Remote sensing image classification exploiting multiple kernel learning
Claudio Cusano, Paolo Napoletano, Raimondo Schettini

Abstract—We propose a strategy for land use classification which exploits Multiple Kernel Learning (MKL) to automatically determine a suitable combination of a set of features without requiring any heuristic knowledge about the classification task. Standard MKL requires many training examples, otherwise the learned classifier fails to generalize to unseen data. In this paper we present a novel procedure that allows MKL to achieve good performance in the case of small training sets as well. Experimental results on publicly available datasets demonstrate the feasibility of the proposed approach.

Index Terms—Remote sensing image classification, multiple kernel learning.

I. INTRODUCTION

The automatic classification of land use is of great importance for several applications in agriculture, geology, forestry, weather forecasting, and urban planning. Land use classification consists in assigning semantic labels (such as urban, forest, sea etc.) to aerial or satellite images. In the past this problem has been often addressed by exploiting spectral analysis techniques to independently assign the labels to each pixel in the image [1]. Recently, several researchers experimented with pattern recognition techniques consisting, instead, in extracting image features and in classifying them according to models obtained by supervised learning. Multiple features should be considered because the elements in the scene may appear at different scales and orientations and, due to variable weather and time of the day, also under different lighting conditions. The features are usually related to color, shape, and texture, posing the problem of finding a suitable scheme for their combination.

Sheng et al. designed a two-stage classifier that combines a set of complementary features [2]. In the first stage Support Vector Machines (SVM) are used to generate separate probability images using color histograms, local ternary pattern histogram Fourier (LTP-HF) features, and some features derived from the Scale Invariant Feature Transform (SIFT). To obtain the final result, these probability images are fused by the second stage of the classifier. Risiojević et al. applied non-linear SVMs with a kernel function especially designed to be used with Gabor and Gist descriptors [3]. In a subsequent work, Risiojević and Babić considered two features: the Enhanced Gabor Texture Descriptor (a global feature based on cross-correlations between subbands) and a local descriptor based on the Scale Invariant Feature Transform (SIFT). They identified the classes of images best suited for the two descriptors, and used this information to design a hierarchical approach for the final fusion [4]. Shao et al. [5] employed both non-linear SVMs and L1-regularized logistic regression in different classification stages to combine SIFT descriptors, shape feature as proposed by Xia et al. for image indexing [6], texture feature based on Local Binary Patterns (LBP) and a bag-of-colors signature.

In this paper, we propose a strategy for land use classification which exploits Multiple Kernel Learning (MKL) to automatically combine a set of features without requiring any heuristic knowledge about the classification task. One of the drawbacks of MKL is that it requires a large training set to select the features and to train the classifier simultaneously. In order to apply MKL to small training sets as well, we also introduce a novel automatic procedure that produces candidate subsets of the available features before solving the optimization problem defined by the MKL framework.

The proposed strategy exploits both features commonly used in scene classification as well as new features specially designed by the authors to cope with the land use classification problem.

We evaluated our proposal on two publicly available data sets: the Land use data set of aerial images [7], and a data set of satellite images obtained from the Google Earth service [8]. We compared our framework with others from the state of the art. The results show that the proposed framework performs visibly better than other methods when the training set is small.

II. PROPOSED CLASSIFICATION SCHEME

Multiple Kernel Learning (MKL) is a powerful machine learning tool which allows, in the framework of Support Vector Machines (SVM), to automatically obtain the suitable combinations of features as well as the suitable mixture of kernels for each feature [9]. We evaluated the MKL framework as proposed by Sonnenburg et al. [10] in the case of small size of the training set (results are detailed in the experimental section). We observed that dense mixtures of kernels can fit real data better than sparse mixtures but also that both sparse and non sparse solutions do not outperform other trivial baselines solutions such as those proposed by Risiojević and et al. [4].

