Image Classification Using Marginalized Kernels for Graphs

Emanuel Aldea[1], Jamal Atif[2], and Isabelle Bloch[1]

[1] ENST (GET - Telecom) Paris, Dept. TSI, CNRS UMR 5141 LTCI
46 rue Barrault, 75634 Paris Cedex 13, France
{aldea,bloch}@enst.fr,
[2] Groupe de Recherche sur les Energies Renouvelables (GRER)
Université des Antilles et de la Guyane, Campus de St Denis, 97 300 Cayenne, France
jamal.atif@guyane.univ-ag.fr *

Abstract. We propose in this article an image classification technique based on kernel methods and graphs. Our work explores the possibility of applying marginalized kernels to image processing. In machine learning, performant algorithms have been developed for data organized as real valued arrays; these algorithms are used for various purposes like classification or regression. However, they are inappropriate for direct use on complex data sets. Our work consists of two distinct parts. In the first one we model the images by graphs to be able to represent their structural properties and inherent attributes. In the second one, we use kernel functions to project the graphs in a mathematical space that allows the use of performant classification algorithms. Experiments are performed on medical images acquired with various modalities and concerning different parts of the body.

1 Introduction

Most of the traditional machine learning techniques ultimately cope with basic numeric features given in the form of arrays [1]. Such input information is processed for various purposes, like classification or regression.

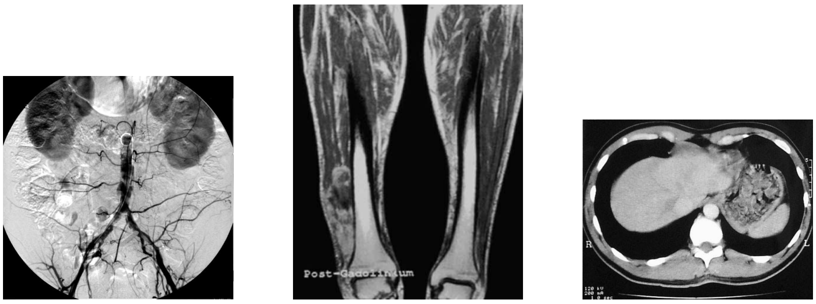
Nevertheless, it has become clear recently that machine learning should be able to cope equally with more complex input data, such as images, molecules, graphs or hypergraphs. The attributes that one can use to describe the input information are complex and very often inaccurate. In this context, classical learning methods do not provide a generic solution to the problem of processing complex input data.

Instead of changing the classical machine learning algorithms, our choice is to go in the opposite direction and to adapt the input for classification purposes so as to decrease structural complexity and at the same time preserve the attributes that allow assigning data to distinct classes.

As these complex structures started to emerge from various scientific areas (computer science, chemistry, biology, geography), one possible approach that we also employ in the current work has been to add a supplementary preprocessing step involving

* This work has been partially funded by GET and ANR grants during J. Atif's post-doctoral position at Telecom Paris.

structure and attribute extraction. In this way, we can return to the vectorial case by projecting a complex structure x belonging to a certain general space X into the n-dimensional real vector space $\mathbb{R}^n$ or in an infinite-dimensional Hilbert space. Different approaches have been used to project images in classifiable spaces. In one of them, images are treated as indivisible objects [2] and only global attributes are extracted. Another strategy has been to treat images as 〝bags〞 that contain indivisible objects [3], and interpret them by indexing these objects and their attributes, but ignoring whatsoever the relationships among them. However, the novel strategy that best defines our approach is to interpret images as organized sets of objects [4–6] and extract at the same time object attributes and structural information.



(a) Angiogram (b) MRI (c) CT

Fig. 1. Examples of medical images

We intend to use our work for the classification of 2D gray level medical images acquired using different techniques and concerning different parts of the human body. More precisely, the test base includes angiograms (Fig. 1(a)), sonograms, MRIs - magnetic resonance images (Fig. 1(b)), X-rays, CTs - computed tomograms (Fig. 1(c)), acquired with different imaging systems and following different protocols (even in each class). Most images also present small annotations, intended for the human reviewers. A good classification technique for this family of images should be able to cope with the generality factor due to the variety of classes and of acquisition techniques and equally with that due to noise (annotions, arrows) which are placed by instruments or by technicians and which are supposed to facilitate the work of medical teams.

