto unknown test data. The purpose of this section is to try to embed the set of graphs in a metric space, such that positive semi-definiteness of kernels can be shown (e.g. by using closure properties), and extended to unknown data because of their continuity with respect to the defined metric. We present in sections 3.1. - 3.3. three different approaches to that aim, of which two are erroneous, and one solves indeed our problem.

3.1. Haussler's extension of Mercer kernels to non-vectorial data.

Haussler's convolution kernel has received much attention since it is a possible way of extending positive semidefinite kernels to structured data such as trees, strings and graphs. In his introductory technical report (10), Haussler also furnishes an attempt to define a metric on any set for the purpose of defining a continuous positive-semidefinite kernel on it (p. 30). A solution to this would exempt us from any further study of generalizing positive semidefiniteness to test data. The approach is as follows:

- Definition of a metric on the set, rendering it a separable metric space
- Choice of a symmetric positive semidefinite kernel function on a space
- Demonstration that the kernel is continuous on the set with respect to the defined metric
- Conclusion: all other datapoints in the set are vectors in the Hilbert space defined by the kernel

The reasoning – like the Mercer theorem – exploits the fact that continuity of the kernel function guarantees its positive semidefiniteness on the entire set (i.e. its domain). Haussler defines the metric $d(x_1, x_2) = \delta(x_1, x_2)$. His application shows that he intends $d(x_1, x_2) = 0$ if $x_1 \neq x_2$ and $d(x_1, x_2) = 1$

if $x_1 = x_2$. This is clearly not a metric, since it does not fulfil $d(x_1, x_1) = 0$. Yet, it seemingly allows to define a real continuous function on that metric space, fulfilling the following definition:

A function f: $X \to Y$ (X, $Y \subset \mathbb{R}$) is continuous if for all $\epsilon > 0$ there exists a $\rho > 0$ such that $d(x_1, x_2) < \rho \Rightarrow d(f(x_1), f(x_2)) < \epsilon$.

Trivially, if the distance between two distinct elements x_1 and x_2 is set to zero, $d(x_1, x_2) < \rho$ is true for any ρ, independently of ϵ, so the continuity criterion is always true. Haussler uses this argument to prove continuity of any kernel function defined on a set with that metric. As soon as a valid distance is used (e.g. what is commonly called discrete metric), his reasoning does not work any more, so we discard this proof. (However, Haussler's convolution kernel can nevertheless be positive semidefinite.)

3.2. Definition of a discrete metric on the set of graphs.

Let us pursue Haussler's basic idea and try a *well-defined* discrete metric. Any arbitrary countable set can be endowed with the discrete metric, such that for two elements x_1, x_2 the distance d between them is $d(x_1, x_2) = 1$ if $x_1 \neq x_2$ and $d(x_1, x_2) = 0$ if $x_1 = x_2$. In such a space, all distinct objects have the same distance to each other. To be able to apply the Mercer theorem, also compactness is required, which intuitively gives the notion of closeness of two elements. Compactness guarantees that if one takes infinitely many sample points, one will find at least one of them which is arbitrarily close to some other point of the space. This can be fulfilled for two reasons: because one finds a neighbourhood of the chosen point in which there are infinitely many sample points, or because the chosen point is itself sampled a second time (while taking infinitely many samples). The latter case necessarily happens if the space is finite. Thus, finite sets with a discrete metric are compact.

One may argue that in the domain of machine learning, all data sets are necessarily finite, so we only have to do with compact metric spaces. To understand why this answer is too simple consider the task that support vector machines shall solve: to learn a linear classifier from a training set in order to predict class labels for test data. When the kernel function is defined, the test data is completely unknown, so both training and test set together must be considered as infinite. The kernel needs to be positive semidefinite for any Gram matrix, arising from both training and test data.

The problem becomes clear if one imagines the training set embedded in an Euclidean space such that every element of the set constitutes the unit

vector of one dimension, i.e. for this set the Euclidean metric coincides with the discrete metric. Any separating hyperplane computed by SVM is only defined in the dimensions in which are the training set points. By definition the test set points are in its own dimensions, on which the classifier function is not defined, so for the function all test points appear at to be 'at the origin', and the hyperplane cannot classify the test data.

