

# Laguerre Kernels –Based SVM for Image Classification

Ashraf Afifi

Computer Engineering Department  
Faculty of Computers and Information Technology  
Taif University  
Taif, KSA

**Abstract**—Support vector machines (SVMs) have been promising methods for classification and regression analysis because of their solid mathematical foundations which convey several salient properties that other methods hardly provide. However the performance of SVMs is very sensitive to how the kernel function is selected, the challenge is to choose the kernel function for accurate data classification. In this paper, we introduce a set of new kernel functions derived from the generalized Laguerre polynomials. The proposed kernels could improve the classification accuracy of SVMs for both linear and nonlinear data sets. The proposed kernel functions satisfy Mercer’s condition and orthogonally properties which are important and useful in some applications when the support vector number is needed as in feature selection. The performance of the generalized Laguerre kernels is evaluated in comparison with the existing kernels. It was found that the choice of the kernel function, and the values of the parameters for that kernel are critical for a given amount of data. The proposed kernels give good classification accuracy in nearly all the data sets, especially those of high dimensions.

**Keywords**—Laguerre polynomials; kernel functions; functional analysis; SVMs; classification problem

## I. INTRODUCTION

Improving efficacy of classifiers have been an extensive research area in machine learning over the past two decades, which led to state-of-the-art classifiers like support vector machines, neural networks and many more. Support vector machine (SVM) is a robust classification tool, effectively over comes many traditional classification problems like local optimum and curse of dimensionality[1].Support vector machines (SVMs) algorithm [2-3] has been shown to be one of the most effective machine learning algorithms. It gives very good results in terms of accuracy when the data are linearly or non-linearly separable. When the data are linearly separable, the SVMs result is a separating hyperplane, which maximizes the margin of separation between classes, measured along a line perpendicular to the hyperplane. If data are not linearly separable, the algorithm works by mapping the data to a higher dimensional feature space (where the data becomes separable) using an appropriate kernel function and a maximum margin separating hyperplane is found in this space. Thus the weight vector that defines the maximal margin hyperplane is a sufficient statistic for the SVMs algorithm (it contains all the information needed for constructing the separating hyperplane). Since this weight vector can be

expressed as a weighted sum of a subset of training instances, called support vectors, it follows that the support vectors and the associated weights also constitute sufficient statistics for learning SVMs from centralized data.

One issue for improving the accuracy of SVMs is finding an appropriate kernel for the given data to improve the accuracy of SVMs. Most research relies on a priori knowledge to select the correct kernel, and then tweaks the kernel parameters via machine learning or trial-and-error. While there exist rules-of-thumb for choosing appropriate kernel functions and parameters, this limits the usefulness of SVMs to expert users, especially since different functions and parameters can have widely varying performance. Williamson et al.[4] published a method for the use of entropy numbers in choosing an appropriate kernel function. It was an attempt to explain kernel function choice by more analytical means rather than previous ad-hoc or empirical methods. The entropy numbers associated with mapping operators for Mercer kernels is discussed. In [5], it was stated that previous work on invariance transformations was mostly appropriate only for linear SVM classifiers. For non-linear SVM classifiers, an analytical method of utilizing kernel principal component analysis (PCA) map for incorporating invariance transformations was presented in[6].

Tsang et al.[7] discussed a way to take advantage of the approximations inherent in kernel classifiers, by using the Minimum Enclosing Ball algorithm as an alternative means of speeding up training. Training time had previously been reduced mostly by modifying the training set in some way. Their final classifiers, which they called the Core Vector Machine, converged in linear time with space requirements independent of the number of data points. Zanaty and Aljahdali [8] investigated the performance of different kernels when they are applied to different data sets. Zanaty et al. [9-10] combined GF and RBF functions in one kernel called “universal kernel” to take advantage of their respective strengths. The universal kernels constructed the most established kernels such as radial bases, gauss, and polynomial functions by optimizing the parameters using the training data. SedatOzer et al., [11] introduced a set of new kernel functions derived from the generalized Chebyshev polynomials, where the generalized Chebyshev kernel approaches the minimum support vector number and maximum classification performance. Zhi-Bin Pan et al. [12] introduced support vector machine based on orthogonal Legendre polynomials, to reduce

the redundancy in feature space due to the orthogonality of Legendre polynomials, which may enable the SVM to construct the separating hyperplane with less support vectors. These kernels satisfy Mercer's condition and converge faster than the existing kernels.

Completely achieving a SVM with high accuracy classification therefore, requires specifying high quality kernel function. In this paper, a new set of Laguerre functions is introduced that could improve the classification accuracy of SVMs. A class of Laguerre kernel functions on the basis of the properties of the common kernels is proposed, which can find numerous applications in practice. The proposed set of kernel functions provides competitive performance when compared to all other common kernel functions on average for the simulation datasets. The results indicate that they can be used as a good alternative to other common kernel functions for SVM classification in order to obtain better accuracy.

The rest of this paper is organized as follows: In section 2, SVM classifiers are discussed. The kernel functions are discussed in section 3. The generalized Legendre kernels are discussed in section 4. Section 5 presents the functional analysis of the proposed Laguerre kernels. Experimental and comparative results are given in section 6. Finally, section 7 shows the conclusion.

