Kernel approximations for W-operator learning

Igor S. Montagner, Nina S. T. Hirata, Roberto Hirata Jr.
University of São Paulo,
São Paulo, Brazil
igordsm,nina,hirata@ime.usp.br

Stéphane Canu
University of Rouen - INSA,
Rouen, France
stephane.canu@insa-rouen.fr

***Abstract*—Designing image operators is a hard task usually tackled by specialists in image processing. An alternative approach is to use machine learning to estimate local transformations, that characterize the image operators, from pairs of input-output images. The main challenge of this approach, called W-operator learning, is estimating operators over large windows without overfitting. Current techniques require the determination of a large number of parameters to maximize the performance of the trained operators. Support Vector Machines are known for their generalization performance and their ability to estimate nonlinear decision surfaces using kernels. However, training kernelized SVMs in the dual is not feasible when the training set is large. We estimate the local transformations employing kernel approximations to train SVMs, thus with no need to compute the full Gram matrix. We also select appropriate kernels to process binary and gray level inputs. Experiments show that operators trained using kernel approximation achieve comparable results with state-of-the-art methods in 4 public datasets.**

***Keywords*-Kernel approximation, W-operator learning, Machine learning, Image Processing**

I. Introduction

Image processing techniques are used to extract useful information from images for many different tasks, such as document analysis and medical imaging. However, techniques used in a domain may not be useful or relevant when processing images from other domains, since they exploit characteristics specific to their domain of application. An alternative approach is to use Machine Learning to estimate a transformation between images.

The most general way of expressing image processing tasks is by forming pairs of input-output images like the ones in Figures 1 and 2. Although this formulation completely defines a task, it does not contain information about how to represent the transformation of the input image into the output.

There has been much progress in estimating image operators from input-output pairs for W-operators [3], [4], [5]. These are translation invariant image transformations restricted to a finite neighborhood called *window*. They can also be seen as pixel classifiers that take as input the pixel values inside the window and output a pixel value.

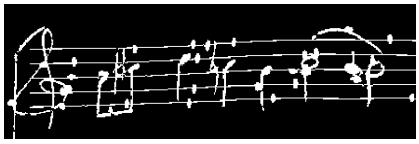
The main challenge in W-operator learning is to balance window size and generalization performance. Small windows lack the representative power to discriminate complex or large patterns, resulting in poor performance. On the other hand, large windows are able to discriminate complex patterns but also require much more training data to avoid overfitting.

In earlier works [6], [7], [5], [8] W-operators have been trained using Decision Trees or the ISI algorithm [9] (for the minimization of Boolean functions). Both algorithms do not perform well when using large numbers of features. Although more advanced techniques, such as two-level operators [5], were developed to mitigate this problem, they also require the determination of a much bigger set of parameters.

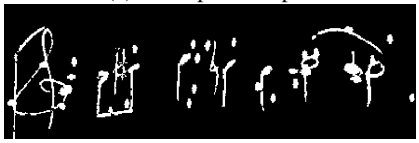
Support Vector Machines (SVM) are known to generalize well even in high-dimensional problems due to the maximization of the classification margin [10], [11]. Also, they can estimate nonlinear decision surfaces by applying the kernel trick, which makes them powerful tools for classification. Thus, SVMs and kernels are adequate tools to estimate W-operators. However, typical problems in W-operator learning involve hundreds of thousands, sometimes millions, of training examples. This makes the use of kernel methods difficult, since for a given training set with N samples it is usually necessary to compute the full $N \times N$ Gram matrix.

An approach that has been gaining momentum in the last years is kernel approximation [12], [13], [14]. Instead of computing the full Gram matrix for a kernel k, these works focus on computing a feature map φ such that $k(\mathbf{x}, \mathbf{x}') \approx \varphi(\mathbf{x})^T \varphi(\mathbf{x}')$ and on optimizing a linear classifier in the transformed space. These techniques have been successfully used in large scale learning scenarios [15], [16], where they have obtained results competitive with state-of-the-art Deep Neural Networks.

The objective of this work is to investigate the use of kernel methods in the estimation of W-operators by approximating



(a) Example of input



(b) Expected output

Fig. 1: Input-output pair for the staff removal task (*staffs* dataset [1]).

(a) Example of input (b) Expected output

Fig. 2: Input-output pair for the eye vessel segmentation task (*DRIVE* dataset [2]).

φ from data. The approximation is computed by the Nyström method [12] and can be used to process both binary and gray level inputs. We also chose adequate kernels for each case based on their visual properties.

The two most widely known methods of kernel approximation are the Nyström method and Random Fourrier features. The Nyström method [12] approximates φ from a subset of the data. Random Fourrier features [13] estimate φ by sampling from a random distribution. It is faster than Nyström, but it can only be used for shift-invariant kernels (i.e. $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$). A comparison between these methods reported in [14] indicates that data dependent approximations (such as Nyström) achieve better results. Since we also work with polynomial kernels, we opted for using the Nyström method.