To improve the results in the case of small training sets, we introduce a novel heuristic procedure that automatically selects candidate subsets of the available features and kernels before solving the MKL optimization problem.
A. Multiple Kernel Learning

Support Vector Machines exploit the ‘kernel trick’ to build non-linear binary classifiers. Given a feature vector \( x \), the predicted class \( y \) (either \(-1\) or \(+1\)) depends on a kernel function \( k \):

\[
y = \text{sign}(b + \sum_{i=1}^{N} \alpha_i y_i k(x_i, x))
\]

where \((x_i, y_i)\) are the features/label pairs forming the training set. The parameters \( b \) and \( \alpha_i \) are determined during the training procedure, which consists in solving the following quadratic optimization problem:

\[
\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_{i=1}^{N} \alpha_i,
\]

under the constraints \( \sum_i \alpha_i y_i = 0 \) and \( 0 \leq \alpha_i \leq C \) (for a set value of the penalization parameter C). Beside the linear kernel \( k(x_1, x_2) = x_1^T \cdot x_2 \), other popular kernels are the Gaussian RBF \( k(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|^2)\) and the \( \chi^2 \) kernel \( k(x_1, x_2) = \exp(-\gamma \sum_{i,j} \frac{(x_{1i} - x_{2j})^2}{x_{1i} + x_{2j}}) \) both depending on an additional parameter \( \gamma \).

The choice of the kernel function is very important since it implicitly defines the metric properties of the feature space. Multiple kernel learning (MKL) is an extension of Support Vector Machines (SVMs) that combines several kernels. It represents an appealing strategy when dealing with multiple features \( x = (x^{(1)}, x^{(2)}, \ldots, x^{(F)}) \) and with multiple kernel functions \( k^{(1)}_f, k^{(2)}_f, \ldots, k^{(M)}_f \) for each feature. In MKL, the kernel function used in (1) is a linear combination of these \( F \times M \) kernels

\[
k_{\text{comb}}(x, x_i) = \sum_{f=1}^{F} \sum_{j=1}^{M} \beta^{(f)}_j k^{(f)}(x^{(f)}, x^{(f)}_j).
\]

The weights \( \beta^{(f)}_j \geq 0 \) are considered as additional parameters to be found when solving the optimization problem (2) and with the inclusion of the new constraint \( \sum_j \beta^{(f)}_j = 1 \).

The first approaches to MKL aimed at finding sparse mixtures of kernels with the introduction of \( l_1 \) regularization for the weights of the combination. For instance, Lanckriet et al. induced the sparsity by imposing an upper bound to the trace of \( k \). The resulting optimization was computationally too expensive for large scale problems. Moreover, the use of sparse combinations of kernels rarely demonstrated to outperform trivial baselines in practical applications.

Sonnenburg et al. introduced an efficient strategy for solving \( l_p \)-regularized MKL with arbitrary norms, \( p \geq 1 \). The algorithm they proposed allows to estimate optimal weights and SVM parameters simultaneously by iterating training steps of a standard SVM. This strategy allows both sparse (\( p = 1 \)) and non-sparse (\( p > 1 \)) solutions.

Compared to SVMs, MKL struggles to deal with small training sets. This fact mostly depends on the increased number of parameters that have to be set during training, and is particularly evident when there are many features and kernels.

B. Heuristic MKL

We introduce, here, a novel heuristic approach to MKL that finds sparse mixtures of kernels without imposing the sparsity condition (\( p = 1 \)). Using a small number of kernels is very important in the case of small training sets because it reduces the number of parameters to be learned and, consequently, limits the risk of overfitting the training data. In fact, using all the available kernels is usually worse than using a small selection of good kernels. Sparsity could be enforced by constraining to be zero a subset of the coefficients of the kernels before solving the \( l_2 \) regularized optimization problem. The optimal solution could be found by considering all the \( 2^F \times M \) possible subsets. However, this approach would easily result intractable even for relatively small numbers of features \( F \) and kernels \( M \). A tractable greedy solution would consist in selecting one kernel at a time on the basis of their individual merits. This approach, instead, would fail to capture most of the interactions among the kernels.

Our heuristic strategy deals with the limitations of the greedy algorithm, without resulting intractable as the exhaustive search. Briefly, it consists in the automatic selection of a small number of kernels (one for each feature) in an iterative augmentation of such an initial selection. New kernels are not included only because of their performance in isolation (measured on a given training set), but they are chosen by taking into account how much they complement those that have been already selected. Since complementarity is more easily found across different features, at each iteration the algorithm considers for the inclusion at most one kernel for each feature. More in detail, the procedure is composed of four major steps. For each step the goodness of a set of kernels is evaluated by training the MKL with \( p = 2 \) and by a five-fold cross validation on the training set. The steps are:

1) for each of the \( F \) features and for each of the \( M \) kernels, a classifier is trained and evaluated; the best kernel for each feature is then included in the initial selection;
2) the inclusion of each non-selected kernel is individually evaluated by temporarily adding it to the current selection;
3) a set of candidate kernels is formed by choosing the most improving kernel for each feature, as determined in step 2 (if for a given feature, no kernel improves the accuracy that feature will remain without a candidate);
4) from the set of candidates all the subsets of kernels are temporarily joined to the current selection, and the best subset is permanently added to the selection.