## II. SVM CLASSIFIER

SVMs [14] are a relatively new approach for creating classifiers that have become increasingly popular in the machine learning community. They present several advantages over other methods like neural networks in areas like training speed, convergence, complexity control of the classifier, as well as a stronger mathematical background based on optimization and statistical learning theory. In the novel learning paradigm embodied in support vector machines "learning" (selection, identification, estimation, training or tuning), the parameters are not predefined and their number depends on the training data used [14-15]. The support vector machines combine two main ideas. The first one is concept of an optimum linear margin classifier, which constructs a separating hyperplane that maximizes distances to the training point. The second one is concept of a kernel. In its simplest form, the kernel is a function which calculates the dot product of two training vectors. Kernels calculate this dot product in feature space, often without explicitly calculating the feature vectors, operating directly on the input vectors instead. When we use feature transformation, which reformulates input vector into new features, the dot product is calculated in feature space, even if the new feature space has higher dimensionality. So the linear classifier is unaffected. Margin maximization provides a useful trade off with classification accuracy, which can easily lead to overfitting of the training data. Consider an input space  $X$  with input vectors  $x$ , a target space  $Y = \{1, -1\}$  and a training set  $T_r = \{(x_1, y_1), \dots, (x_N, y_N)\}$  with  $x_i \in X$  and  $y_i \in Y$ . In SVM classification, separation of the two classes  $Y = \{1, -1\}$  is done by means of the maximum margin hyperplane, i.e. the hyperplane that maximizes the distance to the closest data points and guarantees the best generalization on new, unseen examples. Let us consider two hyperplanes:

$$\langle w, x_i \rangle + b \geq 1 \quad \text{if } (y_i = 1) \quad (1)$$

$$\langle w, x_i \rangle + b \leq -1 \quad \text{if } (y_i = -1) \quad (2)$$

The distance from the hyperplane to a point  $x_i$  can be written:

$$d(w, b; x_i) = \frac{|\langle w, x_i \rangle + b|}{\|w\|}$$

Consequently the margin between two hyperplanes can be written as:

$$\min_{x_i: y_i} d(w, b; x_i) + \min_{x_i: y_i} -d(w, b; x_i)$$

To maximize this margin we have to minimize  $\|w\|$ . This comes down to solving a quadratic optimization problem with linear constraints. Notice however that we assumed that the data in  $T_r$  are perfectly linear separable. In practice however this will often not be the case.

Therefore we employ the so called soft-margin method in contrast to the hard-margin method. Omitting further details we can rewrite the soft-margin optimization problem by stating the hyperplane in its dual form, i.e. find the Lagrange multipliers  $\alpha_i \geq 0$  ( $i = 1, \dots, N$ ) that *Maximize* :

$$L(\alpha_1, \dots, \alpha_N) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \quad (3)$$

$$\text{subject to } \sum_{i=1}^N \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C$$

Considering the dual problem above, we can now write the maximum margin hyperplane as a linear combination of support vectors. By definition, the vectors  $x_i$  corresponding with non-zero  $\alpha_i$  are called the support vectors and this set consists of those data points that lie closest to the hyperplane and thus are the most difficult to classify. In order to classify a new point  $x_{new}$ , one has to determine the sign of

$$\sum_{x_i \in SV} \alpha_i y_i \langle x_i, x_{new} \rangle + b \quad (4)$$

If this sign is positive  $x_{new}$  belongs to class 1, if negative to class -1, if zero  $x_{new}$  lies on the decision boundary. Note that we have restricted the summation to the set of support vectors because the other  $\alpha_i$  are zero anyway.

## III. KERNEL FUNCTIONS

Support vector machine is one of kernel-based learning algorithms that consist of a learning algorithm and the kernel function [16, 17]. The kernel function creates the hypothesis space where the learning process searches for. The kernel can be considered as a similarity measure between two inputs which corresponds to their inner product in some feature space into which the original inputs are mapped. This is very useful, for instance, when the concept to be learned depends nonlinearly on the data, but the learning algorithm is able to learn only linear dependencies.

Since support vector machines are linear classifiers, it is necessary to map the input vectors with a nonlinear mapping in order to learn non-linear relations. The resulting vectors are usually called features. Formally, let  $k$  denote the input space, which can be any set, and  $F$  denote the feature vector space. For any mapping:

$$\phi: k \rightarrow F$$

The inner product of the mapped inputs is called a kernel function:

$$k(x, z) = \langle \phi(x), \phi(z) \rangle$$

A necessary condition for this is that  $k(x, z)$  is symmetric and finitely positive semi definite [18-19]. There are many different types of kernels that can be found in the literature [18-19].

#### IV. PROPOSED KERNELS

A critical step in support vector machine classification is choosing a suitable kernel of SVMs for a particular application, i.e. various applications need different kernels to get reliable classification results. It is well known that the two typical kernel functions often used in SVMs are the radial basis function kernel and polynomial kernel. More recent kernels are presented in [9-12,20-23] to handle high dimension data sets and are computationally efficient when handling non-separable data with multi attributes. However, it is difficult to find kernels that are able to achieve high classification accuracy for a diversity of data sets. In order to construct kernel functions from existing ones or by using some other simpler kernel functions as building blocks, the closure properties of kernel functions are essential [16-18].

For given non-separable data, in order to be linearly separable, a suitable kernel has to be chosen. Classical kernels, such as Gauss RBF and POLY functions, can be used to transfer non-separable data to separable, but their performance in terms of accuracy is dependent on the given data sets. The following POLY function performs well [20] with nearly all data sets, except high dimension ones:

$$k(x, z) = (x^T z + 1)^d$$

where  $d$  is the polynomial degree.