We compare our work with earlier methods in W-operator learning [17], [8], [7], [6] and heuristic algorithms using 4 public datasets. Results show that our approach achieves performance comparable with both state-of-the art learned operators and specialized algorithms, beating them in some cases.

This paper is organized as follows. In Section II we review the Nyström method and explain how it can be used to estimate φ from data. In Section III we present the basic formulation of W-operator learning. In Section IV we describe our method and justify the choices of kernels for binary and gray level inputs. We present comparisons with other algorithms in Section V and discuss advantages and limitations of our method in Section V-C. Finally, we present our conclusions in Section VI.

II. Kernel approximation using the Nyström method

Williams and Seeger [12] propose a method to approximate the eigenvalues and eigenfunctions of a kernel given a sample of N points. Then they show how to use this result to build an approximation $\tilde{K}$ of the Gram matrix K of these points using only a subset of size m. Finally, we find an approximation of φ by a careful analysis of the $\tilde{K}$ matrix.

The authors call this kernel approximation technique the "Nyström method".

A. Approximating the eigenfunctions of a kernel

Let $k : \mathcal{X}, \mathcal{X} \to \mathbb{R}$ be a positive definite kernel. We know from Mercer's theorem[11] that k has an expansion in terms of eigenvalues/vectors

$$k(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{x}) \phi_j(\mathbf{x}'), \tag{1}$$

where $\lambda_j \in \mathbb{R}$ and ϕ_j is continuous in $\mathcal{X}$, such that

$$\int_{\mathcal{X}} k(\mathbf{x}', \mathbf{x}) \phi_j(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \lambda_j \phi_j(\mathbf{x}') \tag{2}$$

where $p(\mathbf{x})$ is the probability density of input pattern $\mathbf{x} \in \mathcal{X}$. This integral can be estimated from a sample $\{x_i \in \mathcal{X}\}, i = 1, \ldots, N$, taken according to $p(\mathbf{x})$, as

$$\frac{1}{N} \sum_{i=1}^{N} k(\mathbf{x}', x_i) \phi_j(x_i) \approx \lambda_j \phi_j(\mathbf{x}') \tag{3}$$

Let $K^{(N)} \in \mathbb{R}^{N \times N}$ be the Gram matrix of k w.r.t the examples $x_i, i = 1, \ldots, N$. Its eigendecomposition is

$$K^{(N)} U^{(N)} = U^{(N)} \Lambda^{(N)}, \tag{4}$$

where $U^{(N)}$ is orthonormal and $\Lambda = diag(\lambda_i^{(N)})$. By evaluating equation (3) in points x_j and comparing the result with equation (4) we obtain the following approximations:

$$\phi_j(x_i) \approx \sqrt{N} U_{i,j} \tag{5a}$$

$$\lambda_j \approx \frac{\lambda_j^{(N)}}{N} \tag{5b}$$

Combining equations (5) and (3) again, we obtain the following expression for $\phi_j(\mathbf{x}), \mathbf{x} \in \mathcal{X}, j = 1, \ldots, N$.

$$\phi_j(\mathbf{x}) \approx \frac{\sqrt{N}}{\lambda^{(N)}} \sum_{k=1}^{N} k(\mathbf{x}, x_k) U_{k,j} \tag{6}$$

Thus, we can approximate the eigenvalues λ_j and their corresponding eigenfunctions ϕ_j up to $j = N$ from a sample of $\mathcal{X}$ of size N.

B. Approximating the Gram matrix

The results above can be used to approximate the Gram matrix $K \in \mathbb{R}^{N \times N}$ of points $\{x_i\}$ using only a subset of m examples. For convenience, we select the first m examples. Let K be partitioned as below.

$$K = \begin{bmatrix} K_{(m,m)} & K_{(m,N-m)} \\ K_{(N-m,m)} & K_{(N-m,N-m)} \end{bmatrix} \tag{7}$$

Williams and Seeger [12] show that, by applying the Nyström method to approximate the eigenvalues and eigenvectors of K we obtain an approximated Gram matrix $\tilde{K}$ such that

$$\tilde{K} = K_{(N,m)} K_{(m,m)}^{-1} {K_{(N,m)}}^T \quad (8)$$

and, more specifically,

$$\tilde{K}_{(N-m,N-m)} = K_{(N-m,n)} K_{(m,m)}^{-1} K_{(m,N-m)}. \quad (9)$$

This implies that to compute a Gram matrix for the unused $N-m$ points we just need to evaluate the kernel between the new points and the selected m points. Thus, given the eigendecomposition $K_{(m,m)}^{-1} = U^{(m)} diag(\lambda^{(m)}) {U^{(m)}}^T$, we can define the feature map $\varphi : \mathcal{X} \to \mathbb{R}^m$ [14]

$$\varphi(\mathbf{x}) = diag(\frac{1}{\sqrt{\lambda_j}}) {U^{(m)}}^T K_{\mathbf{x}}, \quad (10)$$

with $K_{\mathbf{x}} = (k(\mathbf{x}, x_1), \ldots, k(\mathbf{x}, x_m))$. From Eq. (9), we verify that $k(\mathbf{x}, \mathbf{x}') \approx \varphi(\mathbf{x})^T \varphi(\mathbf{x}')$.