The steps 2–4 are repeated until the set of candidates found in step 3 is empty (this would eventually happen since each step adds at least one kernel until no kernel improves the accuracy, or until all the kernels have been selected). Figure 1 depicts a graphical representation of the whole procedure.

Step 4 requires the evaluation of up to \( 2^F - 1 \) combinations of candidates, and that step is repeated up to \( M \times F \) times (since at least one kernel is added at each iteration). Therefore, in the worst case the number of trained classifiers is \( O(F \times M \times 2^F) \). Such a number can be kept manageable if the number of features \( F \) is reasonable. As an example, in the
class dataset described in the next section, the training with combinations of kernels. To have an idea, on 10% of the 19-
of the art on publicly available data sets for the evaluation of land use classification. The evaluation includes:

- several order of magnitude less than the brute force solution that would require the evaluation of up to $2^{4\times9} = 6.87 \times 10^{10}$ combinations of kernels. To have an idea, on 10% of the 19-class dataset described in the next section, the training with our strategy set required about 45 minutes on a standard PC.

**III. EXPERIMENTAL EVALUATION**

To assess the merits of the classifier designed according to our proposal, we compared it with other approaches in the state of the art on publicly available data sets for the evaluation of land use classification. The evaluation includes:

- image features: we evaluated several image features and their concatenation using SVMs with different kernels (linear, RBF and $\chi^2$);
- MKL: we evaluated three versions of MKL ($p = 1$, $p = 2$ and the proposed strategy);
- other approaches in the state of the art: we evaluated the method proposed by Risojević et al. by using the code provided by the authors [4].

For all the experiments, the “one versus all” strategy is used to deal with multiple classes.

**A. Data**

We considered two different datasets.

1) **21-Class Land-Use Dataset**: this is a dataset of images of 21 land-use classes selected from aerial orthoimagery with a pixel resolution of one foot [17]. The images were downloaded from the United States Geological Survey (USGS) National Map of some US regions [10]. For each class, 100 images at $256 \times 256$ are available. These classes contain a variety of spatial patterns, some homogeneous with respect to texture, some homogeneous with respect to color, others not homogeneous at all. An example of each class is shown in Fig. 2.

2) **19-Class Satellite Scene**: this dataset[5] consists of 19 classes of satellite scenes collected from Google Earth (Google Inc.). Each class has about 50 images, with the size of $600 \times 600$ pixels [8, 6]. The images of this dataset are extracted from very large satellite images on Google Earth, where the illumination, appearances of objects and their locations vary significantly, with frequent occlusions. An example of each class is shown in Fig. 3.

**B. Image features**

In our experiments we considered four image features: two of them have been taken from the state of the art and have been chosen because of the good performance that are reported in the literature. The other two features have been specially designed, here, to complement the others.

1) **Bag of SIFT**: in this work we considered SIFT [12] descriptors quantized into a codebook of 1096 “visual words”. This codebook has been previously built by clustering the descriptors extracted from more than 30,000 images. To avoid unwanted correlations with the images used for the evaluation, we built the codebook by using images downloaded from the flickr web service. The final feature vector is a normalized histogram of the occurrences of the 1096 visual words.

2) **Gist**: these are texture features computed from a wavelet image decomposition [13]. Each image location is represented by the output of filters tuned to different orientations and scales. The resulting representation is then downsampled to $4 \times 4$ pixels. We used eight orientations and four scales thus, the dimensionality of the feature vector is $8 \times 4 \times 16 = 512$.

---

Fig. 1. Scheme of the four main steps that form the proposed heuristic for the selection of the kernels.

Fig. 2. Examples from the 21-Class Land-Use dataset. From the top left to bottom right: agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersections, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis courts.