The same performance [20] is obtained with the Gauss RBF of the following form:

$$k(x, z) = \exp(-\gamma \|x - z\|^2)$$

where  $\gamma$  is appositive parameter controlling the radius. Zanaty et al in [9] presented the polynomial Radial basis function (RBPf) as:

$$PRBF = ((1 + \exp(\theta)) / V)^d$$

Where  $\theta = |x - z|, V = p * d$

where  $p$  is a parameter. Zanaty et al in [10] presented Support vector machines (SVMs) with universal kernels,

called Gaussian radial basis polynomials function (GRPF) given by:

$$GRBF = \left( \frac{b + e^{-\gamma^a \|x-x\|^r}}{V} \right), V = PD, a=b-1, r >= 1.$$

where  $a, b, \gamma = \frac{1}{2\sigma^2}$ , and  $d$  are the kernel parameters for the Gaussian, polynomial and universal kernels, respectively.  $\beta$  and  $\sigma$  are the scaling parameters for the polynomial kernel and determines the width of the Gaussian kernel respectively. Kernel functions should be applied onto input vectors directly instead of applying them onto eachelement and combining the results by a product, since the kernel functions are supposed to provide a measure of the correlation of two input vectors in a higher dimensional space.

#### A. Laguerre polynomials

The Laguerre polynomials are defined by the equation:

$$(1-t)^{-1} \exp\left(-\frac{xt}{1-t}\right) = \sum_{n=0}^{\infty} \frac{L_n(x)}{n!} t^n \quad (5)$$

The exponential function can be expanded to give:

$$\sum_{n=0}^{\infty} \frac{L_n(x)}{n!} t^n = \frac{1}{1-t} \sum_{\gamma=0}^{\infty} \frac{(-1)^\gamma x^\gamma t^\gamma}{\gamma! (1-t)^\gamma} = \sum_{\gamma=0}^{\infty} \frac{(-1)^\gamma x^\gamma t^\gamma}{\gamma!} (1-t)^{-(\gamma+1)} \quad (6)$$

Recall the binomial expansion:

$$(1-t)^{-m} = 1 + (-m)(-t) + \frac{(-m)(-m-1)}{2!} (-t)^2 + \dots = \sum_{s=0}^{\infty} \frac{m(m+1)\dots(m+s-1)}{s!} t^s \quad (7)$$

using the notation:

$$(\alpha)_\gamma = \alpha(\alpha+1)(\alpha+2)\dots(\alpha+\gamma-1) \quad (8)$$

Equation (6) may therefore be written as

$$\sum_{n=0}^{\infty} \frac{L_n(x)}{n!} t^n = \sum_{\gamma=0}^{\infty} \sum_{s=0}^{\infty} \frac{(-1)^\gamma (r+1)_s x^\gamma t^{\gamma+s}}{r! s!} \quad (9)$$

Equating powers of  $t^n$ , we get:

$$L_n(x) = n! \sum_{\gamma=0}^n \frac{(-1)^\gamma (r+1)_{n-\gamma}}{r! (n-\gamma)!} x^\gamma \quad (10)$$

The summation is taken from  $r=0$  to  $r=n$ , in view of the factor  $(n-r)!$  in the denominator. Note that:

$$(r+1)_{n-\gamma} = (r+1)(r+2)\dots(r+1+n-r-1) = \frac{n!}{r!} \quad (11)$$

$$(-n)_\gamma = (-n)(-n+1)\dots(-n+r-1) = (-1)^\gamma n(n-1)\dots(n-r+1) = (-1)^\gamma \frac{n!}{(n-r)!} \quad (12)$$

$$(1)_\gamma = (1)(2)\dots(1+r-1) = r! \quad (13)$$

Equation (5) then gives

$$L_n(x) = n! \sum_{\gamma=0}^n \frac{(-n)_{\gamma}}{(1)_{\gamma}} \cdot \frac{x^{\gamma}}{r!} \quad (14)$$

Or  $L_n(x) = n! F_1(-n; 1; x)$  (15)

Note that the series in Equation (14) terminates after  $n$  terms, i.e.  $L_n(x)$  is a polynomial of degree  $n$  (see Fig.(1)).

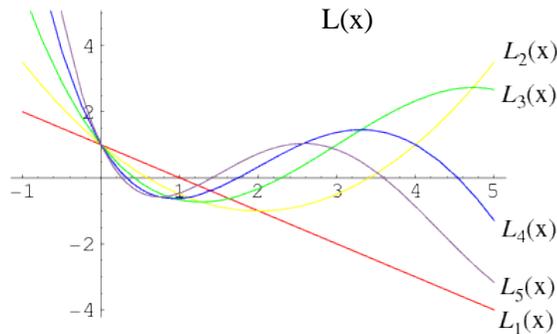


Fig. 1. The Laguerre function for the first five polynomials.

**B.1 Rodrigues' Formula for the Laguerre polynomial**  
Using Equation (11), Equation (11) may be written as

$$L_n(x) = e^x \sum_{\gamma=0}^n \frac{n!}{r!(n-r)!} (-1)^{\gamma} e^{-x} \frac{n!}{[n-(n-r)]!} x^{n-(n-\gamma)} \quad (16)$$

Recall the Leibniz formula for the  $m^{\text{th}}$  derivative of a product:

$$\frac{d^m}{dx^m} (AB) = \sum_{\gamma=0}^m \frac{m!}{\gamma!r!(m-r)!} \left( \frac{d^{\gamma}}{dx^{\gamma}} A \right) \left( \frac{d^{m-\gamma}}{dx^{m-\gamma}} B \right) \quad (17)$$

and that

$$\frac{d^p}{dx^p} x^n = \frac{n!}{(n-p)!} x^{n-p} \quad (n \geq p) \quad (18)$$

Equation (13) may therefore be written as

$$L_n(x) = e^x \sum_{\gamma=0}^n \frac{n!}{r!(n-r)!} \frac{d^{\gamma}}{dx^{\gamma}} e^{-x} \frac{d^{n-\gamma}}{dx^{n-\gamma}} x^n \quad (19)$$

or  $L_n(x) = e^x \frac{d^n}{dx^n} (e^{-x} x^n)$  (20)

