ON IMAGE CATEGORIZATION USING GENETIC PROGRAMMING AND MULTIPLE KERNEL SUPPORT VECTOR MACHINE

Jyothi Korra
Computer Science and Engineering
Christ University Faculty of Engineering
Bangalore, India

ABSTRACT
In Computer Vision, problem of identifying or classifying the objects present in an image is called Image Categorization. It is challenging problem, especially when the images have clutter background, occlusions or different lighting conditions. Many vision features have been proposed which aid image categorization even in such adverse conditions. Past research has shown that, employing multiple features rather than any single features leads to better recognition. Multiple Kernel Learning (MKL) frameworkshave been developed for learning an optimal combination of features for object categorization. MKL on Support Vector Machines (SVMs) has been a popular front of research in recent times due to its success in application problems like Object Categorization. Main disadvantage with MKL methods of SVM is that only linear combination of base kernels can be learned efficiently, which may not be optimal for object categorization. Real-world object categorization may need to consider complex combination of kernels(non-linear) and not only linear combination. In this paper we do extensive comparative study of SVM MKL with Genetic Programming based MKL. Experiment results show that non-kernel generated using genetic programming gives good accuracy as compared to linear combination of kernels.

Keywords-genetic programming; non-linear kernel combination; object categorization; support vector machine; multiple kernel learning;

I.INTRODUCTION

Object Categorization is the problem of categorizing the given image into predefined classes of objects like face, car, bike etc. This problem is well studied in the field of Computer Vision. Object Categorization is a challenging problem, especially when the images have clutter background, occlusions or different lighting conditions. When a computer vision system or computer vision algorithm is designed the choice of feature representation can be a critical issue. In some cases, a higher level of detail in the description of a feature may be necessary for solving the problem, but this comes at the cost of having to deal with more data and more demanding processing. In this paper, an instance of a feature representation is referred to as a (feature) descriptor. In some applications like object categorization it is not sufficient to extract only one type of feature to obtain the relevant information from the image data. Instead many feature descriptors have been proposed which aid object categorization even in those adverse conditions. Each descriptor has its own merits and de-merits. Some descriptors are invariant to transformations while the others are more discriminative. Past research has shown that employing multiple feature descriptors rather than any single descriptor leads to better recognition. This paper focuses on the problem of learning the optimal combination of the available descriptors for a particular classification task.

Support Vector Machines (SVMs) [13] have emerged as powerful tools for classification problems. Key to the accurate classification using SVMs is the choice of Kernel functions (for definition of Kernel

function please see [7]). This issue was first studied in Lanckriet et. al. [8] where the problem of Multiple kernel learning (MKL) was first introduced. They have been successfully applied to a variety of domains e.g. text, object recognition [1, 2], protein structures [9]. Even though the idea was to explore the space of all possible linear combinations of the specified kernels, the functional frame- work associated with it could only select the best kernel from the set of specified kernels. Recently, many other approaches have been proposed to overcome this limitation [10]. While some of them select all the kernels and some have sparse solutions that choose a subset of the specified kernels in a weighted combination, none of them have explicit control over sparsity. In [1], [2], the authors employ the Multiple Kernel Learning (MKL) framework to find the optimal combination of feature descriptors (kernels). The goal of MKL is to simultaneously optimize the combination of kernels and the usual classification objective. Existing MKL methods for combining kernels are linear combinations of base kernels. But non-linear combination of base kernels may be helpful to boost performance of the classifier. This would be ideal for applications such as object categorization, in which combination of the descriptors is known to perform better than any single descriptor. This is the first attempt to have a comparative study of non-linear kernel combination with linear kernel combination for object categorization. Experiments areconducted on datasets like Caltech-5, Caltech-101 etc. to verify advantage of this hybrid model for improving object categorization. The outline of the paper is as follows: section II discusses the past work in this area. Section III briefly reviews genetic programming and its advantages. Section IV and V describes the methodology for carrying out object categorization using SVM linear kernel learning and genetic programming non-linear kernels. Section VI gives the implementation details and results on two datasets. This is followed by the conclusion.