This article starts with a brief presentation of support vector machines in Section 2 and of kernel methods for graphs in Section 3, emphasizing the one that represents the starting point for our work. Afterwards we describe in Section 4 our graph model for images, based on a generic, non-supervised segmentation followed by attribute extraction on the resulting structure. In Section 5, we explain where the difficulty of working with these attributes resides, and we propose a classification method adapted for image-issued graphs. This constitutes indeed the main contribution of the paper. Preliminary results on medical images are discussed in Section 6.

2 SVM Classifiers and Kernel Machines

In its basic form, a Support Vector Machine (SVM) classifier uses two sets of discriminative examples for training; these examples belong to a vector space endowed with a dot product. The main advantage of this classifier is the fact that it minimizes the classification error while maximizing the distance from the training examples to the separating hyperplane. It also allows the definition of a soft margin to prevent the mislabeled examples from perturbing too much the classification. Although SVMs have been originally designed as linear classifiers, they have been extended to perform nonlinear discrimination [7] by using a ‶kernel trick″, that replaces the dot product needed in computation by a non-linear positive definite kernel function. As a consequence, the examples are projected into a Hilbert space of higher dimension, called the feature space, which allows the construction of a linear classifier that is not necessarily linear in the initial space.

An important observation is that the classifier only needs the value of the kernel function between the examples. An additional advantage of this approch is that it allows classifying elements issued from spaces which are not naturally endowed with inner products (such as graph, tree or string spaces), as long as we use a valid kernel function.

3 Marginalized Graph Kernels

In this section, we briefly describe the marginalized kernel for labeled graphs.

We perform feature extraction on an undirected graph G, whose set of vertices is $\mathbb{X}$. The graph is labeled using the functions $v : \mathbb{X} \to \mathbb{S}_v$ for its vertices and $e : \mathbb{X} \times \mathbb{X} \to \mathbb{S}_e$ for its edges, $\mathbb{S}_v$ and $\mathbb{S}_e$ being two label sets. For the sake of clarity, we note $v(x)$ by v_x and $e(x_1, x_2)$ by $e_{x_1 x_2}$.

Feature extraction is carried out by first creating a set of random walks [8, 9]. The first element of the walk is a vertex x_1 given by a certain probability distribution over $\mathbb{X}$. At a subsequent moment during the generation, the walk will end at the current vertex x_i with a fixed (small) probability or it will continue by visiting a neighboring vertex x_{i+i}.

For each walk $h = (x_1, x_2, \ldots, x_n)$, labeled as $l_h = (v_{x_1}, e_{x_1 x_2}, v_{x_2}, \ldots, v_{x_n})$, the probability to obtain it may be expressed as:

$$p(h|G) = p_s(x_1) \prod_{i=2}^{n} p_t(x_i|x_{i-1}) \tag{1}$$

in which p_s and p_t have to be chosen in order to build $p(h|G)$ as a probability distribution in the random walk space $\mathbb{X}^* = \cup_{i=1}^{\infty} \mathbb{X}^i$, the union of all random walk spaces of a certain finite length i . One proposal for p_s and p_t, that we have also adopted for our model, is given for example in [9].

The kernel between two graphs G and G' measures the similarity of all the possible random walk labels, weighted by their probabilities of apparition:

$$K(G, G') = \sum_{h} \sum_{h'} k(h, h') p(h|G) p(h'|G') \tag{2}$$

As for the kernel between two random walk labels, a natural option is to define it as 0 if the walks have different lengths, and the product of all the kernels for their corresponding constituent parts otherwise:

$$k(h, h') = k^v(v_{x_1}, v_{x'_1}) \prod_{i=2}^{n} k^e(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i}) k^v(v_{x_i}, v_{x'_i}) \tag{3}$$

where k^v and k^e denote the kernel functions used for computing vertex and edge similarity, respectively. In computational chemistry, where this kernel has been extensively and successfully used, label functions have a limited range and therefore an appropriate kernel for assessing vertex or edge label similarity is the Dirac kernel:

$$k_\delta(z, t) = \begin{cases} 1 & , \ if\ z = t \\ 0 & , \ otherwise \end{cases} \tag{4}$$

