ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Study on the loss of information caused by the "positivation" of graph kernels for 3D shapes

Gaëlle Loosli[1]

1- Clermont Université - Université Blaise Pascal
CNRS, UMR 6158, LIMOS
Aubière, France

Abstract. In the presented experimental study, we compare the classification power of two variations of the same graph kernel. One variation is designed to produce semi-definite positive kernel matrices ($K_{matching}$) and is an approximation of the other one, which is indefinite (K_{max}). We show that using adaptated tools to deal with indefiniteness (KSVM), the original indefinite kernel outperforms its positive definite approximate version. We also propose a slight improvement of the KSVM method, which produces non sparse solutions, by adding a fast post-processing step that gives a sparser solution.

1 Introduction

Until recent developments in the usage of indefinite kernels [1, 2, 3], it was mandatory to produce positive definite kernels in order to use SVM. To do so, in particular in graph kernels, it is quite usual to add treatment steps that garantee the definite-positiveness of the produced kernel. In this paper we question the impact of such steps on the classification ability of the kernels, through an empirical study on 3D shapes classification from SHREC'09 database [4]. In this study, we compare the classification results using two versions of the bag of paths kernel. On the one hand, we implement the K_{max} kernel [5] which happens to be indefinite in general but which corresponds to the intention of the authors in the design of the kernel. On the other hand, we use its approximate positive definite version named $K_{matching}$, that conveniently avoids the usage of the *max* operator, responsible for the indefiniteness of K_{max}. For the SVM solver, we use KSVM [1], (for SVM in Kreĭn spaces). Indeed, to make a fair comparison we need a solver that provides a solution in the original indefinite space of K_{max}. In [1], the interest of KSVM is shown compared to methods that transform or project indefinite kernels. In this study we address a slightly different question: the adaptation of the kernel to the mathematical requirements of standard solvers is included in the kernel design. We intend to exhibit the fact that the indefinite part of the kernel contains pertinent information. In section 1.1 we provide the definition of the kernels we use. In section 1.2 we describe KSVM and in section 2 we propose a slight improvement on the sparcity of the solution. In section 3 we present our experimental study, that covers the validation of the KSVM post processing step (section 3.1) and a comparison of many kernels computed from SHREC dataset and the classification performance comparison between K_{max} and $K_{matching}$ (sections 3.1.1 and 3.1.2). Section 4 concludes this study.

1.1 Graph kernels

The two graph kernels we use in this study are taken from [5]. The comparison between two graphs is based on the comparisons between paths that are extracted from each graph. A first kernel is defined to compare two paths h and h':

$$K_L(h,h') = K_v\big(l(v_1), l(v_1^{'})\big)\prod_{i=2}^{n} K_e\big(l(v_{i-1},v_i), l(v_{i-1}^{'},v_i^{'})\big)K_v\big(l(v_i), l(v_i^{'})\big) \tag{1}$$

where v_i are the vertices of a path h of length n, (v_{i-1}, v_i) denotes an edge, K_v is a kernel fonction on the vertex labels (obtained through function ℓ) and K_e is a kernel function on the edges labels. Let's now suppose that sets of paths P_1 and P_2 have been extracted from graphs G_1 and G_2. The graph kernels we use are defined as follows:

$$K_{max}(G_1,G_2) = \frac{1}{2}\Big(\frac{1}{|P_1|}\sum_{i:h_i\in P_1}\max_{j:h_j\in P_2} K_L(h_i,h_j) + \frac{1}{|P_2|}\sum_{i:h_i\in P_2}\max_{j:h_j\in P_1} K_L(h_i,h_j)\Big) \tag{2}$$

This kernel being non positive definite due to the *max* operator, it is approximated by a Gaussian kernel $K_{d_L}(h_i,h_j) = exp(\frac{K_L(h_i,h_j)^2}{-2\sigma^2})$ which gives the matching kernel:

$$K_{matching}(G_1,G_2) = \frac{1}{|P_1|}\frac{1}{|P_2|}\sum_{i:h_i\in P_1}\sum_{j:h_j\in P_2} K_{d_L}(h_i,h_j) \tag{3}$$

In both cases, the idea is to retrieve the contribution of the most matching pairs of paths: only the most matching in the first case, and mostly the most matching in the second.

