Multiple Kernel Learning from Sets of Partially Matching Image Features

Siyao Fu * Zengguang Hou * Zize Liang * Qi Zuo ** Min Tan *

* Key laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences. P.O. Box 2728, Beijing 100190, China.(email:siyao.fu@ieee.org)
** Department of Automation, North China University of Technology, Beijing 100044, China.

Abstract: Kernel classifiers based on Support Vector Machines (SVM) have achieved state-of-the-art results in several visual classification tasks, however, recent publications and developments based on SVM have shown that using multiple kernels instead of a single one can enhance interpretability of the decision function and improve classifier performance, which motivates researchers to explore the use of homogeneous model obtained as linear combinations of kernels. Multiple Kernel Learning (MKL) allows the practitioner to get accurate classification results and identify relevant and meaningful features. However, the use of multiple kernels faces the challenge of choosing the kernel weights, and an increased number of parameters that may lead to overfitting. In this paper we show that MKL problem can be formulated as a convex optimization problem, which can be solved efficiently using projected gradient method. Weights on each kernel matrix (level) are included in the standard SVM empirical risk minimization problem with an $L_2$ constraint to encourage sparsity. We demonstrate our algorithm on classification tasks, including object recognition and classification, which is based on a linear combination of histogram intersection kernels, computed at multiple pyramid levels of image encoding, and we show that the proposed method is accurate and significantly more efficient than current approaches.

1. INTRODUCTION

Recent work in object recognition and image classification has shown that significant performance gains can be achieved by carefully combining multi-level, coarse-to-fine, layered feature encodings, and learning methods. Top-scoring classifiers on image databases like Caltech or Pascal tend to be discriminative and kernel-based, in particular, the support vector machines (SVM), a methodology well justified both theoretically and practically [4]: the resulting program is convex with global optimality guarantees, and efficient algorithms like Sequential Minimal Optimization (SMO) can be used to solve large scale problems, with hundreds of thousands of examples, which have proven to be efficient tools for solving learning problems.

In kernel methods, the data representation is implicitly chosen through the kernel function $K(x,x')$, the kernel actually defines the similarity between two examples $x$ and $x'$ with an appropriate regularization term for the problem. The solution of the learning problem can be formulated as follows:

$$f(x) = \sum_{i=1}^{l} \alpha_i^* y_i K(x,x_i) + b^*$$  \hspace{1cm} (1)

where $x_i, y_i$ are training sets, $l$ is the number of learning examples, $K(.,.)$ is a positive definite kernel function which implicitly maps examples to a feature space given by a feature map $\Phi$ via the identity $K(x_i,x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$, and $\alpha^*, b^*$ are coefficients need to be learned from examples. However it is often unclear what the most suitable kernel for the task at hand is, and hence the user may wish to combine several possible kernels. Recent applications (1,3) and developments based on kernel learning methods have shown that using multiple kernels instead of single one can enhance interpretability of the decision function and improve classifier performance. Within this framework one needs to specify the coefficients for SVM and the weights for the kernels in a single optimizing problem, which is known as the multiple kernel learning (MKL) problem. Thus, MKL is a way of optimizing kernel weights while training the SVM to get good classification accuracies.

The successful proliferation of MKL in computer vision has recently motivated several researchers to explore the use of homogenous models obtained as linear combinations of histogram intersection kernels, computed at multiple levels of image encoding [9, 10,11,12,23]. These more sophisticated classifiers have been demonstrated convincingly and state-of-the art results have been achieved, however, they face the challenge that the coefficients of these kernels need to be specified. One problem with simply adding kernels is that using uniform weights is apparently not optimal. Adding positive weight to an unrelated kernel means adding noise to the system. For histogram intersection based

* This work was supported in part by the National Technological Agency. (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase.
match problem, promising results have been obtained using geometric approximate weights to the optimal bipartite matching, but this problem dependent intuition may not be available for all kernels, or may not be as effective when combining heterogeneous kernels with different feature spaces, e.g. histogram intersections, polynomials and RBFs. What’s more, [2, 5] have shown that for some real world problems, the best performance is achieved only for a subset of the kernels/levels of encoding. Sometimes only a single one, which means MKL we consider is a problem of selecting a parsimonious kernel subset for a desired accuracy computation trade-off, a sparse kernel subset selection problem.