Even so, computing the marginalized kernel for two graphs is difficult in the absence of two supplementary variables [10]. The first one $\Pi_s = ((\pi_s(x, x'))_{(x,x') \in \mathbb{X}^2}$ is a $|\mathbb{X}||\mathbb{X}'|$ vector containing the joint start probabilities of two vertices $x \in \mathbb{X}$ and $x' \in \mathbb{X}'$ if they have the same label, and 0 otherwise. The second variable needed for the kernel computation is $\Pi_t = ((\pi_t((x_1, x'_1)|(x_2, x'_2))))_{(x_1,x'_1),(x_2,x'_2) \in \mathbb{X}^2}$ is a $|\mathbb{X}||\mathbb{X}'| \times |\mathbb{X}||\mathbb{X}'|$ square matrix whose elements assess the joint transition probability between two pairs of vertices belonging to the first and to the second graph, if and only if these vertex pairs and the corresponding edge pair are identically labeled (otherwise the probability is null):

$$\begin{cases} \pi_s(x, x') = p_s^G(x) p_s^{G'}(x') \\ \pi_t((x_1, x'_1)|(x_2, x'_2)) = p_t^G(x_1|x_2) p_t^{G'}(x'_1|x'_2) \end{cases} \tag{5}$$

Using these new variables and $\mathbb{1}$ - the vector with all its values equal to 1, the kernel can be evaluated as:

$$K(G, G') = \Pi_s^T (I - \Pi_t)^{-1} \mathbb{1} \tag{6}$$

Due to the inversion of Π_t which dominates the computation cost, the problem has an order of complexity of $O((|X||X'|)^3)$. However, one may take advantage of the sparsity of Π_t, as well as of other methods [10, 11], in order to boost the performance of the algorithm. Many of these improvements are conditioned by a small range of labels and a low degree of vertex connectivity.

4 Graph Models of Images

The first step of our method consists in extracting and modeling image information. In order to achieve this, we use a labeled graph support (vertices are labeled as well as edges). The graph is obtained by first segmenting the initial image into regions, which allow us to describe its structure and to facilitate the information extraction step. Distinct regions correspond to vertices, while edges model the spatial relationships between regions. Beside this information brought by the structural expressivity of the graph, we integrate in the labeling relevant intrinsic information that we describe in detail later.

Therefore, the interest of using a graph structure goes beyond the structural expressivity and is due to the possibility that it offers to save various data and link them to particular components.

Unsupervised segmentation, as the low-level processing stage of our classification system, is an important and at the same time difficult task. Good results of a classifier with no prior information on the elements to be classified imply the use of a segmentation method that works reasonably well for any input image type.

For our processing stage we adopt a generic hierarchical image segmentation paradigm [12–14]. We suppose that the image is divided into components that may be further divided into subcomponents. This decomposition may be represented by a tree whose root node is the whole image and whose leaf nodes represent a partition of tiny regions built at the beginning of the processing step. This partition may be for example the set of pixels of the image. The advantage of employing a hierarchical segmentation method is that changes are gradual, unlike for other methods where the variation of one parameter may induce a completely different segmentation map. This aspect is relevant because medical images which have been acquired using different protocols but show the same body parts are sensitive to segmentation methods that use absolute thresholds. As opposed to that, hierarchical segmentation gives emphasis to relative relationships between image subconstituents.

To generate the leaf node partition of the tree, instead of employing each pixel as a terminal node in a tree, we use a watershed over-segmentation that leaves us however with a very large number of small regions. At this point we start climbing in the tree structure by merging neighboring regions that have the closest average gray levels:

$$dif_g(r_1, r_2) = \left| av_g(r_1) - av_g(r_2) \right| \tag{7}$$

where $av_g(r)$ denotes the average gray level in the region r.