III. Fundamentals of W-operator learning

Let E be an image definition domain. Then, the set of all images defined on E with gray-scales in a set L is denoted L^E. If $L = \{0, 1\}$ then the images are binary. Given two sets L_1 and L_2, a mapping from L_1^E to L_2^E is called an image operator.

Image operators that are translation-invariant and locally defined with respect to a non-empty window W are called W-operators [18]. An important property of a W-operator Ψ is its local characterization by a function $\psi : L_1^W \to L_2$, i.e., given any image $f \in L_1^E$ and p in E,

$$[\Psi(f)](p) = \psi(f_{-p}|_W) \quad (11)$$

where $f_{-p}|_W$ is the subimage of f around p, or image pattern, constrained within W [19].

Thanks to this property, the problem of learning W-operators can be reduced to the problem of learning its local function [18], [5]. For each image pattern $\mathbf{x}$, the aim is to estimate $\psi(\mathbf{x})$. If image patterns are vectorized, they can be understood as elements in L_1^M where M is the window size ($M = |W|$). Therefore, the problem of image operator learning can be reduced to the problem of learning a classifier $\psi : L_1^M \to L_2$ and L_1 and L_2 can be understood, respectively, as the set of feature values and the set of class labels.

In order to learn ψ, pairs of input-output images, that exemplifies the desired processing (as the ones shown in Figures 1 and 2), are used to extract the training samples. A scheme that illustrates the process is shown in Figure 3. Patterns $\mathbf{x}$ in L_1^W are collected at each pixel p of the input images, together with the corresponding expected output value y (value of the output image at the same pixel). Then, patterns are vectorized and used to learn ψ. Note that a large number of training samples can be extracted from a single input-output pair of images.

The majority of previous works deal with only binary W-operators (i.e., $L_1 = L_2 = \{0, 1\}$) [18], [5], [17], due to computational and statistical difficulties related to the number of features and amount of training data. Two level operators [5] are the most recent method designed to cope with large windows. It proposes to train an ensemble of w-operators using stacked generalization [20]. More recent works have been concerned on selecting appropriate operators to compose the ensemble [17], [21], [8]. Details of a standard learning procedure for single or two-level operators can be found, for instance, in [5].

With regard to gray-scale image operators, the class of aperture operators has been proposed earlier to mitigate computational and statistical issues [22], [4]. Aperture filters assume locality with respect to the range of gray-levels in addition to locality with respect to the spatial domain. The size of the patterns is defined by W, and the intensities are constrained within an interval $[-a, a]$ (for a relatively small a).

When the output images are binary, the performance of W-operators can be expressed in terms of standard measures such as accuracy, recall, precision and F1-measure, computed on the pixels.

IV. Kernelized image operator learning

Our method focuses on estimating local functions of the form $\psi^{(k)}(\mathbf{x}) = sign(w^T \varphi(\mathbf{x}) + b)$ where $\varphi : \mathcal{X} \to \mathbb{R}^m$ is a feature map that approximates the kernel k.

Let $\{x_i, y_i\}, i = 1, \ldots, N$ be the set of patterns extracted from an image set as described in Section III and $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a kernel. Our method determines $\psi^{(k)}$ in three steps. First we estimate φ from a set of m examples randomly selected from $\{x_i\}$ according to Section II. Then let $Z \in \mathbb{R}^{N \times m}$ be a matrix such that $Z_{i\bullet} = \varphi(x_i)$. We determine w by minimizing the SVM loss, defined below, using the transformed features Z. The operation $[\cdot]_+$ denotes $max(0, \cdot)$.

$$\min_w \sum_{i=1}^{N} [1 - y_i \cdot (Zw + b)]_+ + \frac{1}{2} ||w||_2^2, \quad (12)$$

Since $k(x_i, x_j) \approx \varphi(x_i)^T \varphi(x_j)$, by solving the SVM using the transformed features Z we are approximating the solution of a SVM trained in the dual using kernel k and the original features. However, there is no need to compute the Gram matrix explicitly, since we can solve the SVM minimization problem in the primal.