The only solution to this discrepancy would be to know all possible test data beforehand, and the positive semidefinite kernel could be defined on the entire data. The metric space would be finite and compact, yet no classification task would be left where support vector machines are needed.

We conclude from this and the preceding subsection that it is not possible to make a compact metric space out of the set of graphs for the purpose of supervised learning in a way that is universally valid. Yet, there is remaining the possibility of defining a Reproducing Kernel Hilbert Space (RKHS) on any arbitrary set in a less general way. We will show how such a space can be defined on the set of graphs in the following.

3.3. Defining a Reproducing Kernel Hilbert Space by the explicit mapping Φ

This approach is convenient if we cannot identify an underlying metric, since a kernel that defines a Reproducing Kernel Hilbert Space (RKHS) (9)(3) can be defined on any set – for example the set of graphs. It can be defined as follows:

Let $\mathcal{X}$ be a nonempty set, x its elements, and $\mathcal{F}$ be the set of functions f: $\mathcal{X} \to \mathbb{R}$. $\mathcal{F}$ is called a Reproducing Kernel Hilbert Space (RKHS) endowed with a scalar product $\langle ., . \rangle$ if there exists a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that K has the so-called reproducing property

$$\langle f, K(x, .) \rangle = f(x) \tag{13}$$

for all $f \in \mathcal{F}$ (and also $K(x, .) \in \mathcal{F}$). This implies in particular

$$\langle K(x_1, .), K(x_2, .) \rangle = K(x_1, x_2). \tag{14}$$

The definition states that for any positive semidefinite kernel on some set there exists an associated RKHS in which the kernel function is a scalar product. Therefore the kernel can be used in SVM. We know that a positive-semidefinite Gram matrix spans the space in which the matrix is diagonal.

As stated before, there is no representation of all datapoints of the set in that RKHS unless the kernel is positive semidefinite on the *entire* set.

How do we show positive semidefiniteness of kernels on an arbitrary set? We cannot formulate a kernel in terms of a continuous function if it only defined on a discrete set of points. The closure properties are not useful in this case. The only possibility left is to know explicitly the mapping Φ: $\mathcal{X} \to \mathbb{R}^n$ from the set to the feature space, which is equivalent to saying that the basis of the RKHS is known, not just the kernel that defines it. The extracted descriptors can be seen as basis vectors of a space in which each element of the set is represented as a vector (e.g. the examples in (14)), the scalar product of that space being directly in the SVM algorithm. Finding meaningful descriptors (or features) is the central difficulty in this approach, since other information about the elements gets lost. This proof of positive semidefiniteness applies to many graph kernels that have been proposed. Either the scalar product in the RKHS itself is used as the similarity measure, or another positive semidefinite kernel is defined in that space, using closure properties.

The essential fact that positive semidefiniteness of a kernel is fulfilled once the basis of the feature space is known (and the kernel is the scalar product in that space), has been formalised by Shin et Kuboyama (11) in a different way: their necessary and sufficient requirements for a positive definite kernel is that what they term *mapping system* be symmetric and transitive, implying that for any sample of data the same comparison criteria need to be used. Phrasing it in terms of the definition of SVM, these criteria are the extracted features, which span the dimensions of the feature space[2].

4. Examples of graph kernels

Here, for some classes of proposed graph kernels we prove their positive semidefiniteness by showing that the kernel is indeed a scalar product in the feature space, by explicitly enunciating its basis. In most cases this turns out to be possible. However, some graph kernels have been introduced without a proof of its positive semidefiniteness or with an invalid proof (15). We consider a graph to be a set of nodes (or vertices) and edges between nodes. In a labelled graph, both nodes and edges have labels. To simplify matters we will use examples in which only the nodes of the graphs carry labels.

The following graph kernels have been proposed for applications in chemoin-

[2]Shin and Kuboyama studied the particular case of tree kernels in (12).

formatics, where a graph corresponds to a molecule, a node to an atom or a functional group, and the different labels are the different chemical elements or names of the functional groups[3].

Generally speaking, the condition that a kernel has a basis in the feature space is fulfilled whenever the features extracted by the kernel are the number of common substructures. The features become the unit vectors of the feature space. Every graph has a representation in that space.[4]

4.1. Kernels where the extracted features are paths

These kernels typically count the number of common paths in pairs of graphs, i.e. sequences of node labels of a certain length (here 3). Consider the graphs in figure 4.1..