Laguerre polynomials of low order can be evaluated by using the Rodrigues' formula (21) :

$$\begin{aligned} L_0(x) &= 1 \\ L_1(x) &= 1 - x \\ L_2(x) &= 2 - 4x + x^2 \\ L_3(x) &= 6 - 18x + 9x^2 - x^3 \end{aligned} \quad (21)$$

**B. Recurrence Relations**

We write the defining equation (5) in the form

$$\exp\left[x\left(1 - \frac{1}{1-t}\right)\right] = (1-t) \sum_{n=0}^{\infty} \frac{L_n(x)}{n!} t^n \quad (22)$$

Differentiating both sides with respect to  $t$ , we get

$$-\frac{x}{(1-t)^2} \exp\left[x\left(1 - \frac{1}{1-t}\right)\right] = (1-t) \sum_{n=0}^{\infty} \frac{L_n(x)}{(n-1)!} t^{n-1} - \sum_{n=0}^{\infty} \frac{L_n(x)}{(n-1)!} t^n \quad (23)$$

Multiplying through by  $(1-t)$ , we get

$$x \sum_{n=0}^{\infty} \frac{L_n(x)}{n!} t^n + (1-t)^2 \sum_{n=0}^{\infty} \frac{L_n(x)}{(n-1)!} t^{n-1} - (1-t) \sum_{n=0}^{\infty} \frac{L_n(x)}{n!} t^n = 0 \quad (24)$$

and equating coefficients of  $t^n$ , we obtain

$$\frac{xL_n(x)}{n!} + \frac{L_{n+1}(x)}{n!} - 2 \frac{L_n(x)}{(n-1)!} + \frac{L_{n-1}(x)}{(n-2)!} - \frac{L_n(x)}{n!} + \frac{L_{n-1}(x)}{(n-1)!} = 0 \quad (25)$$

and hence the recurrence relation

$$L_{n+1}(x) + (x - 2n - 1)L_n(x) + n^2 L_{n-1}(x) = 0 \quad (26)$$

If, on the other hand, we differentiate Equation (23) with respect to  $x$ , we get

$$-\frac{t}{1-t} \exp\left[x\left(1 - \frac{1}{1-t}\right)\right] = (1-t) \sum_{n=0}^{\infty} \frac{L'_n(x)}{n!} t^n \quad (27)$$

or  $t \sum_{n=0}^{\infty} \frac{L'_n(x)}{n!} t^n + (1-t) \sum_{n=0}^{\infty} \frac{L'_n(x)}{n!} t^n = 0$  (28)

Equating coefficients of  $t^n$  yields the identity

$$\frac{L'_n(x)}{n!} - \frac{L'_{n-1}(x)}{(n-1)!} + \frac{L_{n-1}(x)}{(n-1)!} + \frac{L_{n-1}(x)}{(n-1)!} = 0 \quad (29)$$

and hence the recurrence relation

$$L'_n(x) - nL'_{n-1}(x) + nL_{n-1}(x) = 0 \quad (30)$$

**C. Orthogonality of the Laguerre Polynomials**

Laguerre's differential equation can be cast into self-adjoint form by first writing it as

$$x \frac{d^2 y}{dx^2} + \frac{(1-x)}{x} \frac{dy}{dx} + \frac{n}{x} y = 0 \quad (31)$$

and multiplying throughout by the "integrating factor"

$$\exp\left(\int \frac{1-x}{x} dx\right) = \exp(\ln x - x) = xe^{-x}$$

This gives

$$xe^{-x} \frac{d^2 y}{dx^2} + (1-x)e^{-x} \frac{dy}{dx} + ne^{-x} y = 0 \quad (32)$$

$$\text{or } \frac{d}{dx} \left( x e^{-x} \frac{dy_n}{dx} \right) + n e^{-x} y_n = 0 \quad (33)$$

where we have included the subscript  $n$  in order to associate the solution  $y_n$  with the eigenvalue  $n$ . Replacing  $n$  by  $m$ , we have:

$$\frac{d}{dx} \left( x e^{-x} \frac{dy_m}{dx} \right) + m e^{-x} y_m = 0 \quad (34)$$

We now adopt the standard procedure: multiply Equation (39) by  $y_m$  and Equation (40) by  $y_n$ , and subtract. This gives:

$$(m-n)e^{-x} y_m(x) y_n(x) = y_n(x) \frac{d}{dx} [x e^{-x} y_m'(x)] - y_m(x) \frac{d}{dx} [x e^{-x} y_n'(x)] \quad (35)$$

Integrating both sides from  $0$  to  $\infty$ , and using the rule for the derivative of a product, we get

$$(m-n) \int_0^{\infty} e^{-x} y_m(x) y_n(x) dx = \int_0^{\infty} \frac{d}{dx} [x e^{-x} \{y_n(x) y_m'(x) - y_m(x) y_n'(x)\}] dx = 0 \quad (36)$$

It follows that if  $m \neq n$  the integral on the Left must be zero,

$$\text{i.e. } \int_0^{\infty} e^{-x} y_m(x) y_n(x) dx = 0, \quad \text{if } m \neq n \quad (37)$$

We have shown that the Laguerre polynomial  $L_n(x)$  is a solution of Laguerre's equation. We may therefore substitute  $y_n(x) = L_n(x)$  in Equation (37), to give:

$$\int_0^{\infty} e^{-x} L_m(x) L_n(x) dx = 0, \quad \text{if } m \neq n \quad (38)$$

The case  $m = n$  can be examined by noting that:

$$L_n(x) = e^x \frac{d^n}{dx^n} (x^n e^{-x}) = n! \sum_{\gamma=0}^n \frac{(-n)_{\gamma}}{r! r!} x^{\gamma} \quad (39)$$