This approach has several advantages in comparison with training standard kernelized SVMs. First of all, computing the Gram matrix may not be feasible. Typical problems in image operator learning have hundreds of thousands, sometimes millions, of examples. The classification of a pattern is also faster, since the decision surface has m components instead of N (and $m << N$).

The choice of an adequate kernel for each problem can result in significant differences in the performance of the trained image operator. Given the differences between binary and gray level patterns we selected different kernels for each case.

In the binary case we use the polynomial kernel, which is defined as below. Parameter d controls the degree of the polynomial and $c \geq 0$ is a non-negative constant.

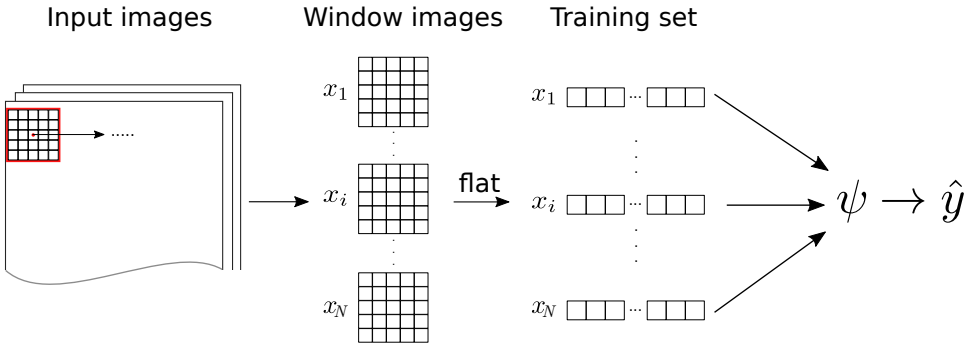


Fig. 3: Feature extraction process.

$$k_d(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d \tag{13}$$

Its feature map φ^d contains all interaction terms $\mathbf{x}_1^{e_1} \mathbf{x}_2^{e_2} \dots \mathbf{x}_M^{e_M}$ such that $\sum_{i=1}^{M} e_i = d$. An interaction term acts as a visual binary pattern recognition, since it evaluates to 1 iff all its variables are 1. The decision surface $w^T \varphi^d(\mathbf{x})$ is, thus, a linear combination of pattern detectors. It also bears resemblance with the disjunctive normal form of Boolean functions. Note that there is some redundancy in φ, since all interaction terms that contain the same variables are equal no matter the exponent of the individual variables.

Although φ^d can be computed directly, its dimension grows exponentially on the degree d and polynomially in M. To avoid this, we approximate φ^d using the Nyström method.

One of the challenges when dealing with gray level input images is that frequently the extracted patterns are visually similar but differ in most, if not all, variables. The same pattern can also appear with different brightness, making this even harder. A Gaussian kernel (defined below) may address these challenges.

$$k^\gamma(\mathbf{x}, \mathbf{x}') = e^{-\gamma ||\mathbf{x} - \mathbf{x}'||^2} \tag{14}$$

It can be seen as a soft threshold of the visual similarity between $\mathbf{x}$ and $\mathbf{x}'$. If both patterns are visually similar the norm of $\mathbf{x} - \mathbf{x}'$ will be small, since the small differences in the variables will maintain the norm relatively small. On the other hand, when patterns are visually different most, if not all, variables will have significant differences, making the norm comparatively much larger. By tuning the γ parameter we can control the tolerance of the similarity measure. A "big" γ emphasizes differences, while a small one allows more patterns to be considered similar.

An interesting visual property of the Gaussian kernel is that as long as the relative brightness between $\mathbf{x}$ and $\mathbf{x}'$ does not change much, the Gaussian kernel will output the similar values. Thus, this kernel is robust to global changes in brightness.

V. Experiments

We trained kernelized image operators on 4 publicly available datasets: *CharS* [5] (character segmentation), *TexRev* [5] (text segmentation), *staffs* [1] (music score staff removal) and *DRIVE* [2] (retina vessel segmentation). Both the input and output images in *CharS, TexRev* and *staffs* are binary. In *DRIVE* the inputs are gray level and the outputs are binary.

Dataset	Domain	Degree	m
CharS	9×7	3	2000
CharS	11×9	5	2000
TexRev	9×7	5	2000
TexRev	11×11	7	2000
Staffs	11×11	3	2000

TABLE I: Chosen parameters for the binary datasets *CharS, TexRev* and *Staffs*.

A. Binary datasets and the polynomial kernel

The *CharS* and *TexRev* datasets were first published in [5]. Their images were divided into three parts: T_1 and T_2 were used for training image operators and T_3 was used as a test set. Later in [8] a fourth part T_4 was added, which used T_3 as validation and T_4 as test. In this work we train the kernelized operators in $T_1 \cup T_2$ and test in T_3 and T_4.