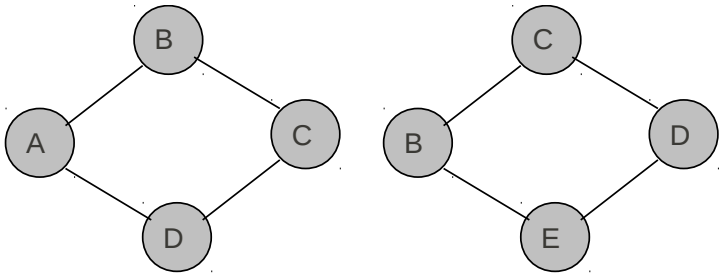

Figure 1: Two labelled graphs: $ex.(1)$ and $ex.(2)$

The set of these two graphs contains the following paths of length three: ABC, ADC, BAD, BCD, BED, CDE, CBE. If each of these 7 paths are considered as defining a dimension of the feature space, the graphs can be represented as vectors. $Example\ (1)$ becomes $(1, 1, 1, 1, 0, 0, 0)$, $example\ (2)$ $(0, 0, 0, 1, 1, 1, 1)$.

Larger graphs with repeated occurrence of a sequence can have integers > 1 as respective vector entries. The canonic scalar product of that space is clearly positive semidefite and can thus be used in SVM.

[3]We do not focus on differences in the implementation, e.g. whether a product graph formalism as in (**?**) is used or not.

[4]To be precise, the graphs are elements in a semiring (endowed with the Euclidean metric), since their vector representations have only positive integer components. Nevertheless, the hyperplane can have noninteger vector entries and thus lies in a semifield. The feature space is thus a half-space restricted to positive vector entries. An unfamiliar reader can think of a vector space in order to fully understand the reasoning.

A positive semidefinite graph kernel which has been widely used (e.g. (16)) is the "Tanimoto kernel" introduced by (14). The basis of its feature space is the one defined above. If $K(.,.)$ is the scalar product in this space, the Tanimoto kernel is defined as

$$K^t(x_1, x_2) = \frac{K(x_1, x_2)}{K(x_1, x_1) + K(x_2, x_2) - K(x_1, x_2)}. \tag{15}$$

(The Tanimoto kernel is the same as the so-called Jaccard similarity coefficient, yet the distance calculated by the Tanimoto kernel in the feature space is not the Jaccard distance (17)).

4.2. Kernels where the extracted features are walks of varying length

Another common kernel is the Random Walk Kernel (e.g. (13)), which counts the number of common walks (i.e. paths with repeated visits) on two graphs. The possible walks (of varying length) constitute the basis vectors of the feature space. For the example above these are A, B, C, D, E, AB, AD, BC, BE, CD, DE, ABA, ABC, ADA, ADC, BAB, BAD, BCB, BEB, BED, CBC, ... In this basis $ex.1$ becomes the vector $(1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, ...)$, $ex.2$ becomes $(0, 1, 1\ , 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, ...)$. For an unanimous representation of each graph several choices need to be taken, e.g. whether walking back and forth is allowed (then the components in the dimensions AB and ABA are necessarily the same) or whether only nonredundant paths are allowed (14). In this space, the random walk kernel resembles the standard scalar product, but the components in the dimensions spanned by walks of length k are downweighted by a decay factor $\lambda^k > 0$. It is important that the decay factor is chosen such that the scalar product converges.[5]

Another class of kernels represent graphs as vectors by extracting common subgraphs (10) (**?**). All possible subgraphs become the respective basis vectors of the feature space.

4.3. The Optimum Assignment Kernel

The optimum assignment kernel has been introduced in (15) as follows:

[5]However in practice, this is a minor problem, since any algorithm can only consider walks up to a certain length.

Let $\mathcal{X}$ be the set of graphs x, $\mathcal{Y}$ the set of nodes y. Let $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be some nonnegative, symmetric and positive semidefinite kernel, and π the permutation operator. Then $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with

$$K(x_1, x_2) = \max_{\pi} \sum_{i=1}^{|x_2|} k(y_{1i}, y_{2\pi(i)}) \tag{16}$$

is called an optimum assignment kernel. $|x|$ denotes the cardinality of the graph x, i.e. the number of its nodes, and we set x_2 to be graph with lower cardinality, i.e. $|x_2| \leq |x_1|$. The graph x_2 is constituted by the nodes $\{y_{2i}\}$.