II.PAST WORK

There has been some work in genetic programming for evolving kernels for Support Vector Machine [3], [4], [5]. [5] Uses Genetic Programming for evolving the kernel for SVM classifier. The approach presented there combines the two techniques of SVMs and GP, using the GP to evolve a kernel for a SVM. The goal there is to eliminate the need for testing various kernels and their parameter settings. They claim the approach might also be possibly used to discover new kernels that are particularly useful for the type of data under analysis. They show that their method performs better than manual choosing of the kernel and adjusting parameters. [4] Uses a set of standard kernels for evolving expression for new kernel, which performs better for given problem using genetic programming. Terminal set contains feature vectors; first level from terminals in the GP trees contains only standard kernels defined beforehand. Variable set contains functions, which take two kernels as arguments, and provides a kernel as output.[3] Tries to learn a regression function where kernels act as the regression variables. Each GP chromosome gives the complex combination of the set of kernels that is defined already. This is closely related to [3], where they try to evolve regression function using GP where kernels acts as regression variables. This paper studies how these non-linear functions of base kernels affect the performance of the object categorization. Even though these try to evolve kernels, no work has been done on evolving non-linear kernels for object categorization. The state-of-the-art works in object categorization considers many descriptors and try to find the optimal combination of the descriptors. [1], [2] considers combining descriptors using multiple kernel learning. Descriptors are extracted from the image and each descriptor will have many kernels formed using the feature vector of the descriptors. And these kernels are combined using the Multiple Kernel Learning(MKL) in Support Vector Machine(SVM) framework. The principle idea is to combine kernels in linearly. But real-world object categorization may perform better when we have non-linear combination of these descriptors. In our work we find non-linear kernel combination for improving performance. The next section explains genetic programming and how to evolve non-linear kernel combination using GP.

III. INTRODUCTION TO GENETIC PROGRAMMING

Genetic programming [6] is a systematic method for getting computers to automatically solve a problem. Genetic programming genetically breeds a population of computer programs to solve a problem. Specifically, genetic programming iteratively transforms a population of computer programs into a new generation of programs by applying analogs of naturally occurring genetic operations. The genetic operations include crossover (sexual recombination), mutation and reproduction. Analogs of developmental processes are sometimes used to transform an embryo into a fully developed structure. Genetic programming is an extension of the genetic algorithm in which the structures in the population are not fixed-length character strings that encode candidate solutions to a problem, but programs that, when executed, are the candidate solutions to the problem.

Programs are expressed in genetic programming as syntax trees rather than as lines of code. The tree includes nodes and links. The nodes indicate the functions to execute. The links indicate the arguments for each function. Leaf nodes are called terminals and internal nodes are called functions. More details on Genetic programming can be found on [6].

A. Preparatory Steps for GP

Genetic programming requires a set of parameters to start with. These parameters are listed below.

The set of terminals (e.g., the independent variables of the problem, zero- argument functions, and random constants) for each branch of the to-be- evolved program,

The set of primitive functions for each branch of the to-be-evolved program,

The fitness measure (for explicitly or implicitly measuring the fitness of individuals in the population),

Certain parameters for controlling the run, and

The termination criterion and method for designating the result of the run.

IV. MULTIPLE KERNEL SUPPORT VECTOR MACHINES

Goal: we want to find the hyper plane (i.e. decision boundary) linearly separating our classes. Our boundary will have equation: $w^T x + b = 0$. Anything above the decision boundary should have label 1, that is xisuch that $w^T x_i + b > 0$ will have corresponding $y_i = 1$. Similarly, anything below the decision boundary should have label −1 that is xisuch that $w^T x_i + b < 0$ will have corresponding $y_i = -1$. The decision function to $f(x) = sign(w^T x + b)$ since $f(x) = +1$ for all x above the boundary, and $f(x) = -1$ for all x below the boundary. The decision boundary is learned using the following quadratic programming:

$$\min_{\{w,b,\epsilon\}} \frac{w^T w}{2} + C \sum_i \epsilon_i$$

$$s.t. \quad y_i(w^T x_i + b) \geq 1 - \epsilon_i \quad and \quad \epsilon_i \geq 0 , \forall i$$