We compared our method with four earlier works in operator learning: IT [17], WER [8], Manual [5] and NILC [6]. We determined the parameters of the polynomial kernel (degree d and number of components m) using a validation set and the parameter grid $d = \{3, 5, 7, 9\}$ and $m = \{100, 500, 1000, 2000\}$. The chosen parameters are shown in Table I. We set the penalty parameter $C = 1$ for the SVMs.

IT, WER, Manual and NILC are based on the two level approach [5] for image operator learning. Two level operators combine the results of various image operators trained with different windows. This technique was shown to improve the performance when compared to individual operators trained using large windows.

The *staffs* dataset is a subset of the CVC-MUSCIMA dataset published in [1]. We chose the subset of images deformed by both adding ink splots and the 3D surface deformation and used its corresponding test set as described in [1]. The objective is to remove the stafflines from digitized music scores, as shown in Figure 1.

We compared our method with both earlier two-level operators (FS [7], Manual [21] and NILC) and heuristic methods

Domain	Method	*CharS*		*TexRev*	
		T_3	T_4	T_3	T_4
9×7	NILC	0.0040	0.0045	**0.029**	**0.017**
	KA	**0.0039**	**0.0042**	0.034	0.024
	IT	0.007^3	–	–	–
	WER	0.0041^2	0.0046^2	0.037^2	0.022^2
	Manual	0.006^1	0.005^2	0.046^1	0.026^2
11×9	NILC	0.0037	**0.0040**	-	-
	KA	**0.0035**	0.0043	-	-
	Manual	0.004^1	-	-	-
11×11	NILC	-	-	0.024	**0.013**
	IT	-	-	0.031^3	-
	KA	-	-	**0.021**	0.014
	Manual	-	-	0.031^1	-

TABLE II: Comparative test errors for datasets *CharS* and *TexRev*. A "-" indicates that the authors of the method did not test a certain combination of window size and training set.

Method	Acc (%)	Rec (%)	Spec (%)	F1
LTC	87.58	64.66	**99.53**	80.40
Skeleton	94.50	86.97	99.03	92.24
LRDE	97.03	94.02	98.84	95.97
FS	**97.34**	**95.87**	98.22	95.90
Manual	96.89	94.37	98.41	95.51
NILC	97.10	94.58	98.61	**96.07**
KA	96.23	92.85	98.26	94.88

TABLE III: Comparative results, accuracy (Acc), recall (Rec), specificity (Spec) and F1-measure (F1), for the *staffs* dataset.

(LRDE [23], Skeleton and LTC [24]). All methods were reexecuted with their default parameters. The authors of all algorithms have published implementations of their methods in the corresponding articles.

We present our results in Table III. KA achieved better results than Skeleton and LTC, but was inferior to the two level methods and LRDE. It is worth noting that (i) we used a smaller window (11×11) than FS, Manual and NILC (11×17); (ii) these methods are the result of a combination of many individual operators; and (iii) due to size of the representation ($m = 2000$), KA was trained using only a fraction of the examples available to FS, Manual and NILC. Taking these into account, KA performed surprisingly well for an individual image operator trained with only a small amount of data.

B. Gray level images and the Gaussian Kernel

The task of interest in the *DRIVE* dataset is the segmentation of blood vessels in retinal images. An example of input-output is shown in Figure 2. This dataset contains 20 images for training and another 20 for testing. Images contain different brightness and contrast.

To process these images we trained a kernelized version of the Aperture operator [4] with $a = 10$ and a 9×9 window. We determined the gaussian kernel parameters using a validation set and the parameter grid $\gamma \in \{0.001, 0.01, 0.1\}$ and $m \in \{100, 200, 500, 1000, 2000\}$.

We compared our technique with 3 heuristic methods in the literature: Staal [2], Niemeijer [25] and Zana [26]. The results are shown in Table IV. KA achieved accuracy close to the tested methods, beating Zana and scoring just 0.5% less than the best method (Staal). As we can see in Figure 4, our method correctly identifies most of the larger vessels and even some of the smallest ones. However, it produced noisier images than the other methods. Both Staal and Niemeijer produce clean images, although Staal seems to recognize small vessels better. Zana failed to identify many small vessels and also missed their width. In general, all methods were unable to segment the smallest vessels. The presence of noise in the outputs of KA accounted for most of the differences between our method and the heuristic ones. Result images for Staal, Niemeijer and Zana were obtained in http://www.isi.uu.nl/Research/Databases/DRIVE/browser.php.

Method	Accuracy (%)
Staal	**9**4.42
Niemeijer	94.16
Zana [26]	93.77
KA	93.91

TABLE IV: Results for the *DRIVE* dataset.