The idea is to find the best matching of the smaller of the two graphs on the bigger one, and then calculate the sum of the kernels k between the vertex labels.

Figure 2: Two matching possibilities: How should the assignment be done?

Here, it is not possible to identify a basis as in the previous examples. The features used for comparison between two graphs are not the same for the entire dataset, but depend on each pair of graphs sampled for comparison. It has been shown that the optimum assignment kernel is not always positive semidefinite. (21) proves this by giving a counter-example, where the Gram matrix has indeed one negative eigenvalue.

Now, having the theory of RKHS in mind, it is easy to understand that the problem stems from the fact that the feature space is not well-defined. In the proof given by (15) of the positive semidefiniteness of the optimum assignment kernel, they use the closure property given in equation (9), stating that the sum $K = K_1 + K_2$ of two positive semidefinite kernels is again a positive semidefinite function.

This is only true if K, K_1 and K_2 are defined in the same space. This means, either

– they are all three kernels between nodes, or

– they are all three kernels between graphs.

In the latter case, the kernel k between nodes could be interpreted as being a kernel between graphs, whose extracted feature is the number of occurrence of a particular node label. In that case every different node label would span one dimension of the feature space, and information of the edges would be lost, and no matching would be performed. In contrast, in the optimum assignment kernel, there are as many addends to the kernel function as the number of nodes of the smaller of the two graphs x_1 and x_2, say n. One could imagine it as an n-dimensional space, where each node, not each node label, spans one dimension. But then it is clear that there exists a different space for every pair of compared graphs, and only by great chance one common feature space exists.

4.4. Indefinite kernels

Sometimes kernel functions that are not positive-semidefinite are used anyway, as the similarity measure "makes sense" for a specific application. They can yield good classification results (15)(22). To be able to use a non-positive kernel, it needs to be regularized, which means that it is modified such that it becomes positive definite. One way of doing this is to add a constant value to every eigenvalue in order to shift them to positive values, for instance done by (22). Another approach is to zeroize the negative eigenvalues (21). Often the algorithm is also used as it is, but stopped before convergence. This works particularly as long as the negative eigenvalues are few and have small absolute values, as it is often the case for the optimum assignment kernel. In fact, algorithms similar to the optimum assignment kernel yield very good classification results (23) without claiming to be a positive semidefinite kernel. Regularization methods generally flaw the classification results, yet the significance of the error depends on how many eigenvalues of the Gram matrix are negative, as well as on their absolute value.

Finally, questionable approaches such as pursued by (24) try to establish a mathematical trick that allows the application of indefinite kernels in support vector machines, but at some late step in their reasoning treat negative eigenvalues as positive ones. This is fatal and leads to the same paradoxes as encountered in special relativity in physics, where the bilinear form in spacetime has the eigenvalues 1 and -1. As its name already tells, the theory states

that no common reference frame exists, which in our case of SVM is explicitly needed. The visualized examples in (24) only work despite nonzero entries in the direction associated with the negative eigenvalue, not because of.

5. Conclusion

In this paper, we investigated the different approaches how to define a positive semidefinite kernel on the set of graphs in order to apply SVM. We recalled the SVM algorithm and Mercer's theorem, and tried to extend its applicability to graphs by defining a general metric on them, in an approach similar to (10). Our conclusion is that this cannot work in a general way, only at the price of losing some information about the graphs. We show that it is nevertheless possible to define a Reproducing Kernel Hilbert Space on the set of graphs by defining a basis consisting of the extracted features. With this method we showed positive semidefiniteness for many graph kernels introduced in the last decade. In the conflicting case of the optimum assignment kernel, this approach cannot work because no features are extracted, and thus no basis of a feature space can be identified. Phrasing it in terms of logic, our conclusion is that it is possible to show positive semidefiniteness only for a dataset that can be expressed in propositional logic, whereas working directly with a predicate logic representation, as the optimum assignment kernel did, is not sufficient. Finally we briefly discuss different approaches how in practice the problem of negative eigenvalues is dealt with.