In this paper, we present an efficient approach for the multiple kernel learning problem, we first follow the framework proposed by Rakotomamonjy [2], we use a weighted 2-norm regularization of the function formulated by kernels, and we control the sparsity of the linear combination of enhanced spatial pyramid kernels. The regularization can be solved through by standard SVM implementation and a projected gradient method. The output is a set of convex and sparse classifiers which can be used for training multiple kernels by jointly optimizing both the coefficients of the classifiers and the weights of the kernels. The experiment results verify the effectiveness of our approach. The two major contributions of this paper are 1) A pyramid multiple kernel learning framework for object recognition. The model provides a novel solution to trade-off between the combination kernel weights specification and kernel selection. 2) a set of classifiers trained from these learned partitions which can provide very accurate category recognition performance with a very small amounts of labeled data.

2. MULTIPLE PYRAMID KERNEL LEARNING PROBLEM

Grauman and Darrel [9] have successfully used histogram intersection kernels as an indirect means to approximate the number of point correspondence between two feature sets (images), which is called Pyramid Match Kernel (PMK). They have demonstrated that PMK or SPK can achieve comparable results at a significantly lower computational cost than other state-of-the-art approaches by fusing the information from multiple resolutions according to the pyramid structure in the feature domain. However, the fixed weighting scheme for both PMK and SPK limits their applications. We consider the general problem of learning the weights automatically, thus, jointly optimizing the weights in a conic combination of kernel matrices and the coefficients of a discriminative classifier. This problem has been first proposed by Lanckriet [3], Bach [4] have reformulated the problem and proposed a SMO algorithm for solving the large scale problems. Another formulation of this problem is based on semi-definite linear problem which has been proposed by Sonnernburg [24], Rakotomamonjy [2] address the problem through an adaptive 2-norm regularization problem. De la Torre et al [16] learn the weights of a positive combination of normalized kernels using gradient descent. Bosch et al [17] learn the (level) weighting parameters of the spatial pyramid kernel using a validation set. The search for the optimal weighting is exhaustive, hence optimal, but may not scale as many kernels are added. For our experiments, we follow [2].

2.1 Problem Formulation

We consider the classification learning problem from \( n \) data input points \((x_i, y_i)\), where \( x_i \) belongs to some input space \( X \), and \( y_i = \pm 1 \) denoting the class label of examples \( x_i \), the multiple kernel learning problem is formulated as follows:

\[
\min_{\alpha} \sum_{i=1}^{n} L(\alpha y_i \Phi(x_i)) + \lambda \sum_{j=1}^{m} d_j ||\omega_j||^2 \tag{2}
\]

where \( \Phi(x) \) is the feature map function, and \( \omega \) are coefficients for the classifier and \( d_j \) denotes the weights (positive) associated with each kernel. \( \lambda \) is a regularization constant, and \( L \) is the hinge loss function. The goal is to find a linear classifier of the form \( y = sign(\omega^T + b) \), Bach [4] used Moreau-Yosida regularization method to get the primal problem:

\[
\min_{\alpha} \frac{1}{2} \left( \sum_{j=1}^{m} d_j ||\omega_j||^2 \right)^2 + \frac{1}{2} \sum_{j=1}^{m} \alpha_j^2 ||\omega_j||^2 + C \sum_{i=1}^{n} \xi_i \quad \text{s.t.} \quad y_i \left( \sum_{j=1}^{m} \alpha_j \Phi_j(x_i) + b \right) \geq 1 - \xi_i, \forall i \in 1, ..., n
\]

where \( \xi \) is slack variables for soft margin.