C. Discussion of the results

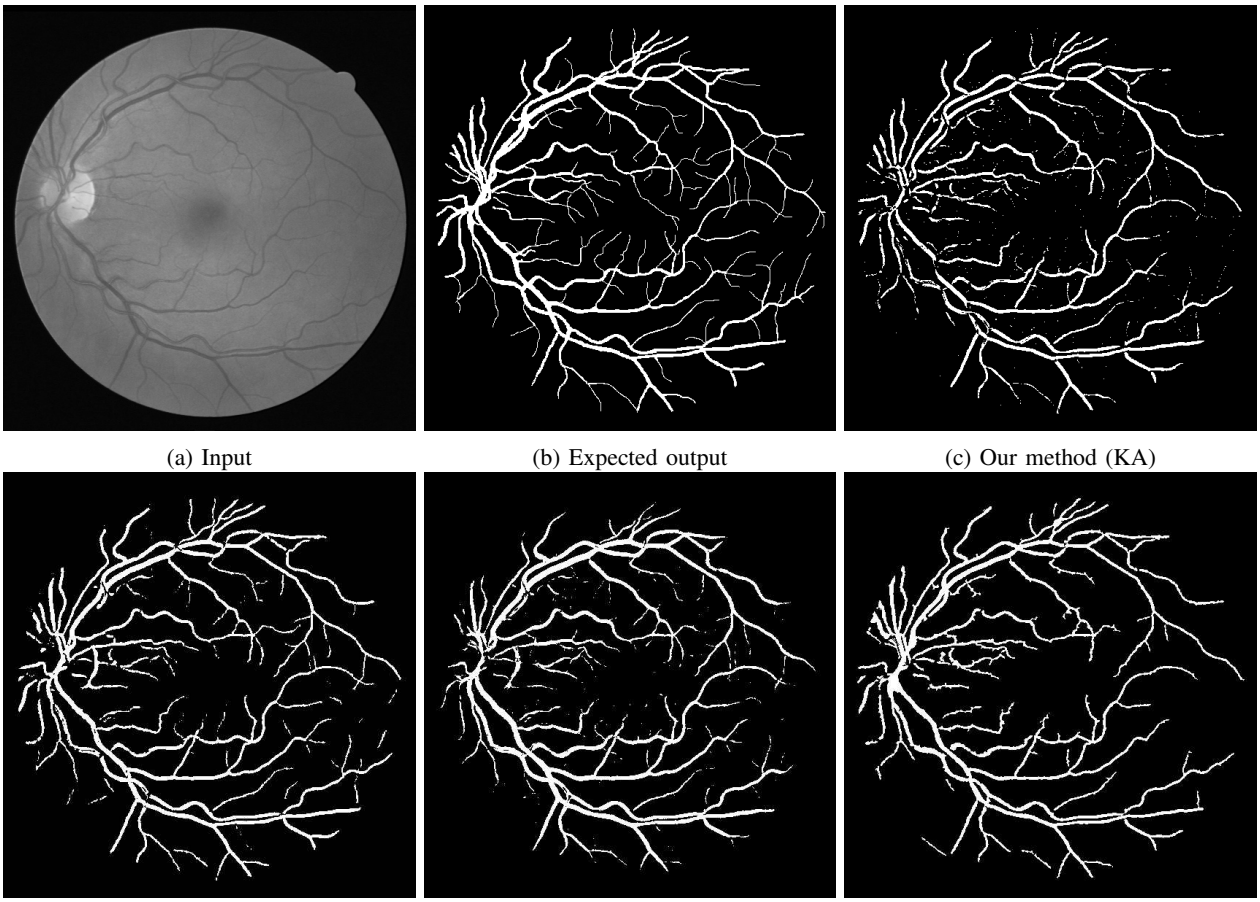
The KA operators achieved promising results in all four datasets. In the case of *staffs* and *DRIVE*, its results were comparable to complicated heuristic methods designed to process only images of their specific domain.

Due to the size of the computed features, training a SVM using the KA approach requires more memory than other non linear methods such as Decision Trees or Neural Networks. However, it still requires much less memory than solving the dual SVM problem using the original features. Memory requirement was the main bottleneck when training operators in *DRIVE* and *staffs*. We were only able to use a fraction of the available data. Thus, we should be able to improve our results if we reduce the memory footprint of the method.

Another point of improvement could be the combination of the two level and KA approaches. Although a set of KA operators can be readily combined into a two level operator, there are still many question regarding the parameters and number of operators necessary.

Finally, the determination of the parameters of the approximated kernel also reveals interesting information about KA. We show in Figure 5 the validation error of all tested parameter sets. In all cases, increasing the number of components m offers diminishing gains in accuracy. When using the polynomial kernel, the degree seems to be less important than the number of components used in the approximation. Also, for approximations of polynomials with higher degrees the difference between training and accuracy grew considerably, which may be a sign that they are more prone to overfitting. Using polynomials with low degrees is also computationally advantageous, since it is cheaper to compute $K_{\mathbf{x}}$ in these cases.