Thus we have:

$$\int_0^{\infty} e^{-x} L_n(x) L_n(x) dx = n! \sum_{\gamma=0}^n \frac{(-n)_{\gamma}}{r! r!} \int_0^{\infty} e^{-x} x^{\gamma} e^x \frac{d^n}{dx^n} (x^n e^{-x}) dx \quad (40)$$

The Right Hand Side may be evaluated by integrating by parts  $n$  times. The procedure is as follows:

$$\int_0^{\infty} x^{\gamma} \frac{d^n}{dx^n} (x^n e^{-x}) dx = \int_0^{\infty} x^{\gamma} d \left[ \frac{d^{n-1}}{dx^{n-1}} (x^n e^{-x}) \right] = \left[ x^{\gamma} \frac{d^{n-1}}{dx^{n-1}} (x^n e^{-x}) \right]_0^{\infty} - \int_0^{\infty} \frac{dx^{\gamma}}{dx} \left[ \frac{d^{n-1}}{dx^{n-1}} (x^n e^{-x}) \right] dx \quad (41)$$

The first term on the Right Hand Side vanishes at both limits, so we obtain:

$$\int_0^{\infty} x^{\gamma} \frac{d^n}{dx^n} (x^n e^{-x}) dx = - \int_0^{\infty} \left( \frac{d}{dx} x^{\gamma} \right) \left[ \frac{d^{n-1}}{dx^{n-1}} (x^n e^{-x}) \right] dx = - \int_0^{\infty} \left( \frac{d}{dx} x^{\gamma} \right) d \left[ \frac{d^{n-2}}{dx^{n-2}} (x^n e^{-x}) \right] dx \quad (42)$$

$$\int_0^{\infty} x^{\gamma} \frac{d^n}{dx^n} (x^n e^{-x}) dx = (-1)^2 \int_0^{\infty} \left( \frac{d^2}{dx^2} x^{\gamma} \right) \left[ \frac{d^{n-2}}{dx^{n-2}} (x^n e^{-x}) \right] dx \quad (43)$$

A continuation of this process leads ultimately to the result:

$$\int_0^{\infty} x^{\gamma} \frac{d^n}{dx^n} (x^n e^{-x}) dx = (-1)^n \int_0^{\infty} \left( \frac{d^n}{dx^n} x^{\gamma} \right) \left[ \frac{d^{n-n}}{dx^{n-n}} (x^n e^{-x}) \right] dx \quad (44)$$

Hence

$$\int_0^{\infty} e^{-x} L_n(x) L_n(x) dx = n! \sum_{r=0}^n \frac{(-n)_{\gamma}}{r! r!} (-1)^n \int_0^{\infty} \left( \frac{d^n}{dx^n} x^{\gamma} \right) (x^n e^{-x}) dx \quad (45)$$

$$\text{Recall that } \frac{d^n}{dx^n} x^{\gamma} = \begin{cases} 0 & \text{if } r < n \\ n! & \text{if } r = n \end{cases} \quad (46)$$

Thus the only term in the summation (45) which survives is the  $r = n$  term; hence we obtain:

$$\int_0^{\infty} e^{-x} L_n(x) L_n(x) dx = n! \frac{(-n)_n (-1)^n n!}{n! n!} \int_0^{\infty} x^n e^{-x} dx \quad (47)$$

$$\text{Note that } (-n)_n = (-1)^n n! \quad (48)$$

$$\text{and } \int_0^{\infty} x^n e^{-x} dx = n! \quad (49)$$

So we finally obtain:

$$\int_0^{\infty} e^{-x} L_n(x) L_n(x) dx = (n!)^2 \quad (50)$$

This result may be combined with the orthogonality relation (38) to give:

$$\int_0^{\infty} e^{-x} L_m(x) L_n(x) dx = (n!)^2 \delta_{mn} \quad (51)$$

The weight function  $e^{-x}$  may be removed by defining a new function:

$$\phi_n(x) = \frac{1}{n!} e^{-x/2} L_n(x) \quad (52)$$

Equation (57) may then be written as:

$$\int_0^{\infty} \phi_m(x) \phi_n(x) dx = \delta_{mn} \quad (53)$$

Functions which satisfy the relation (54) are said to be *normalized*, and we say that the  $\phi_n(x)$  form an *orthonormal* set of functions.

V. GENERALIZED LAGUERRE KERNELS

Here, we propose a generalized way of expressing the kernel function to clarify the ambiguity on how to implement Laguerre kernels. To the best of our knowledge, there was no previous work defining the Laguerre polynomials for vector inputs recursively. Therefore for vector inputs, we define the generalized Laguerre polynomials as:

$$\begin{aligned}
 L_0(x) &= 1 \\
 L_0(z) &= 1 \\
 L_1(x) &= 1 - x \\
 L_1(z) &= 1 - z \\
 L_2(x) &= 0.5x^2 - 2x + 1 \\
 L_2(z) &= 0.5z^2 - 2z + 1 \\
 L_3(x) &= -0.17x^3 + 1.5x^2 - 3x + 1 \\
 L_3(z) &= -0.17z^3 + 1.5z^2 - 3z + 1 \\
 L_4(x) &= 0.042x^4 - 0.667x^3 + 3x^2 - 4x + 1 \\
 L_4(z) &= 0.042z^4 - 0.667z^3 + 3z^2 - 4z + 1 \\
 &\dots \\
 L_{n+1}(x) &= (x - 2n - 1)L_n(x) + n^2 L_{n-1}(x) = 0
 \end{aligned}
 \tag{54}$$

Therefore, the generalized Laguerre,  $L_j(x)$ , and  $L_j(z)$ , yield rowvectors, otherwise, it yields a scalar value. Thus by using generalized Laguerre polynomials, we define

generalized  $n^{\text{th}}$  order Laguerre kernel as

$$k(x, z) = \sum_{j=0}^n L_j(x) \cdot L_j^T(z)
 \tag{55}$$

Where  $x$  and  $z$  are  $m$ -dimensional vectors.