1.2 KSVM, SVM in Kreĭn spaces

We give results from [1], based on [6], where SVM in Kreĭn spaces (more precisely in RKKS - Reproducing Kernel Kreĭn Space - that are indefinite inner product spaces endowed with a Hilbertian topology, in which the positiveness axiom is not required) are defined as eq.4. Let x_i with $i \in [1..\ell]$ be the training examples and y_i their labels. The primal problem is:

$$\left\{\begin{array}{ll} \underset{f\in\mathcal{K}, b\in I\!R}{\text{stab}} & \frac{1}{2}\,\langle f,f\rangle_{\mathcal{K}} \\ \text{s.t.} & \displaystyle\sum_{i=1}^{\ell}\max\big(0, 1-y_i(f(x_i)+b)\big) \le \tau\ . \end{array}\right. \tag{4}$$

where stab means *stabilize*. The stabilization problem arizes as the solution of quadratic problems in Kreĭn spaces. It means that in general the solution is a saddle point and not an extremum, which means that standard optimization techniques can't be used as is. The representer theorem in Kreĭn space gives:

$$\left\{\begin{array}{lll} & f(.) = \displaystyle\sum_{i=1}^{\ell}\alpha_i y_i k(x_i,.) & \\ \text{with} & -C \le \alpha_i \le C & \forall i \in [1..\ell] \end{array}\right. \tag{5}$$

and the dual formualtion is given in eq.6. The stabilization setting is unusual but the authors show that solving system 4 leads to algorithm 1, where G is the kernel matrix such that $G(i,j) = y_i y_j k(x_i, x_j)$.

$$\left\{\begin{array}{ll} \underset{\alpha}{\text{stab}} & -\frac{1}{2}\alpha^\top G\alpha + \alpha^\top 1 \\ \text{with} & \alpha^\top y = 0 \\ \text{and} & -C \le \alpha_i \le C \quad \forall i \in [1..\ell] \end{array}\right. \tag{6}$$

Algorithm 1 KSVM

```
Require: y, C and G
  [U,D] = EigenDecomposition(G)
  $\tilde{G} = USDU^\top$ with S=sign(D)
  [$\tilde{\alpha}$,b] = SvmSolver(y,$\tilde{G}$,C)
  $\alpha = USU^\top\tilde{\alpha}$
  return $\alpha$,b
```

2 Sparse KSVM

In algorithm 1, the produced solution α is not sparse. Having a non sparse solution is not really desirable, this is the reason why we explore a simple trick to reduce the solution size. We start from the observation that with $G = UDU^\top$ and $\tilde{G} = UDSU^\top$, we have

$$\begin{array}{rl} \alpha = & USU^\top\tilde{\alpha} \\ & U^{\top^{-1}}D^{-1}DSU^\top\tilde{\alpha} \\ & (U^{\top^{-1}}D^{-1}U^{-1})(USDU^\top)\tilde{\alpha} \\ & G^{-1}\tilde{G}\tilde{\alpha} \\ G\alpha = & \tilde{G}\tilde{\alpha} \end{array} \tag{7}$$

This says that all training points have exactly the same evaluation value $f(x)$ in each representation (Kreĭn and Hilbert). Note here that the solution in the corresponding Hilbert space $\tilde{\alpha}$ is sparse as usual SVM. To obtain an approximate sparse α, a simple way is to force non support vectors in the Hilbert space to have a nul multiplier α in the original Kreĭn space by solving the following linear system:

$$G_{:,sv}\alpha_{sv} = \tilde{G}_{:,sv}\tilde{\alpha}_{sv} \tag{8}$$

where *sv* designates support vectors in the corresponding Hilbert space. The system is over-determined and we use a QR solver.

3 Empirical study

We first evaluate the interest of sparse KSVM. Then we describe our experiments on graph kernels. All experiments are done in Matlab using KSVM code provided by the authors, except for linear programming which is done using CPLEX. For experiments in a multiclass setting, the 1vsAll setting is applied.