2.2 Solving the Problem

As typical with SVMs a first step for solving the problem would be finding the derivatives for its Lagrangian. The saddle point and stationary conditions of the gradient, as we can see, there are 500 points in the
Algorithm 1 The adaptive 2-norm multiple kernel learning algorithm

\textbf{Input:} Gram matrix $K_0$, constraint matrices $A_i$
\[ k = 1 \]
\[ \beta_k^1 = \frac{1}{n} \text{ for } k = 1, ... , M \]
define map $P = \arg \min \{ ||z - \beta||, z \in \mathcal{X} \}$
\textbf{repeat}
\hspace{1em} for $t = 1, 2, ...$ do
\hspace{2em} Solve the classical SVM with $K = \sum_k \beta_k^t K_k$
\hspace{2em} Compute the gradient $\frac{\partial J}{\partial \beta_k}$ using standard method, for $k = 1, 2, ..., M$
\hspace{2em} $\beta_k^{t+1} = \beta_k^t + \tau D_{t,k}$
\hspace{2em} Map to $X$ we obtain $\beta_k^{t+1} = P(\beta_k^t + \tau D_{t,k})$ for map $P$
\hspace{2em} Dogleg search algorithm to find the step size $\tau$ for the direction $D_{t,k}$.
\hspace{2em} if stopping criterion then break
\hspace{1em} end if
\textbf{end for}
\textbf{until} $||\beta_k|| < \epsilon$

chess dataset. we use a combination of 10 Gaussian kernels, while their weights remain unfixed, for simplicity, we set the sum of their weights to 5, we need to learn the weights for the kernels while doing the classification problem. The final results show that our framework performs very well. The learned weights for the kernels are as follows: $\omega_1 = \{2.1867, 0.4198, 0.12600, 1.3335, 0.0000, 0.0000\}$, which infers that the first, second, fourth and fifth Gaussian kernel are active, while the rest remains unused. This allow us to trace the active kernels during the classification procedure based on which specific kernel are responsible for causing a match at a particular level.

3. PYRAMID SPATIAL MATCH KERNELS

This section briefly describes pyramid match kernels which inspired our study for automatically learning sparse combinations of kernels, and presents a enhancement for the spatial pyramid kernel. Grauman & Darrel [9] and Lazebnik et al. [23] have successfully used histogram intersection kernels as an indirect means to approximate the number of point correspondences between two sets of features. Fig. 1 illustrates the principle of PMK. The basic idea of PMK is embedding feature sets to a multi-resolution histogram that preserves the individual feature’s distinctiveness at the finest level. The histogram pyramids are then compared using a weighted histogram intersection computation scheme. The similarity between the two sets is then defined by the weighted sum of histogram intersection at each level, which can be written as:

\[ K_{\Delta}(\Psi(X), \Psi(Y)) = \sum_{k=0}^{L} \omega_k (\Gamma_k(X_i, Y_i) - \Gamma_{k-1}(X_i, Y_i)) \]
\[ \Gamma_k(X_i, Y_i) = \sum_{b_k} \min(H_k(X_i^n), H_k(Y_i^n)) \]  \hspace{1em} (6)

Where $X, Y$ are feature sets; (2), $H_k(X_i^n)$ is the count in bin $n_i$, $\Gamma$ is a histogram intersection over the point set $X$ having $b_k$ multi-dimensional bins with sides of length $2^k$, $L$ is the number of total resolution levels in the pyramids, and $\omega_k$ reflects the similarity between points matched at level $k$, which depends on the coarseness of the grid: histogram intersections $\Gamma$ at a coarser grid level are overly penalized compared to intersections on finer levels, this is because matches found in larger bins should be penalized more aggressively as they contain increasingly dissimilar features. Grauman and Darrel set $\omega_k = \frac{1}{k}$, which corresponds to a similarity weighting inversely proportional to the bin size; (3), $K_{\Delta}$ is the kernel function formed by the sum of matches. Histogram intersection functions obtained in this way are positive-definite similarity functions, hence kernels. See [9] for details.

PMK works effectively when dealing with large data sets; however, PMK discards all information about the spatial layout of the features, so it is hard to decide the proper match if the surface of the object is not apparent enough to offer discriminative features. Both [6] and [23] suggest a different enhancement instead, which contains spatial information for the objects and can converge faster than the original PMK, the difference between the kernels used in [6] and [23] comes in the way features are mapped to histograms. In [23] the grid is defined over the image, divided into $2^l \times 2^l$ bins with features from spatially corresponding bins matched across any two images, however, one major problem for this definition is that some sparse distributed features will not converge at the same speed because the weights associated with these levels are very small, so points will converge faster there, but it will take longer time to converge at dense part. In this sense, the one major problem for this definition is that some sparse distributed features will not converge at the same speed because the weights associated with these levels are very small, so points will converge faster there, but it will take longer time to converge at dense part. In this sense, it is the dense resolution part of the set that degrades the matching results and slows down the process. One solution we proposed in [6] is that the grid will be divided according to the feature’s distribution, thus, we try to find the trade-off between the computation efficiency and matching score, making the matching speed stay in same for different dimensions. We summarize the enhancement for SPK as follows: first, the feature sets are mapped into feature space where their intersection of histogram are calculated by original SPK, then we set the number of