TABLE I. LIST OF THE GENERATED LAGUERRE KERNEL FUNCTIONS UP TO 4TH ORDER.

Kernel Parameter: n	Kernel function: k(x,z)
0	$L_0(x) * L_0(z)^T$
1	$K_0 + L_1(x) * L_1(z)^T$
2	$K_0 + K_1 + L_2(x) * L_2(z)^T$
3	$K_0 + K_1 + K_2 + L_3(x) * L_3(z)^T$
4	$K_0 + K_1 + K_2 + K_3 + L_4(x) * L_4(z)^T$
5	$K_0 + K_1 + K_2 + K_3 + L_4(x) * L_4(z)^T + L_5(x) * L_5(z)$
6	$K_0 + K_1 + K_2 + K_3 + L_4(x) * L_4(z)^T + L_5(x) * L_5(z) + L_6(x) * L_6(z)$
7	$K_0 + K_1 + K_2 + K_3 + L_4(x) * L_4(z)^T + L_5(x) * L_5(z) + L_6(x) * L_6(z) + L_7(x) * L_7(z)$

construct the mapping  $\phi$  from the eigenfunction decomposition of  $k$ . According to Mercer's work [24], it is known that if  $k$  is the symmetrical and continuous kernel of an integral operator  $O_k : L^2 \rightarrow L^2$ , such that:

$$O_{kg}(x) = \int K(x, z)g(z)dz$$

is positive, i.e.,

$$\int K(x, z)g(x)g(z)dx dz \geq 0 \quad \forall g \in L^2,$$

Then  $k$  can be expanded into a uniformly convergent series

$$K(x, z) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i^T(z),$$

With  $\lambda_i \geq 0$ . In this case, the mapping from input space to feature space produced by the kernel is expressed as

$$\phi : x \mapsto (\sqrt{\lambda_1} \phi_1(x), \sqrt{\lambda_2} \phi_2(x), \dots)$$

such that  $k$  acts as the given dot product, i.e.,

$$(\phi(x), \phi(z)) = \phi^T(x) \phi(z) = K(x, z).$$

**Theorem 1.** A nonnegative linear combination of Mercer kernels is also a Mercer kernel.

**Proof.** Let  $K_i (i = 1, \dots, M)$  be Mercer kernels and let

$$K(x, z) = \sum_{i=1}^M a_i K_i(x, z),$$

where  $a_i \geq 0$  is a nonnegative constant. According to Mercer's theorem, we have

$$\int K_i(x, z)g(x)g(z)dx dz \geq 0, \forall g \in L^2, i = 1, \dots, M.
 \tag{56}$$

By taking the sum of the positive combination of (56) with coefficients  $a_i$  over  $i$ , one obtains

$$\sum_{i=1}^M a_i \int K_i(x, z)g(x)g(z)dx dz \geq 0, \quad \forall g \in L^2.
 \tag{57}$$

Therefore, one reaches

$$\int K(x, z)g(x)g(z)dx dz \geq 0, \quad \forall g \in L^2.$$

In particular, if  $\sum_{i=1}^M a_i = 1, (a_i \geq 0)$ , then we regard  $K(x, z)$  in (57) as the convex combination of the positive definite kernels  $K_i(x, z)$ . This kind of kernel can find numerous applications in practice.

**Theorem 2.** The product of Mercer kernels is also a Mercer kernel.

**The proofs** similar to that of the preceding theorem.

**Theorem 3.** To be a valid SVM kernel, a kernel should satisfy the Mercer Conditions [26-27]. If the kernel does not satisfy the Mercer Conditions, SVM may not find the

optimal parameters, but rather it may find suboptimal parameters. Also if the Mercer conditions are not satisfied, then the Hessian matrix for the optimization part may not be positive definite. Therefore we examine if the generalized Laguerre kernel satisfies the Mercer conditions:

Mercer Theorem: To be a valid SVM kernel, for any finite function  $g(x)$ , the following integration should always be non-negative for the given kernel function  $k(x, z)$  [1]:

$$\iint K(x, z)g(x)g(z) dx dz \geq 0 \tag{58}$$

Where

$$k(x, z) = \sum_{j=0}^n L_j(x)L_j^T(z) = L_0(x)L_0^T(z) + L_1(x)L_1^T(z) + \dots + L_n(x)L_n^T(z) \tag{59}$$

Consider that  $g(x)$  is a function where  $g: \mathbf{R}^m \rightarrow \mathbf{R}$ , then we can evaluate and verify the Mercer condition for  $k(x, z)$  as follows by assuming each element is independent from others:

$$\begin{aligned} \iint k(x, z)g(x)g(z) dx dz &= \iint \sum_{j=0}^n L_j(x)L_j^T(z)g(x)g(z) dx dz \\ &= \sum_{j=0}^n \int L_j(x)L_j^T(z)g(x)g(z) dx dz = \sum_{j=0}^n \left[ \int L_j(x)g(x) dx \int L_j^T(z)g(z) dz \right] \\ &= \sum_{j=0}^n \left[ \left( \int L_j(x)g(x) dx \right) \left( \int L_j^T(z)g(z) dz \right) \right] \geq 0 \end{aligned} \tag{60}$$

Therefore, the kernel  $k(x, z)$  is a valid kernel.