3.1 Evaluation of sparse KSVM

For this set of experiments, we use synthetic datasets (checkers and apple-banana) to illustrate the impact of imposing sparseness. We use a tanh kernel without optimization on it, we just check it is really indefinite: the least eigenvalue is on average between -50 and -400 depending on the datasets. For each problem shape (checkers and apple-banana), we generate 6 training sets, of 3 sizes (300 - 1000 - 2000), and 2 levels of overlapping (none or very low, and high). Each configuration is generated and trained 10 times, and all test sets are of size 10000. We monitored the accuracy, training and testing time, and the solution size, for KSVM and sparse KSVM. For the sake of comparison, we also added LP-SVM since it can be regarded as an alternative method for indefinite kernels [2]. Figure 1 shows the average results over all the configurations. We observe that in terms of accuracy, all methods are similar, with a small loss of accuracy for sparse KSVM. Concerning the solution size (hence the testing time), the gain of sparse KSVM over KSVM is obvious. LP-SVM is better from that point of view, but the training time explodes. Overall, sparse KSVM seems to be a good compromise, at least on those experiments.

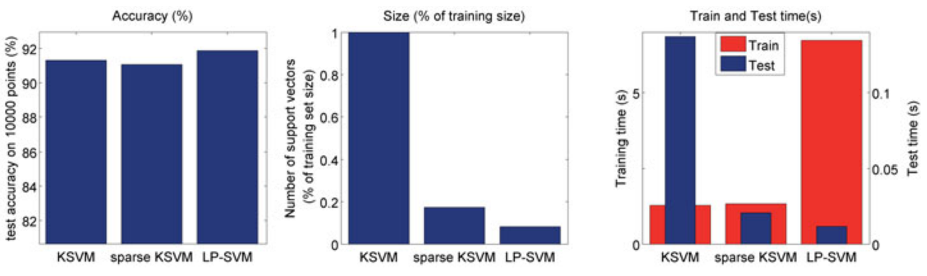
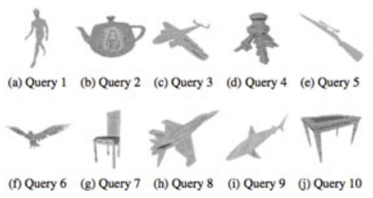


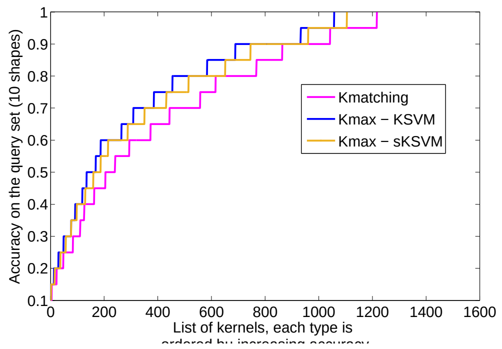
Fig. 1: We compare KSVM, sparse KSVM and LP-SVM on various sets of synthetics data, in terms of accuracy, solution size, training time and testing time. We observe that sparse KSVM is a good compromise.

3.1.1 SHREC'09

In this part we compare graph kernels extracted from SHREC 2009 [4] dataset. This dataset contains 200 shapes from 10 main classes (humans, teapots, planes, birds, chairs, etc.). Each main class has 2 subclasses that are structuraly similar (for instance chairs and tables). For the evaluation of class prediction, this is taken into account by the attribution of 1 point if the shape is classified in its *highly relevant* class and 0.5 in its *marginally relevant* class. The accuracy is computed for the query set represented on fig.2a.
From each shape we extract 8 different graphs following the method proposed in [7]. For each of the 8 sets of graphs, we compute 196 K_{max} kernels and 196 $K_{matching}$ kernels, which correspond to various sets of parameters (kernels on edges and vertices are Gaussians for which we vary σ from 10 to 10^7, and we also vary the maximum path length from 1 to 4.). In total we obtain 1568 indefinite kernels and as many SDP kernels. For each pair of kernels (K_{max} and $K_{matching}$ computed of the same set of graphs with the same hyper parameters), we train a KSVM and a SVM (with a cross-validation on the C parameter). Figure 2b



(a) The query set from SHREC'09



(b) Classification accuracy

Fig. 2: This figure summarizes the classification accuracy on the query set, for each computed kernel for K_{max} and $K_{matching}$. In the case of K_{max}, the training is done using KSVM and sparse KSVM (noted sKSVM). Each set of results is ordered by classification accuracy on the query set. K_{max} kernels outperform $K_{matching}$ kernels and we also observe that the sparse KSVM is as good as KSVM alone.