Fig. 3. Examples from the 19-Class Satellite Scene dataset. From the top left to bottom right: airport, beach, bridge, commercial area, desert, farmland, football field, forest, industrial area, meadow, mountain, park, parking, pond, port, railway station, residential area, river and viaduct.
3) Bag of dense LBP: this feature has been designed to cope with the problem that images of the same category contain spatial variations of the content: same parts and objects of the scene occur in different positions. Local binary patterns are calculated on square patches of size $w \times w$ that are extracted as a dense grid from the original image. The final descriptor is obtained as a bag of such LBP patches obtained from a previously calculated dictionary. As for the SIFT, the codebook has been calculated from a set of thousands of generic scene images. Note that, this descriptor has been calculated on each channel and then concatenated. We have chosen the LBP with a circular neighbourhood of radius 2 and 16 elements. We used LBP with 54 uniform and rotation invariant patterns.

4) LBP of dense moments: this is a brand new feature specially designed to cope with the problem that images of the same category contain spatial variations of the same content. The original image is divided in a dense grid of $N$ square patches of size $w \times w$. The mean and the standard deviation of each patch and of each color channel is calculated thus obtaining two matrices of real values per channel. Finally, the method computes the LBP of each matrix and the final descriptor is then obtained by concatenating the two resulting LBP histograms of each channel. We have chosen the LBP with a circular neighbourhood of radius 2 and 16 elements. We used LBP with 54 uniform and rotation invariant patterns. We set $w = 16$ and $w = 30$ for the 21-classes and 19-classes respectively. The final dimensionality of the feature vector is $54 + 54 = 108$.

C. Experimental setup

For each dataset we used training sets of different sizes. More in detail, we used 10, 50, 80 and 90% of the images for training the methods, and the rest for their evaluation. To make the results as robust as possible, we repeated the experiments 100 times with different random partitions of the data (available on the authors’ web page).

In the experiments for single kernel SVMs we used the parallel version of the library LIBSVM\cite{CC2001} proposed by Li et al. \cite{Li2007}. For MKL we used the algorithm proposed by Sonnenburg et al. \cite{SSSG03,SSSG04} that is part of the SHOGUN machine learning toolbox\cite{Sonnenburg2006}, developed by Sonnenburg et al. For both single and multiple kernel experiments, we considered the linear, Gaussian RBF and $\chi^2$ kernel functions. In the case of

### Table I

**Performance comparisons on the 21-class dataset.**

| Features   | Kernel | 10%   | 50%   | 80%   | 90%   |
|------------|--------|-------|-------|-------|-------|
| Bag of SIFT| $\chi^2$| 60.19 | 74.72 | 77.82 | 78.43 |
| Bag of LBP | $\chi^2$| 52.41 | 74.09 | 78.36 | 79.77 |
| GIST       | $\chi^2$| 50.88 | 70.09 | 74.87 | 75.68 |
| LBP of moments | RBF | 33.87 | 44.34 | 47.21 | 47.32 |
| Concatenation | RBF | 66.75 | 81.58 | 84.80 | 85.30 |
| Metalearner \cite{Güney2008} | Ridge regression C | 70.30 | 88.45 | 91.35 | 91.98 |
| MKL ($p = 1$) \cite{Sonnenburg2006} | Linear, RBF, $\chi^2$ | 64.14 | 87.54 | 91.11 | 91.21 |
| MKL ($p = 2$) \cite{Sonnenburg2006} | Linear, RBF, $\chi^2$ | 70.56 | 88.62 | 90.38 | 90.52 |
| Proposed MKL | Linear, RBF, $\chi^2$ | **74.13** | **88.68** | **91.26** | **91.86** |

### Table II

**Performance comparisons on the 19-class dataset.**

| Features   | Kernel | 10%   | 50%   | 80%   | 90%   |
|------------|--------|-------|-------|-------|-------|
| Bag of SIFT| $\chi^2$| 64.22 | 83.29 | 87.07 | 87.17 |
| Bag of LBP | $\chi^2$| 55.22 | 78.09 | 83.00 | 83.84 |
| GIST       | $\chi^2$| 48.91 | 65.99 | 69.39 | 70.65 |
| LBP of moments | RBF | 43.07 | 57.90 | 60.76 | 60.93 |
| Concatenation | RBF | 80.00 | 94.00 | 95.78 | 96.56 |
| Metalearner \cite{Güney2008} | Ridge regression C | 77.03 | 92.56 | 94.93 | 95.57 |
| MKL ($p = 1$) \cite{Sonnenburg2006} | Linear, RBF, $\chi^2$ | 63.75 | 92.13 | 95.61 | 96.12 |
| MKL ($p = 2$) \cite{Sonnenburg2006} | Linear, RBF, $\chi^2$ | 76.70 | 93.19 | 95.21 | 95.99 |
| Proposed MKL | Linear, RBF, $\chi^2$ | **82.30** | **95.30** | **96.74** | **97.37** |
single SVM model selection procedure; for MKL we used the values $\gamma = \{10, 1, 0.1, 0.01\}$ and $\gamma = \{3, 2, 1, 0.5\}$ for the Gaussian RBF and $\chi^2$ kernels respectively.