In the Gaussian kernel case, we see in Figure 5 that the correct estimation of this parameter is essential to obtain good performance.



(a) Input (b) Expected output (c) Our method (KA)

(d) Staal (e) Niemeijer (f) Zana

Fig. 4: Result images for the *DRIVE* dataset.

VI. Conclusion

We presented a method to train kernelized SVMs for image operator learning by approximating their feature maps. We have also selected appropriate kernels to process binary and gray level inputs and have discussed their visual interpretations. Experiments have shown encouraging results. Our method achieved comparable results with most state-of-the-art techniques.

In the future we plan to train combinations of kernelized operators and study how to build them effectively. Also, we plan to apply large scale optimization techniques to be able to use the training set fully in the bigger datasets (*DRIVE* and *staffs*). Finally, we may study using GPUs to accelerate both training and classification steps.

Acknowledgment

This work is supported by FAPESP (2015/17741-9, 2011/50761-2, 2015/01587-0) and by CNPq (484572/2013-0). Igor S. Montagner is supported by FAPESP (2014/21692-0, 2011/23310-0). N. S. T. Hirata and R. Hirata Jr. are partially supported by CNPq.

References

[1] M. Visaniy, V. Kieu, A. Fornes, and N. Journet, "ICDAR 2013 music scores competition: Staff removal," in *12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 1407–1411.

[2] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.

[3] J. Barrera, R. Terada, R. J. Hirata, and N. Hirata, "Automatic programming of morphological machines by pac learning," *Annales Societatis Mathematicae Polonae. Series 4: Fundamenta Informaticae*, vol. Vol. 41, Nr 1,2, pp. 229–258, 2000.

[4] R. Hirata Jr., M. Brun, J. Barrera, and E. R. Dougherty, "Aperture filters: theory, application, and multiresolution analysis," in *Advances in Nonlinear Signal and Image Processing*, ser. EURASIP Book Series on Signal Processing and Communications, S. Marshall and G. L. Sicuranza, Eds. Hindawi, 2006, vol. 6, pp. 15–48.

[5] N. S. T. Hirata, "Multilevel training of binary morphological operators," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 707–720, April 2009.

[6] I. S. Montagner, N. S. T. Hirata, R. Hirata Jr., and S. Canu, "NILC: a two level learning algorithm with operator selection," in *To appear at IEEE International Conference on Image Processing 2016 (ICIP 2016)*.

[7] I. Montagner, R. Hirata Jr., and N. S. T. Hirata, "A machine learning based method for staff removal," in *22nd International Conference on Pattern Recognition (ICPR)*, 2014, pp. 3162 – 3167.

[8] M. Dornelles and N. Hirata, "Selection of windows for w-operator combination from entropy based ranking," in *28th Conference on Graphics, Patterns and Images (SIBGRAPI)*, Aug 2015, pp. 64–71.