## VI. EXPERIMENTAL RESULTS

The classification experiments are conducted on different data like Cloud, Liver, Seed, Forest Fire and Yeast data available at <http://archive.ics.uci.edu/ml/datasets.html>. These data sets have been given to the algorithm with different sizes (classes and attributes). Table II shows the classification accuracy for five different data sets using Laguerre kernel of order from 2 to 5 implementations. As shown in Figure 2, it is clear that when the order of polynomials increases, the accuracy increases for all data sets.

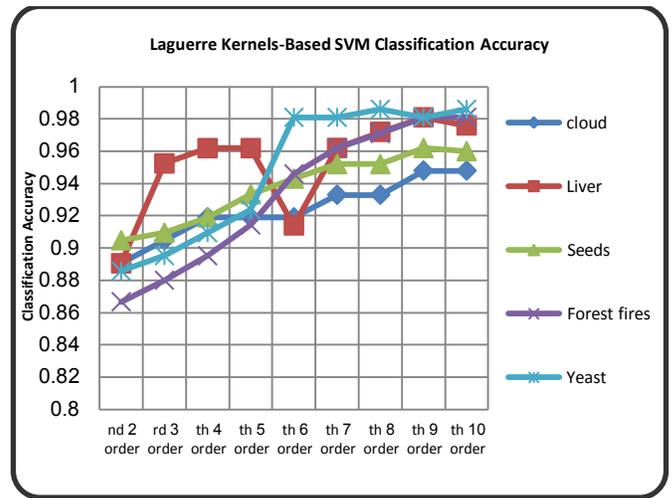


Fig. 2. Laguerre Kernels-Based SVM Classification Accuracy

### A. Comparative results

The performance of the proposed kernel with SVMs, in terms of classification accuracy, is evaluated by application to a variety of data sets available at:

<http://www.cs.toronto.edu/delve/data/image-set/desc.html>.

Firstly, we used LIBSVM with different kernels (linear, polynomial, radial basis function [8]). The parameters used include two parameters for the RBF kernel parameter  $\gamma=0.5$  and  $\sigma=0.5$ ,  $d=1$  for linear and  $d=5$  for polynomial kernels. Table II lists the main characteristics of the seven datasets used in the experiments. In order to evaluate the performance of the support vector machine with different kernels, we carried out some experiments with different data sets from machine learning benchmarks domains [28].

The data has even different classes of image. They contain 210 data for training and another 2100 data for testing. Each vector has 18 elements with different minimum and maximum values. For the training, we have 30 data for the class (+1) and 180 data for the class (-1) and similarly for testing. As can be seen from Table II, the generalized Laguerre kernel results show better generalization ability than the existing Gaussian, Polynomial (POLY) and Chebyshev [12] kernels.

TABLE II. CLASSIFICATION ACCURACY OF DIFFERENT DATA SETS USING LAGUERRE KERNELFUNCTION

Sr. No.	Name of Database	No. of features	No. of Classes	Training Size	Testing Size	Classification accuracy			
						2 <sup>nd</sup> order	3 <sup>rd</sup> order	4 <sup>th</sup> order	5 <sup>th</sup> order
1	cloud	1024	10	150	450	0.8905	0.9048	0.9190	<b>0.9190</b>
2	Liver	345	07	140	205	0.8905	0.9524	0.9619	<b>0.9619</b>
3	Seeds	210	07	105	210	0.9048	0.9095	0.9190	<b>0.9333</b>
4	Forest fires	517	13	195	210	0.8667	0.8800	0.8952	<b>0.9143</b>
5	Yeast	1484	08	150	500	0.8857	0.8952	0.9095	<b>0.9238</b>

For example, in Table II, the 5<sup>th</sup> order generalized Laguerre kernel results in classification accuracy of more than 96% for all test data sets, while the existing kernels achieve less than 96% for most test data sets. More specific, comparing the results of Tables III, the 5<sup>th</sup> order generalized Laguerre kernel always gives good results and may be the best at all, as shown in Figure 3.

TABLE III. RESULTS ON IMAGE SEGMENTATION DATA WITH VARIOUS KERNEL FUNCTIONS.

Data Segmentation	Linear Kernel	Gaussian Kernel	Polynomial Kernel	Chebyshev Kernel	Laguerre Kernel
Brickface	0.9952	0.9809	0.9904	0.9952	0.9952
Sky	1	0.9857	1	1	1
Foliage	0.9762	0.9571	0.9571	0.9714	0.9857
Cement	0.8952	0.9476	0.9714	0.9762	0.9667
Window	0.9524	0.9476	0.9667	0.9476	0.9762
Path	1	0.9761	0.9952	0.9762	0.9952
Grass	1	0.9809	1	0.9857	1

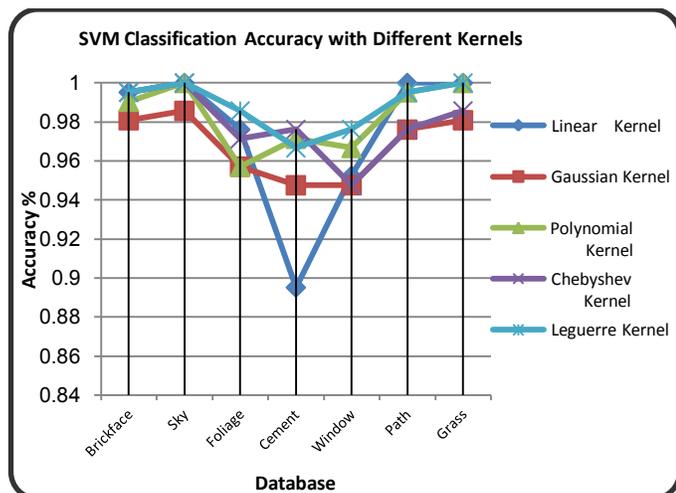


Fig. 3. SVM Classification Accuracy with different Kernels

## VII. CONCLUSION

In this paper, SVMs have been improved to solve the classification problems by mapping the training data into a feature space by the aid of Laguerre kernel functions and then separating the data using a large margin hyperplane. A class of Laguerre kernel functions on the basis of the properties of the common kernels is proposed, which can find numerous applications in practice.