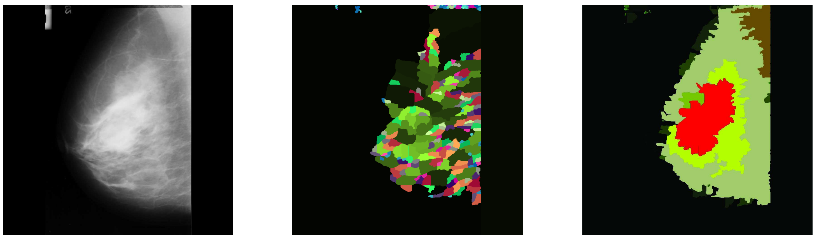
Concerning the stopping condition for the fusion process, we have chosen to set a dynamic threshold t_f. If the smallest gray level difference between two neighboring regions is higher than the threshold, we decide that the regions are not similar enough for the fusion to be performed and we stop. The threshold is dynamic because we compute it at each step as a (fixed) fraction f of the difference between the highest and the lowest region gray levels that exist in the image:

$$t_f = f \cdot (\max_{r \in Im} av_g(r) - \min_{r \in Im} av_g(r)) \tag{8}$$

As an example, we present a typical mammography in Fig. 2(a), along with one of the best possible human-assisted watershed based segmentations (Fig. 2(b)) and the result of the unsupervised method presented above (Fig. 2(c)).

Once the fusion has ended, we compute the following attributes for the resulting regions, encoded as vertices:

- region surface in pixels,
- relative surface, a real value that represents the percent of the image covered by the concerned region,
- average gray value of the region,



(a) Initial image

(b) Direct segmentation

(c) Region fusion

Fig. 2. Mammography segmentation

- relative average gray value, corresponding to an affine transform with respect to the highest and lowest average gray values in the image, g_{min} et g_{max}:

$$gray_{rel}(r) = \frac{g - g_{min}}{g_{max} - g_{min}}, \tag{9}$$

- region perimeter,
- region compacity, in $[0, 1/(4\pi)]$ and defined as the ratio between its surface and its squared perimeter,
- number of neighboring regions.

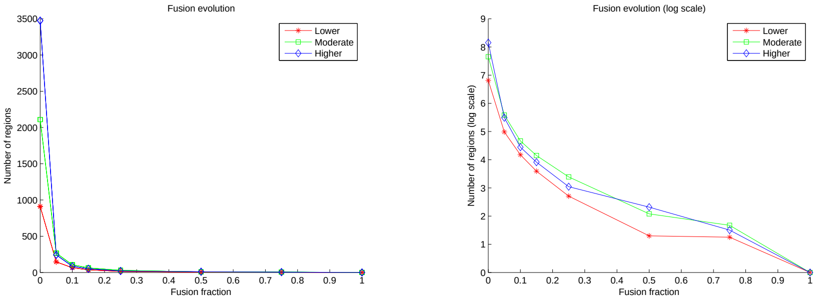
For the time being, the only relationship encoded by the edges (implicitely) is the neighborhood.

In Fig. 3, we present how the region number evolves when we modify the fusion threshold. Regions in images with a stronger initial over-segmentation tend to merge faster, so that for a fraction $f > 0.1$, results will start to be similar enough to those of images that presented a medium and low over-segmentation due to a smaller size or to a lower contrast, for example. This is an interesting result of our approach.

5 A Kernel for Image-Based Graphs

For image-based graphs, we propose a marginalized kernel different of that used in computational chemistry, which is able to better cope with specific image attributes.

A major structural difference in image-based graphs concerns the connectivity. While it is uncommon that atoms present more than four links towards the rest of the structure they belong to, this changes dramatically in the case of image regions, where there is no limit for the number of neighbors a region might possess. Potential optimizations based on graph sparsity become useless and, at the same time, the region neighborhood relationship has a lower importance than it has in a chemical compound, where the number of vertex neighbors $n(x)$ could be a used by a Dirac type kernel, as in Eq. (4). In our case however, the function $n(x)$ is unreliable as it is heavily influenced by the segmentation step, and cannot be helpful in building a vertex kernel or a significant part of it.



(a) Linear scale

(b) Log scale

Fig. 3. Threshold fusion fraction influence on fusion results in terms of number of regions

Another major difference in image-based graphs concerns the labeling. In the initial approach, vertices and edges are labeled using a small set of chemical symbols and possible bindings, and much information is given by the existence of the edge. The fundamental modification in the case of image-based graphs is that the labeling variable space becomes continuous and multi-dimensional, and a significant part of the information migrates from the graph structure to the labeling of its constituent parts.