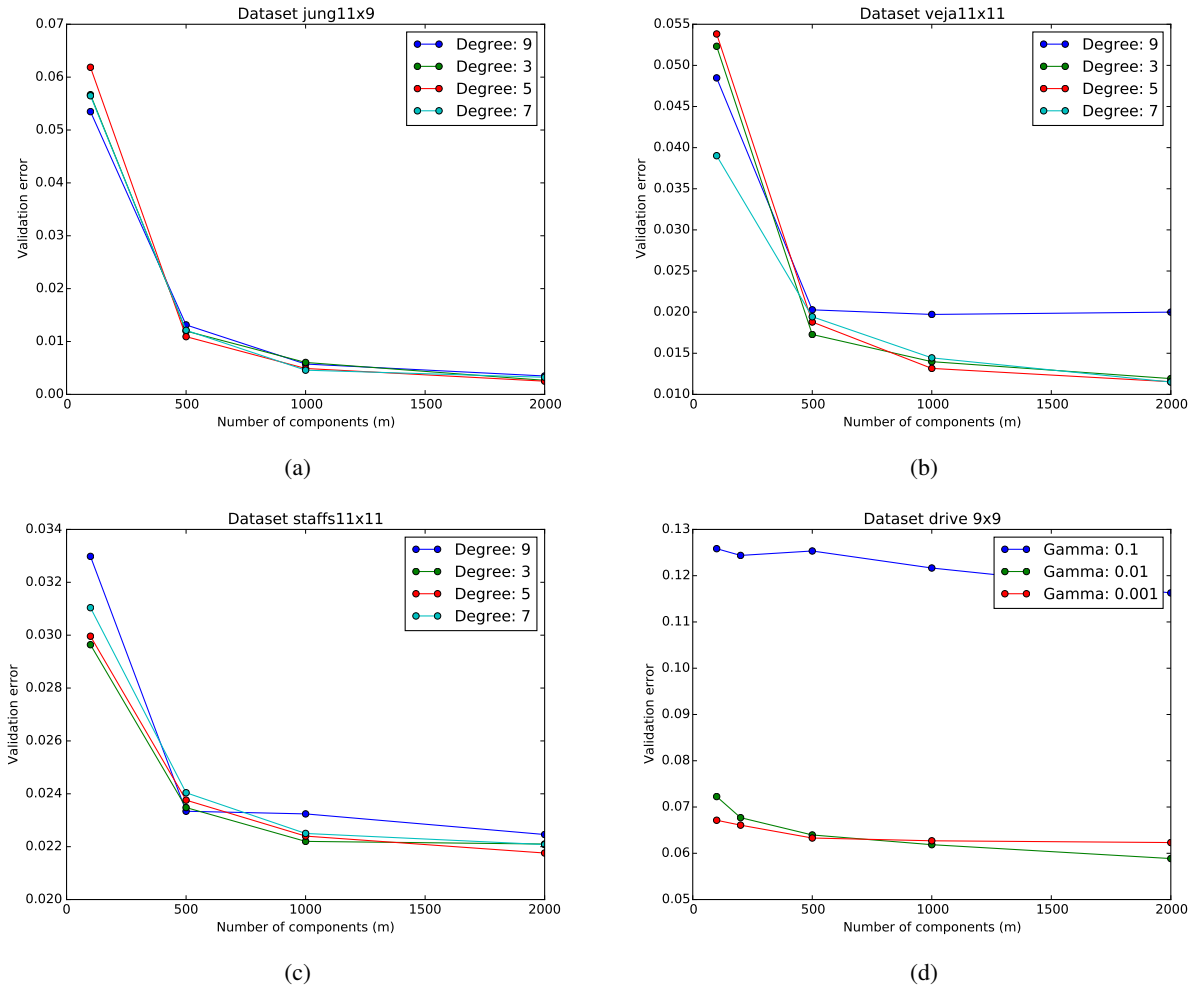


Fig. 5: Validation error w.r.t. to changes in the kernel's parameters (see details in text).

[9] N. S. T. Hirata, R. Hirata Jr., and J. Barrera, "Basis computation algorithms," in *Mathematical Morphhology and its Applications to Signal and Image Processing (Proceedings of the 8th International Symposium on Mathematical Morphology)*, 2007, pp. 15–26.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

[11] V. Vapnik, *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[12] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 682–688.

[13] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1313–1320.

[14] T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, and Z.-H. Zhou, "Nyström method vs random fourier features: A theoretical and empirical comparison," in *Advances in neural information processing systems*, 2012, pp. 476–484.

[15] Z. Lu, A. May, K. Liu, A. B. Garakani, D. Guo, A. Bellet, L. Fan, M. Collins, B. Kingsbury, M. Picheny *et al.*, "How to scale up kernel methods to be as good as deep neural nets," *arXiv preprint arXiv:1411.4000*, 2014.

[16] P.-S. Huang, H. Avron, T. N. Sainath, V. Sindhwani, and B. Ramabhadran, "Kernel methods match deep neural networks on timit," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 205–209.

[17] C. S. Santos, N. S. Hirata, and R. Hirata Jr., "An information theory framework for two-stage binary image operator design," *Pattern Recognition Letters*, vol. 31, no. 4, pp. 297–306, 2010.

[18] J. Barrera, E. R. Dougherty, and N. S. Tomita, "Automatic Programming of Binary Morphological Machines by Design of Statistically Optimal Operators in the Context of Computational Learning Theory," *Electronic Imaging*, vol. 6, no. 1, pp. 54–67, January 1997.

[19] H. J. A. M. Heijmans, *Morphological Image Operators*. Boston: Academic Press, 1994.

[20] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[21] I. Montagner, R. Hirata Jr., and N. S. T. Hirata, "Learning to remove staff lines from music score images," in *IEEE International Conference on Image Processing (ICIP)*, Paris, France, Oct. 2014, pp. 2614 – 2618.

[22] R. Hirata Jr., E. R. Dougherty, and J. Barrera, "Aperture Filters," *Signal Processing*, vol. 80, no. 4, pp. 697–721, April 2000.

[23] T. Géraud, "A morphological method for music score staff removal," in *Proceedings of the 21st International Conference on Image Processing (ICIP)*, Paris, France, 2014, pp. 2599–2603.

[24] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, "A comparative study of staff removal algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 753–766, 2008.

[25] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M. Abramoff, "Comparative study of retinal vessel segmentation methods on a new publicly available database," in *SPIE Medical Imaging*, J. M. Fitzpatrick

and M. Sonka, Eds., vol. 5370, SPIE. SPIE, 2004, pp. 648–656.
[26] F. Zana and J.-C. Klein, "Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation," *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1010–1019, 2001.