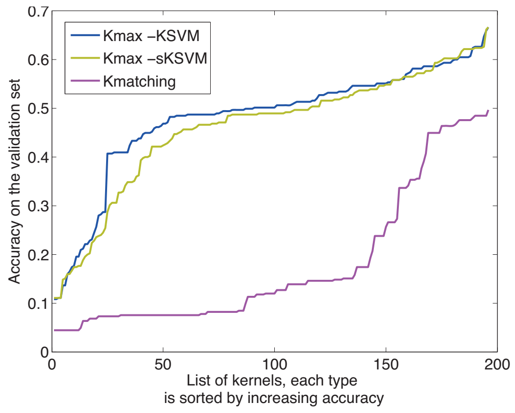
reports results in term of classification accuracy. On average K_{max} outperforms $K_{matching}$.

3.1.2 SHREC'08

From SHREC'08 [8], we extract only one set of graphs. We have 145 training shapes from 12 classes, which makes it a difficult dataset for multiclass learning. The validation set is made of 425 shapes, and we also have a query (test) set of 76 shapes. Each shape belongs to one class so the accuracy is computed in a classic way contrarely to the previous experiment. Figure 3a gives performances for the validation set. Contrarely to the previous dataset, the task is much more difficult, and we observe that $K_{matching}$ is by far outperformed by its indefinite version K_{max}.

4 Conclusion

In this paper we present an emprical study of the potential loss of information implied by kernel positivation strategies. While not being exhaustive since it is based on one type of graph kernel and two datasets, it outlines the fact that one should not take for granted that semi-definite positive kernels are what we need. Another message that can be assessed by this study is that is may be intersting for kernel designers to define their kernels relaxing the positive definite constraint now that adapted solvers are available.



(a) Classification accuracy on validation set

$K_{matching}$	K_{max}	K_{max}
SVM	KSVM	s-KSVM
52.63%	68.42%	65.79%

(b) Results on the query set

Fig. 3: Figure (a) summarizes the classification accuracy on the validation set, for each computed kernel for K_{max} and $K_{matching}$. In the case of K_{max}, the training is done using KSVM and sparse KSVM (noted sKSVM). Each set of results is ordered by classification accuracy on the validation set. K_{max} kernels clearly outperform $K_{matching}$ kernels and we also observe that the sparse KSVM is almost as good as KSVM alone. On table (b), we provide the test accuracy on the query set for the best (from validation) kernel of each type.

References

[1] Gaelle Loosli, Stephane Canu, and Cheng Soon Ong. Learning svm in krein spaces. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, preprint(preprint):preprint, 2015.

[2] Ibrahim M Alabdulmohsin and Xiangliang Zhang Xin Gao. Support vector machines with indefinite kernels. In *Asian Conference on Machine Learning, ACML*, 2014.

[3] Frank-Michael Schleif and Peter Tino. Indefinite proximity learning: A review. *Neural computation*, 2015.

[4] J Hartveldt, Michela Spagnuolo, Apostolos Axenopoulos, Silvia Biasotti, Petros Daras, Helin Dutagaci, Takahiko Furuya, Afzal Godil, Xiaolan Li, Athanasios Mademlis, et al. Shrec'09 track: Structural shape retrieval on watertight models. In *3DOR*, pages 77–83, 2009.

[5] Frédéric Suard, Alain Rakotomamonjy, and Abdelaziz Benrshrair. Kernel on bag of paths for measuring similarity of shapes. In *ESANN*, pages 355–360, 2007.

[6] Cheng Soon Ong, Xavier Mary, Stéphane Canu, and Alexander J. Smola. Learning with non-positive kernels. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 81, New York, NY, USA, 2004. ACM.

[7] Vincent Barra and Silvia Biasotti. 3d shape retrieval using kernels on extended reeb graphs. *Pattern Recognition*, 46(11):2985–2999, 2013.

[8] Daniela Giorgi and Simone Marini. Shape retrieval contest 2008: classification of watertight models. 2008.