Mapping your data vectors, xi, into a higher-dimension (even infinite) feature space may make them linearly separable in that space (whereas they may not be linearly separable in the original space). The formulation of the quadratic programming problem is as above, but with all xireplaced with φ(xi), where φ provides the higher-dimensional mapping. So we have the standard SVM formulation:

$$\min_{\{w,b,\epsilon\}} \frac{w^T w}{2} + C \sum_i \epsilon_i$$

$$s.t. \quad y_i(w^T \phi(x_i) + b) \geq 1 - \epsilon_i \quad and \quad \epsilon_i \geq 0 , \forall i$$

Multiple Kernel Learning(MKL) was initially proposed by Lanckriet et. al. [8]. They introduced an Semi-Definite programming(SDP) approach to solve for the combination kernel. As SDP becomes intractable with increase in size and number of kernels, Bach et.al [11] reformulated MKL by considering each feature as a block and applying the l1 norm across the blocks and l2 norm within each block. For this formulation several algorithms[12] were proposed to speed up the optimization process. SimpleMKL [14] proposed by Rakotomamonjy et.al. derived a formulation which is equivalent to the block l1 norm based formulation and provided a Reduced Gradient Descent based algorithm that is faster than the SLIP algorithm pro- posed previously.

Let the kernel function $K(x_i,x_j) = < \phi(x_i), \phi(x_j) >$. The dual of the MKL formulation is given by,

$$\max_{\alpha\geq 0}[\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j K(x_i,x_j)$$

IV. NON-LINEAR KERNEL LEARNING USING GENETIC PROGRAMMING

As we note that in previous section all SVM based kernel learning is linear combination of kernels. This section describes the algorithm for learning non-linear combination of kernels using Genetic Programming. For aiding object categorization in adverse conditions, many descriptors (and instance of feature representation) have been proposed in the computer vision literature. Suppose there are n descriptors d1,d2,...,dnextracted from the m images I1,I2,...,Im. Using these n descriptors, n kernels K1,K2,...,Knof size $m \times m$ are formed. Steps involved for producing the hybrid classifier using GP and SVM is described below.

A. Algorithm for Finding Non-Linear Kernel Combination

- Create a random population of kernel functions, represented as trees
- Evaluate the fitness of each individual by building an SVM from the kernel tree and test it on the validation data
- Select the fitter kernel trees as parents for recombination
- Perform random crossover and mutation on the newly created offspring
- Replace the old population with the offspring
- Repeat Steps 2 to 5 until the population has converged
- Build final SVM using the fittest kernel tree found from GP

These are the parameters used for above algorithm. Terminal set = {K1,K2,K3,...,Kn}, where Ki is the kernel formed from any of the descriptors. Function set = {+,∗ }. Fitness function is the classification error of the particular chromosome on the training set. In other words fitness value for each chromosome in this GP will be based on the accuracy of SVM with that chromosome (the non-linear kernel combination given to SVM). One alternative is to base the fitness on a cross-validation test (e.g. leave-one-out cross-validation) in order to give a better estimation of a kernel trees ability to produce a model that generalizes well to unseen data.

V. RESULTS

Extensive experiments are conducted for evaluation of kernel learning methods of Genetic Programming and SVM MKL. The Experiments is validated using real-world image categorization datasets like Caltech-5 and Caltech-101. Caltech-5 contains five classes of objects cars, airplane, faces, leopards and bikes. Caltech-101 contains 101 categories of objects. Each category contains roughly from 30-100 images. Accuracy of the proposed method is compared to the best kernel (K1,K2,...,Kn) and addition kernel which is K1 + K2 + ... + Kn. All the experiments follow 1-Vs-1 SVM classification methods.