The marginalized kernel presented in Section 3 employs a Dirac kernel for vertices and edges, which is useful for assessing structural similarities but is not adapted for a graph whose labeling is a major source of information. Under these circumstances, we have tried to adapt the vertex and edge kernels in order to define a proper similarity estimate for them.

The original graph kernel $K(G, G')$ defined in Eq. (2) is estimated by summing the similarities of all pairs of random walks of equal length. For a certain pair of such random walks (h, h'), let us suppose that we get simultaneously to a pair of corresponding vertices (x_i, x'_i). At this point we analyze the next transition in each walk; if the labels of the next two edges $(e_{x_i x_{i+1}}, e_{x'_i x'_{i+1}})$ and the labels of the next two vertices $(v_{x_{i+1}}, v_{x'_{i+1}})$ are not identical, the similarity brought by these walks will be null and we start analyzing another pair of walks. Otherwise we multiply the current similarity of the walks by the probabilities for the two transitions occurring in each walk. This leaves us with a probability of getting these random walks from start to end of:

$$p(h, h') = \left(p_s^G(v) \prod_{i=2}^{n} p_t^G(v_i | v_{i-1}) \right) \cdot \left(p_s^{G'}(v'_1) \prod_{i=2}^{n} p_t^{G'}(v'_i | v'_{i-1}) \right) \tag{10}$$

in which we suppose implicitely that for the walks (h, h'), the labels of all the constituent parts are identical. Using a Dirac similarity function for vertices and edges, it is obvious that random walk kernels in Eq. (3) will be also Dirac functions, so the graph kernel in Eq. (2) is reduced to the direct sum of all the probabilities $p(h, h')$ as in Eq. (10) computed for identically labeled random walks.

This strategy works for discrete ranged kernel functions, but in the case of region attributes like gray level or surface, we need a less discriminative kernel. Possible solutions to this problem are the Gaussian radial basis function (RBF) kernel and the triangular kernel [15]:

$$K^{RBF}(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$
$$K^{\Delta}(x,y) = \begin{cases} \frac{C-\|x-y\|}{C} \text{ , } if \ \|x-y\| \leq C, \\ 0 \text{ , } otherwise \end{cases} \tag{11}$$

The advantage of the first kernel over the second is that it offers a smoother, Gaussian discrimination compared to the uniform discrimination of the triangular kernel. However, beside an increase in computation time, the disadvantage of K^{RBF} is that it does not vanish at finite bounds, while the triangular kernel has a compact support.

We are entitled to use any of these kernels in the place of the Dirac kernel because they are also known to be positive definite and their use inside the graph kernel respects the closure properties of the family of kernel functions.

The next step is to integrate these values in the graph kernel computation. If we employ in Eq. (2) the joint probability from Eq. (10) and we replace the generic value $k(h,h')$ with that of Eq. (3), we get:

$$\begin{aligned} K(G,G') = \sum_{h}\sum_{h'} \Bigg[& k^v(v_{x_1}, v_{x'_1}) \prod_{i=2}^{n} \left(k^e(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i}) \cdot k^v(v_{x_i}, v_{x'_i}) \right) \\ & \times p_s^G(x_1) \cdot p_s^{G'}(x'_1) \cdot \prod_{i=2}^{n} \left(p_t^G(x_i|x_{i-1}) p_t^{G'}(x'_i|x'_{i-1}) \right) \Bigg] \end{aligned} \tag{12}$$

By comparing the kernel equation Eq. (2) with its revised form Eq. (12), we can notice the adaptation of the variables from Eq. (5) that we must perform in order to use the same method for the computation of the new kernel function:

$$\begin{cases} \pi_s(x,x') = p_s^G(x) p_s^{G'}(x') \cdot k^v(v_{x_1}, v_{x'_1}) \\ \pi_t((x_1,x'_1)|(x_2,x'_2)) = p_t^G(x_1|x_2) p_t^{G'}(x'_1|x'_2) \cdot k^e(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i}) k^v(v_{x_i}, v_{x'_i}) \end{cases} \tag{13}$$

The vertex and edge kernel functions appear in this model as probability multipliers along transitions, which penalize paths with respect to their constituent dissimilarities. Using the revised variables π_s and π_t from Eq. (13), we can now employ Eq. (6) to compute the revised graph kernel from Eq. (12).