![Fig. 1. A pyramid match determines a partial correspondence by matching points once they fall into the same histogram bin. A multi-resolution histograms are shown, the similarity between the two sets $Y$ and $Z$ is defined by the weighted sum of histogram intersections at each level. The weights $w_i$ are proportional to the resolution of each level, thus, $1, \frac{1}{2}, \frac{1}{4}$, See text for details. (This figure is best viewed in color.)](image-url)
number of test images to 40 per class, thus, the counterweights, suspension clamps, partition sticks, insulators, jumpers and conductors, respectively. We follow the experimental setup of Grauman and Darrell [9]. For experiments described in this paper, we detected all the salient points with Harris-Affine interest operator [6], then the image is decomposed into sets of SIFT features. We use PCA to reduce the dimensionality of the SIFT descriptors to 10.

For the dataset, we select 25 images for training while the remaining images in each class are used for testing. The values between a test image and the training examples are computed using SVM. After being compared with all classifiers, the test image is labeled as the category with the highest match correspondence. For the kernel selection we use three types of kernels, the original PMK, SPK and its enhancement: E-SPK, for PMK, we set four levels in the pyramid structure with 8, 4, 2 and 1 bins per dimension respectively, as described above, we know the first level containing 8 bins is the finest while the fourth level is the coarsest (containing only 1 bin). For original SPK we have four levels with grid size 8 × 8, 4 × 4, 2 × 2, 1 × 1, respectively. For our E-SPK, we have set four levels with grid size 4 × 4, 3 × 3, 2 × 2, 1 × 1.

While running our algorithm, a coefficient \( \beta \) smaller than 0.001 is forced to 0. All the experiments have been run on a Pentium D-3 GHz and 2GB of RAM. For a fair comparison, we have selected the same termination criterion for these iterative algorithms: iteration terminates when \( \| \beta_{i+1} - \beta_{i} \| < 0.01 \) or the maximal number of iteration (500) has been reached. Training sets have been normalized to zero mean and unit variance and the test sets have also been rescaled accordingly. Different values of the hyperparameter \( C \) (0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100) have been tried. For the problems solved here, we need to consider all kinds of kernels from all levels and take their weights into account, which means we need to specify which kernel has been activated during the classification procedure. The pattern of sparsity is shown in figure. 5 with binary coded diagrams. The number on the vertical states the one-vs-all problems being trained, while the number on the horizontal means the kernel series from the left to the right, corresponding to the fine to coarse levels in the pyramid. (The first one to four stands for PMK, the second for SPK, while the third is E-SPK). Each black rectangular bar corresponds to the kernel has a non-zero weight, while white means it is turned off, without taking part into the computation, thus, the diagrams show which kernels are active during the procedure. Notice that the highest weighted kernel is usually the finest level, this is consistent with the fact that the most accurate match is always occurred in the finest match level, which in turn contribute most to the whole match procedure. An interesting observation is that the active kernels at the coarser levels (3 and 4) are significantly less than the first two levels. One possible explanation is that the coarser kernels can no longer provide sufficient discriminative power should not be included, thus, they are assigned a zero weight and be considered as non-informative for classification.

We also compare the mean recognition rate by using the support kernel machines with those original prefixed weighting histogram intersection kernels. We set their standard configurations and define the results obtained

4. EXPERIMENTS

In this section we evaluate our framework with a test data set generated on our inspection robot, which contains 500 images with six object classes. For efficiency, we limit the

bins corresponding to the sparsity of the features obtained above, for the dense distributed part, we set more bins there, while for the sparred region, the number of bins will be smaller, when a bin is being resized, the scale factor \( s_j \) (instead of 2) is applied to the \( j_{th} \) side of the bin. Note that the scale factors are computed based on the reference object. The same scale factors are applied to the query object when it is compared to the reference one. The final kernel is then the sum of the separate channel kernels. For details, please refer to [6].
Fig. 3. The performance for the multiple kernel learning framework, left part is the set containing 500 points, right part is the classification results, notice that the gray stripe is the classification interface. See text for details.