A. Results on Caltech-5 dataset

This section presents results on Caltech-51using new MKL formulation and descriptors (csift, opponentsift, rgsift, sift, transformedcolorsift) provided from ColorDescriptor software. The experiments in this section are carried out using descriptors available from Color Descriptor software2. Caltech-5 dataset contains images of airplanes, cars, faces, leopards and bikes. We have generated kernels on 5 descriptors provided using Gaussian kernel. This experimental procedure was repeated 10 times with different training-test data splits. It can be seen from Table 1 that the non-linear kernel method is giving better accuracy as compared to the SVM MKL, best kernel and the addition of kernels. Figure 1 shows plot of mean accuracy as number of iterations. Note that in all the iterations, proposed non-linear kernel is better than other kernel combinations. Figure 2 shows plot of mean accuracy as number of binary classifier on Caltech5, totally 10 binary classification problem. Note that non-linear kernel combination gives higher accuracy than other kernel combinations in all binary classifications. Figure 3 shows non-linear kernel tree generated from GP, which gives high accuracy than other kernel combination on binary classification problem 9 in previous graph. In other binary classification problem, GP ends in selecting best kernel. Figure 4 shows some nonlinear kernel tree generated from GP for Caltech5 dataset.

Kernel	Caltech5	Caltech101
Addition of Kernels	90.12±2.61	40.91±0.76
Best Kernel	91.00±2.04	36.40±0.89
SVMMKL	92.81±1.34	41.21±1.48
Non-Linear GP Kernel	94.76±1.71	42.71±1.48

Table I: Percentage accuracy obtained using different kernel combinations

B. *Results on Caltech-101 dataset*

This section presents results on Caltech-101 dataset using new non-linear kernel combination and six kernels are taken.

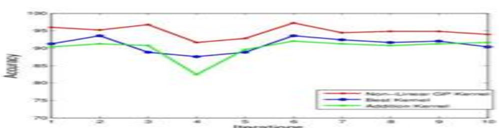
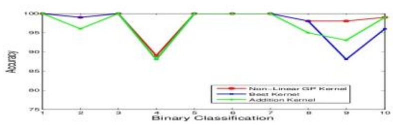


Figure 1: Shows plot of mean accuracy as number of iterations



[1] http://www.robots.ox.ac.uk/ vgg/data/data-cats.html
[2] http://staff.science.uva.nl/ ksande/research/colordescriptors/

Figure 2: Shows plot of mean accuracy as number of binary classifier on Caltech5

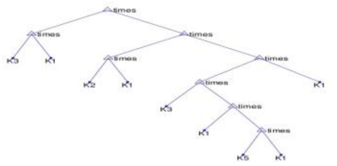


Figure 3: Shows non-linear kernel tree generated from GP for Caltech5 dataset

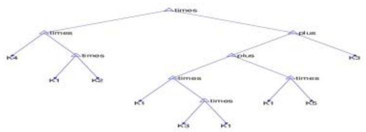


Figure 4: Shows non-linear kernel tree generated from GP for Caltech5 dataset.

We have taken 30 images for each class, of which 15 are randomly taken as the training in which 5 are taken for validation data and the remaining as test data. This experimental procedure was repeated 5 times with different training-test data splits. It can be seen from Table 1 that the non-linear kernel method is giving better accuracy as compared to the best kernel and the addition of kernels. Figure 5 shows plot of mean accuracy as number of iterations in Caltech 101 dataset. Note that in all the iterations proposed non-linear kernel is the best. Figure 6 shows non-linear kernel tree generated from GP for Caltech101 dataset. This kernel tree is nothing but the addition kernel, which is generated for iteration 4 in Caltech101 (see figure 5) where GP non-kernel and addition kernel give almost same accuracy.