In the general case of an attribute set $A = \{a_1, \ldots, a_n\}$ associated to a graph component, the kernel function will be extended in order to take into account all the elements of A. Kernel functions related to these various attributes allow us to treat them in a unified way, merging them in a unified similarity estimate [16]. As each kernel provides us with a partial description of data properties, we are interested in building a parameterized combination that employs each attribute according to its relevance. In our work, we have employed a linear combination of base kernels:

$$K_A = \sum_{i=1}^{|A|} \lambda_i K_{a_i} \tag{14}$$

where the multipliers $\lambda_i \geq 0$ satisfy $\sum_{i=1}^{n} \lambda_i = 1$. This time too, the weighted sum of definite positive functions preserves the key property of definite positiviness of the result.

6 Experimental Results

Based on this adapted marginalized kernel, we have conducted some preliminary experiments, whose purpose is to assess its viability and the impact of different graph attributes on its performance. As training examples, we have used ten head X-rays (coronal view) for the first class and ten mammographies (sagittal view) for the second one. For the moment, edges are not, beside their implicit structural importance, taken into account; therefore, we consider them as having the same label and we concentrate on the richer vertex attributes. We have particularly analyzed two of them which are adjusting to global image content: the relative surface s_{rel} with respect to the image surface and the relative average gray value $gray_{rel}$ defined in Eq. (9). They are less prone to perturbations, rescaling, contrast or brightness variations, etc.

In a first phase of our experiment, we have compared the performances of K^{RBF} and K^{Δ} in Eq. (11) for the s_{rel} attribute and for different parameterizations of C and respectively σ, on a testing sample of 42 images. For obvious statistics reasons, results for the two kernels are directly comparable in the situations where the value of C is at the 3-sigma level: $C = 3\sigma$.

Recognition rate	$C = 0.05$ $\sigma = 0.0167$	$C = 0.15$ $\sigma = 0.0500$	$C = 0.2$ $\sigma = 0.0667$	$C = 0.5$ $\sigma = 0.1667$	$C = 0.6$ $\sigma = 0.2000$	$C = 1$ $\sigma = 0.3333$
K^{Δ}	0.81	0.74	0.83	0.83	0.83	0.86
K^{RBF}	0.93	0.95	0.93	0.86	0.86	0.86

Table 1. Recognition rates based on the relative surface attribute s_{rel}

Results in Table 1 show that the RBF kernel performs well in the case of a strong discrimination (i.e. if region areas differ by more than one tenth of the image surface, the kernel returns a very small similarity value). While simplifying the discrimination function, the triangular kernel does not manage to discriminate as efficiently as the RBF kernel in the initial range of the surface attribute.

In a second step, we have built a linear kernel as in Eq. (14) based on both s_{rel} and $gray_{rel}$, in order to analyze the classification performance as a function of the individual kernel multipliers. The tests are performed, as before, on the sample of 42 images. The discrimination thresholds are fixed at 0.2 and 0.5 for the surface and gray level attributes respectively. Gray level weight is gradually increased from 0 to 1 in the unified kernel equation.

The graph shown in Fig. 4 proves that performance may be improved drastically by combining multiple attributes in the global kernel function. Even for the limited use of two vertex attributes in the absence of edge labeling, preliminary results are

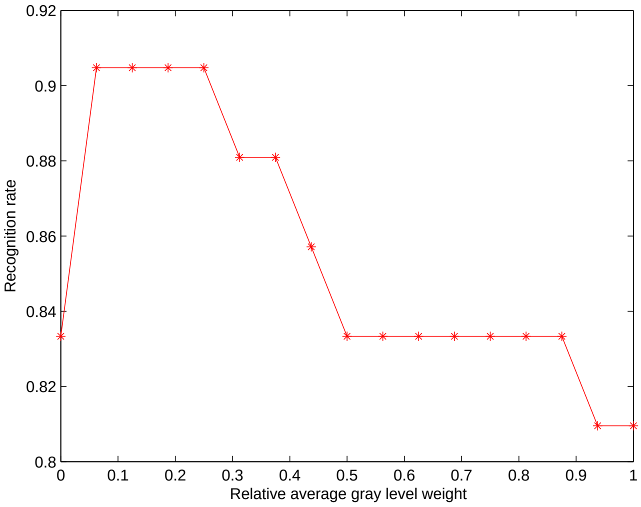


Fig. 4. Performance of a linear combination between a triangular kernel for relative surface (C=0.2) and a triangular kernel for relative average gray level (C=0.5)

encouraging. The weighted combination of kernels should be able to use information from multiple data sources by assessing the relative importance of each of them.