Fig. 4. The pattern sparsity for all three types of kernels, while the entire 12 kernel set corresponding to PMK, SPK, E-SPK is considered. Black and white plots show the activity for the kernels. Black means the corresponding kernel has a non-zero weight, while white means it is not activated. See text for details.

by these methods as baseline results. The weights of each level for PMK is $\frac{1}{2}$, from the finest to coarsest level, and the same for SPK, for E-SPK it turns out to be 0.5, 0.3, 0.25, 0.125. We repeat the experiment with various number of training images per class (10, 20, 25, 30), figure. 5 shows mean classification results for different methods, the learned kernel combination outperforms each of PMK, SPK, and E-SPK alone.

Another experiment shows the comparison for the specific classification score with other representative measures including original PMK (LibPMK), and SPK based kernels. All experiments are repeated five times with different randomly selected training and test images. Recognition performance is measured as the mean and standard deviation of the results from the individual runs. Semi-supervised constraints are of the “must group” form between pairs of unlabeled examples, and an equal number of such constraints is randomly selected for each class from among the training pool. Table 1 shows the classification results with our method and the comparison with other methods, we notice that some classes such as strings and clamps on which MKL achieved high performance and outperform the other methods, while partition sticks and jumpers on which our method performed poorly due to the reason that partition sticks and jumpers are textureless obstacles, making neither PMK nor SPK could offer correct feature information for recognition, thus, their combination couldn’t give the more satisfactory results. However, the results still show a significant boost in the classification performance given by our method than the traditional approaches in general. [9]

Table 1. Classification results for the obstacle database with our method and comparison with two other methods.

<table>
<thead>
<tr>
<th>Recognition rate</th>
<th>LibPMK</th>
<th>SPK</th>
<th>E-SPK</th>
<th>MKL</th>
</tr>
</thead>
<tbody>
<tr>
<td>suspension clamps</td>
<td>74.3%</td>
<td>81.3%</td>
<td>85.7%</td>
<td>87.1%</td>
</tr>
<tr>
<td>insulator strings</td>
<td>85.5%</td>
<td>89.6%</td>
<td>91.5%</td>
<td>93.4%</td>
</tr>
<tr>
<td>conductors</td>
<td>73.2%</td>
<td>84.2%</td>
<td>88.4%</td>
<td>88%</td>
</tr>
<tr>
<td>counterweights</td>
<td>88.3%</td>
<td>85%</td>
<td>84%</td>
<td>84.4%</td>
</tr>
<tr>
<td>jumpers</td>
<td>72.4%</td>
<td>79%</td>
<td>77%</td>
<td>72.5%</td>
</tr>
<tr>
<td>partition sticks</td>
<td>43.3%</td>
<td>63.5%</td>
<td>68.1%</td>
<td>64.7%</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS AND FUTURE WORK

Kernel machines (e.g. SVM, KLDA) have shown state-of-the-art performance in several visual classification tasks. The classification performance of kernel machines greatly depends on the choice of kernels and its parameters. In this paper, we introduce a novel histogram intersection kernel and applied it to the problem of multiple kernel learning framework in support vector machines. The kernel is based on spatial pyramid match structure. We propose a method to search over a space of parameterized kernels using a projected gradient-descent based method. Our method effectively learns a non-linear representation of the data useful for classification by estimating both the parameters of a sparse linear combination of kernels, and a discriminative classifier in one convex problem. We demonstrate our algorithm on classification tasks, including object recognition and classification, in both synthetic and real examples in images. We are also looking for other possible kernel parameterizations that can extract interesting image features in a supervised or unsupervised manner. Future works aims at improving both speed and
sparsity in kernels of the algorithm and extending the framework to other field.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (Grants 60635010 and 60775043). The authors also wish to acknowledge the help and suggestions from Gongping Wu of the College of Power and Mechanics Engineering, Wuhan University.

REFERENCES