References

[1] Vapnik, V.N.: The Nature of Statistical Learning Theory, Springer (1995)
[2] Vapnik, V.N.: Statistical Learning Theory, John Wiley and Sons (1998)
[3] Schölkopf, B., Smola, A.J.: Learning with Kernels, The MIT Press (2002)
[4] Mercer, J.: Functions of Positive and Negative Type, and their Connection with the theory of integral equations, Philosophical Transactions of the Royal Society of London, Series A 209, p. 415-446 (1909)
[5] Dunford, N., Schwartz, J.T.: Linear Operators, Part II: Spectral Theory, John Wiley and Sons (1963)
[6] Ferreira, J.C., Menegatto, V.A.: Eigenvalues of Integral Operators Defined by Smooth Positive Definite Kernels, Integral Equations ans Operator Theory 64 , p. 61 - 81 (2009)

[7] Berg, C., Christensen, J.P.R., Ressel, P.: Harmonic Analysis on Semigroups, Springer (1984)
[8] Cortes, C., Haffner, P., Mohri, M.: Rational Kernels: Theory and Algorithms, Journal of Machine Learning Research 1, p. 1-50 (2004)
[9] Aronszajn, N.: Theory of reprducing kernels, Transactions of the American Mathematical Society, Vol. 68, No. 3 p. 337-404 (1950)
[10] Haussler, D.: Convolution Kernels on Discrete Structures, Technical Report UCSC-CRL-99-10 (1999)
[11] Shin, K., Kuboyama, T.: A Generalization of Haussler's Convolution Kernel – Mapping Kernel, Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland (2008)
[12] Shin, K., Kuboyama, T.: A Generalization of Haussler's Convolution Kernel – Mapping Kernel and its application to Tree Kernels, Journal of Computer Science and Technology Nr. 25, p. 1040 - 1054 (2010)
[13] Kashima, H., Inokuchi, A.: Kernels for Graph Classification, IEEE International Conference on Data Mining (ICDM 2002) Workshop on Active Mining, Maebashi, Japan (2002)
[14] Ralaivola, L., Swamidass, S.J., Saigo, H., Baldi, B.: Graph kernels for Chemical Informatics, Neural networks, Special issue on neural networks and kernel methods for structured domains, 18(8), p. 1093-1110 (2005)
[15] Froehlich, H., Wegner, J.K., Sieker, F., Zell, A.: optimum assignment kernels for Attributed Molecular Graphs, Proceedings of the 22nd International Conference on Machine Learning, New York, NY, USA, p. 225-232. (2005)
[16] Jacob, L., Vert, J.-P.: Protein–ligand interaction prediction: an improved chemogenomics approach, Bioinformatics, Vol. 24 no. 19, p. 2149–2156 (2008)
[17] Lipkus, A. H.:A proof of the triangle inequality for the Tanimoto distance, Journal of Mathemathical Chemistry 26, p.263-265 (1999)
[18] Gärtner, T.: A Survey of Kernels for Structured Data, SIGKDD Explorations pp.49 - 58 (2003)
[19] Mohr, J., Jain, B., Sutter, A., Laak, A.T., Steger-Hartmann, T., Heinrich, N., Obermayer, K.:A Maximum Common Subgraph Kernel Method for Predicting the Chromosome Aberration Test, Journal of Chemical Information Modelling no. 50, p. 1821 - 1838 (2010)
[20] Vishwanathan, S.V.N., Borgwardt, K. M., Kondor, I.R., Schraudolph, N.N.:Graph Kernels, Journal of Machine Learning Research 9, p.1-41 (2010)

[21] Vert, J.-P.: The Optimal Assignmant Kernel is not positive definite, Technical Report HAL-00218278, (2008)

[22] Hinselmann, G., Fechner, N., Jahn, A., Eckert, M., Zell, A.: Graph kernels for chemical compounds using topological and three-dimensional local atom pair environments, Neurocomputing 74, p. 219 - 229 (2010)

[23] Aci, S., Bisson, G., Roy, S., Wieczorek, S.: Clustering of Molecules: Influence of the Similarity Measures, in: Selected Analysis in Data Analysis and Classification, p. 433 - 445, Springer (2007)

[24] Haasdonk, B.: Feature Space Interpretation of SVMs with Indefinite Kernels, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, no. 4(2005)