VI. CONCLUSION

This paper methodologically comparesGenetic Programming non-linear kernel combinations with Support Vector Machine Multiple Kernel Learning on the application of image categorization. Genetic Programming approach eliminates the need for user to create non-linear kernel combination. Proposed framework is applied to Object Categorization. Experimental results show that Genetic Programming based framework for non-linear kernel combination performance better than existing state-of-the-art kernel combinations including Support Vector Machine Kernel Learning methods.

REFERENCES

[1] A. Kumar and C. Sminchisescu. Support kernel machines for object recognition. In ICCV, 2007.

[2] M.Varma and D. Ray. Learning the discriminative powerinvariance trade-off. In ICCV, pages 1–8, 2007.

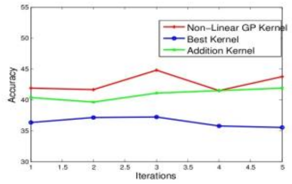


Figure 5: Shows plot of mean accuracy as number of iterations in Caltech 101 dataset

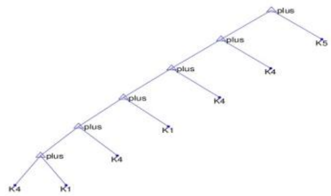


Figure 6: Shows non-linear kernel tree generated from GP for Caltech5 dataset

[3] Laura Diosan, MihaiOltean, Alexandrina Rogozan, and Jean Pierre Pecuchet. Genetically designed multiple-kernels for improving the svm performance. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1873–1873, New York, NY, USA, 2007. ACM.

[4] Keith M. Sullivan and Sean Luke. Evolving kernels for support vector machine classification. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1702–1707, New York, NY, USA, 2007. ACM.

[5] Tom Howley and Michael G. Madden. The genetic evolution of kernels for support vector machine classifiers. In *In 15th Irish Conference on Artificial Intelligence*, 2004.

[6] John Koza. Genetic programming: On the programming of computers by means of natural selection. MIT Press.

[7] Bernhard Schlkopf and Alexander J. Smola, Learning with Kernels Support Vector Machines, Regularization, Optimization, and Beyond, *MIT Press, 2002.*

[8] Lanckriet, G.R.G. and Cristianini, N. and Bartlett, P. and El Ghaoui, L. and Jordan, M.I., Learning the Ker- nel Matrix with Semidefinite Programming, *Journal of Machine Learning Research, vol 5, pages 27-72*, 2004.

[9] Damoulas, T. Girolami M, Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection, *Bioinformatics*, 2008.

[10] SakethaNathJagarlapudi, Dinesh Govindaraj, Raman S, Chiranjib Bhattacharyya, Aharon Ben-Tal, K. R. Ramakrishnan, On the Algorithmics and Applications of a Mixed-norm based Kernel Learning Formulation, *Proceedings of the Neural Information Processing Systems*, 2009.

[11] F. Bach and G. R. G. Lanckriet and M. I. Jordan, Multiple Kernel Learning, Conic Duality, and the SMO Algorithm, *International Conference on Machine Learning*, 2004.

[12] SorenSonnenburg and Gunnar Ratsch and Christin Schafer and Bernhard Scholkopf, Large Scale Multiple Kernel Learning, *Journal of Machine Learning Research*, vol 7, pages 1531-1565, 2006.

[13] Vladimir Vapnik, Statistical Learning Theory, Wiley-Interscience, 1998.
[14] A. Rakotomamonjy and F. Bach and S. Canu and Y Grandvalet, SimpleMKL, Journal of Machine Learn- ing Research, vol 9, pages 2491-2521, 2008.

AUTHOR BIBLIOGRAPHY

Prof. JyothiKorra received her master degrees in computer science from Indian Institute of Science, Bangalore. Her thesis is using Genetic Programming for the problem of kernel learning. Application of multiple kernel learning is applied for Object Categorization problem. She joined Christ University as Assistant Professor on 2011. She continues her research on Genetic Programming Applications in various areas.