Experimental results illustrate the validity and effectiveness of the proposed kernel. The experimental results

show that the proposed kernel function results in the best accuracy in nearly all the data sets especially in the data set with large number of attributes. The obtained results are encouraging and suggest that the proposed method is worth further consideration.

## REFERENCES

- [1] Vapnik V. N., "The nature of statistical learning theory", Springer-Verlag, New York, NY, USA, 1995.
- [2] Kim H., Pang S., Je H., Kim D., Bang S.Y., "Constructing support vector machine ensemble", Pattern Recognition, vol.36, no.12, pp.2757-2767, 2003.
- [3] Du P., Peng J., Terlaky T., "Self-adaptive support vector machines", modeling and experiments Computational Management Science, vol. 6, no.1, pp. 41-51, 2009.
- [4] Williamson, R., A. Smola, and B. Schölkopf, "Entropy numbers, operators and support vector kernels", pp. 44-127, Cambridge, MA: MIT Press, 1999.
- [5] Chapelle, O. and B. Schölkopf, "Incorporating invariances in non-linear support vector machines", In T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, Volume 14, pp. 594-609, Cambridge, MA: MIT Press, 2002.
- [6] Hansheng Lei & Venu Govindaraju, "Speeding up multi-class SVM evaluation by PCA and feature selection", Center for Unified Biometrics and Sensors (CUBS), 2005.
- [7] Tsang, I., J. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVMs training on very large data sets", Journal of Machine Learning Research 6, pp. 271-363, 2005.
- [8] E.A.Zanaty and Sultan Aljahdali "Improving the accuracy of support vector machines" accepted in 23rd International conference on computers and their application, Mexico, USA, April 9-11,2008.
- [9] Zanaty E.A, Sultan Aljahdali, R.J. Cripps, "Accurate support vector machines for data classification", Int. J. Rapid Manufacturing, vol. 1, no. 2, pp. 114-127, 2009.
- [10] Zanaty E.A, Ashraf Afifi, "Support vector machines (SVMs) with universal kernels", in International journal of Artificial Intelligence, vol. 25, pp.575-589, 2011.
- [11] Zhi-Bin Pan, Hong C., Xin-H., "Support vector machine with orthogonal legendre kernel" In Proceeding of the international conference on wavelet analysis and pattern recognition, pp. 15-17, 2012.
- [12] Sedat O., Chi H.C., Hakan A. Cirpan, "A set of new Chebyshev kernel functions for support vector machine pattern classification", Pattern Recognition, vol. 44, pp. 1435-1447, 2011.
- [13] Boser, B., Guyon I., and Vapnik V., "A training algorithm for optimal margin classifiers", In D. Haussler (Ed.), Annual ACM Workshop on COLT, Pittsburgh, PA, 144(52), ACM Press, 1992.
- [14] Kecman, "Support vector machines: Theory and applications", Springer, 2005.
- [15] Cortes C., Vapnik V.N., "Support-vector networks", Machine Learning, vol. 20, pp. 273-297, 1995.
- [16] Nello Cristianini and John Shawe-Taylor, "An Introduction to Support Vector Machines", Cambridge University Press, 2000.
- [17] Bishop C., Pattern Recognition and Machine Learning, Springer, 2006.
- [18] Shawe-Taylor J., and Cristianini N., "Kernel Methods for Pattern Analysis", Cambridge University 2004.
- [19] Shawe-Taylor J., Bartlett P.L., Williamson R.C., Anthony M., "Structural risk minimization over data-dependent hierarchies", IEEE Trans. Information Theory, vol. 44, no. 5, pp. 1926-1940, 1998.
- [20] Maji S., Berg A.C., Malik J., "Classification using intersection kernel support vector machines is efficient", IEEE Conference on Computer Vision and Pattern Recognition, June pp.1-8, 2008.
- [21] B. Haasdonk, "Feature space interpretation of SVMs with indefinite kernels", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 4, 2002.
- [22] E.A.Zanaty, Ashraf Afifi, Rania El-Khateeb, "Improving the accuracy of support vector machines via a new kernel functions", International journal of intelligent Computing, sciences, pp. 55-67, 2009.

- [23] Ying Tan, "A Support vector machine with a hybrid kernel and minimal Vapnik chervonenk is dimension", IEEE Transactions on knowledge and data engineering, vol. 16, no. 4, APRIL,2004
- [24] Mercer T. , "Functions of positive and negative type and their connection with the theory of integral equations", Philosophical Trans. of the Royal Soc. of London, Series A, pp. 415-446,1909.
- [25] WilliamsonR.C., SmolaA.J., and Scholkopf B., "Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators", Technical Report 19, Neuro COLT, 1998.
- [26] Scholkopf B., A.J. Smola, "Learning with kernels: support vector machines, regularization, optimization", and Beyond, MIT Press, 2001.
- [27] Vapnik V.N., "Statistical Learning Theory", Wiley-Interscience, New York, 1998.
- [28] <http://archive.ics.uci.edu/ml/datasets.html>.

#### AUTHOR'S PROFILE



Ashraf Afifi is an assistance professor, faculty of computers and information technology, Taif University, Saudi Arabia. He received his MSC Degree in digital communication in 1995 from zagazig University, Egypt. He completed his Ph. D. studies in 2002 from zagzig University,Egypt. His research interests are digital communication, and image segmentation. In these areas he has published several technical papers in refereed international journals or Conference proceedings.