### D. Results

Table I and Table II report the results achieved on the 21-class and 19-class dataset, respectively. Among single features, bag of SIFT obtained the highest accuracy, with the exception of the case of a large training set for the 21-class dataset where the best feature is the bag of LBP. Regardless the size of the training set, the simple concatenation of the four features significantly improved the results. In fact, features such as the LBP of moments that resulted weak when used alone, significantly improved the results. In fact, features such as the metalearners proposed in [10], performed better than simple concatenations. However, such as the metalearners proposed in [4] or the MKL as defined in [13], performed better than simple concatenations. However, it may be successful in those cases where few training data are available and, at the same time, multiple heterogeneous features are needed.

![Hinton’s diagram](image)

**Fig. 4.** Hinton’s diagram of the weights selected by the heuristic MKL. Numbers at the bottom indicate kernels: 1 for linear, 2–5 for RBFs at different scales, orientations and lighting conditions. Therefore, multiple features must be combined in order to obtain good performance. To do so, in this paper we presented a classification strategy based on MKL. Our strategy improves MKL by making it suitable for small training sets. Experimental results on two public land use datasets shown that our method performs better than the other alternatives considered. We believe that our approach could be applied to other image classification tasks. In particular, we expect that it may be successful in those cases where few training data are available and, at the same time, multiple heterogeneous features are needed.

### ACKNOWLEDGMENT

We would like to thank Vladimir Risojević and Zdenka Babić for making available the code of their methods used in our comparison, and for their valuable comments.

### REFERENCES

[1] J. Campbell, *Introduction to remote sensing*. CRC Press, 2002.
[2] G. Sheng, W. Yang, T. Xu, and H. Sun, “High-resolution satellite scene classification using a sparse coding based multiple feature combination,” *Int’l J. of Remote Sensing*, vol. 33, no. 8, pp. 2395–2412, 2012.
[3] V. Risojević, S. Momić, and Z. Babić, “Gabor descriptors for aerial image classification,” in *Adaptive and Natural Computing Algorithms*, 2011, pp. 51–60.
[4] V. Risojevic and Z. Babic, “Fusion of global and local descriptors for remote sensing image classification,” *Geoscience and Remote Sensing Letters*, vol. 10, no. 4, pp. 836–840, 2013.
[5] W. Shao, W. Yang, G.-S. Xia, and G. Liu, “A hierarchical scheme of multiple feature fusion for high-resolution satellite scene categorization,” in *Computer Vision Systems*. Springer, 2013, pp. 324–333.
[6] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, H. Maitre et al., “Structural high-resolution satellite image indexing,” in *ISPRS TC VII Symposium-100 Years ISPRS*, vol. 38, 2010, pp. 298–303.
[7] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *Proc. of the Int’l Conf. on Advances in Geographic Information Systems*. ACM, 2010, pp. 270–279.
[8] D. Dai and W. Yang, “Satellite image classification via two-layer sparse coding with biased image representation,” *Geoscience and Remote Sensing Letters*, vol. 8, no. 1, pp. 173–176, 2011.
[9] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien, “Efficient and accurate lp-norm multiple kernel learning,” *Advances in neural information processing systems*, vol. 22, no. 22, pp. 997–1005, 2009.
[10] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, “Lp-norm multiple kernel learning,” *J. Mach. Learn. Res.*, vol. 12, pp. 953–997, 2011.
[11] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, “Learning the kernel matrix with semidefinite programming,” *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, Dec. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1005532.1005534
[12] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int’l J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[13] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int’l J. Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.
[14] F. Li, J. Carreira, and C. Sminchisescu, “Object recognition as ranking holistic figure-ground hypotheses,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.