Triangular kernels prove to be noticeably faster than Gaussian ones and we hope that further weight optimization [16, 17] will help us increase the performance of a linear kernel based on triangular subcomponents.

7 Conclusions

We have presented a new version of marginalized graph kernel which extends the one being used in computational chemistry and which allows the processing of image-based graphs. This new approach incorporates in the similarity computation specific properties of image-based graphs, such as image attributes, irrelevance of the numbers of neighbors of a segmented region, etc. We have applied this approach to medical image classification, based on a generic segmentation method. Preliminary results validate this model and further work will be needed in investigating which of the possible attributes are relevant for graph-based image representation and classification. We are also interested in labeling edges with relationship attributes which go beyond planar neighborhood and which are essential for expressing globally image content. In the same direction, we could try to use some results concerning the kernel integration theory in order to find the most suitable multipliers for a certain attribute set that we consider relevant.

References

1. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: Proc. 23rd Int. Conf. on Machine Learning. (2006)
2. Chapelle, O., Haffner, P., Vapnik, V.: Svms for histogram-based image classification. In: IEEE Transactions on Neural Networks, special issue on Support Vectors. (1999)
3. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering object categories in image collections. In: Proc. IEEE Int. Conf. on Computer Vision (ICCV). (2005)
4. Ros, J., Laurent, C., Jolion, J.M., Simand, I.: Comparing string representations and distances in a natural images classification task. In: GbR'05, 5th IAPR-TC-15 workshop on graph-based representations. (2005) 72–81
5. Neuhaus, M., Bunke, H.: Edit distance based kernel functions for attributed graph matching. In: GbRPR. (2005) 352–361
6. Neuhaus, M., Bunke, H.: A random walk kernel derived from graph edit distance. In: SSPR/SPR. (2006) 191–199
7. Vapnik, V.: Statistical Learning Theory. Wiley-Interscience (1998)
8. Gaertner, T., Flach, P., Wrobel, S.: On graph kernels: Hardness results and efficient alternatives. In: Proc. 16th Annual Conf. on Computational Learning Theory. (2003) 129–143
9. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Proc. 20st Int. Conf. on Machine Learning. (2003) 321–328
10. Mahé, P., Ueda, N., Akutsu, T., Perret, J.L., Vert, J.P.: Extensions of marginalized graph kernels. In: ICML '04: Proc. 21st Int. Conf. on Machine Learning. (2004)
11. Borgwardt, K., Vishwanathan, S., Schraudolph, N., Kriegel, H.P.: Protein function prediction via faster graph kernels. In: NIPS Bioinformatics Workshop. (2005)
12. Haris, K., Estradiadis, S.N., Maglaveras, N., Katsaggelos, A.K.: Hybrid image segmentation using watersheds and fast region merging. IEEE Transactions on Image Processing **7**(12) (1998) 1684–1699
13. Beaulieu, J.M., Goldberg, M.: Hierarchy in picture segmentation: A stepwise optimization approach. IEEE Trans. Pattern Anal. Mach. Intell. **11**(2) (1989) 150–163
14. Brun, L., Mokhtari, M., Meyer, F.: Hierarchical watersheds within the combinatorial pyramid framework. In: DGCI, Springer (2005) 34–44
15. Mahé, P., Ralaivola, L., Stoven, V., Vert, J.P.: The pharmacophore kernel for virtual screening with support vector machines. J. Chem. Inf. Model. **46**(5) (2006) 2003–2014
16. Schlkopf, B., Tsuda, K., Vert, J.P.: Kernel Methods in Computational Biology. The MIT Press, Cambridge, Massachussetts (2004)
17. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)