A Multiple Kernel Learning Algorithm for Cell Nucleus Classification of Renal Cell Carcinoma

Peter Schüffler[†1] [⋆], Aydın Ulaş[†2], Umberto Castellani[2], and Vittorio Murino[2,3]

[1] ETH Zürich, Department of Computer Science, Zürich, Switzerland
[2] University of Verona, Department of Computer Science, Verona, Italy
[3] Istituto Italiano di Tecnologia (IIT), Genova, Italy

Abstract. We consider a Multiple Kernel Learning (MKL) framework for nuclei classification in tissue microarray images of renal cell carcinoma. Several features are extracted from the automatically segmented nuclei and MKL is applied for classification. We compare our results with an incremental version of MKL, support vector machines with single kernel (SVM) and voting. We demonstrate that MKL inherently combines information from different input spaces and creates statistically significantly more accurate classifiers than SVMs and voting for renal cell carcinoma detection.

Keywords: MKL, renal cell carcinoma, SVM

1 Introduction

Cancer tissue analysis consists of several consecutive estimation and classification steps which require intensive labor practice. The tissue microarray (TMA) technology enables studies associating molecular changes with clinical endpoints [6]. In this technique, $0.6mm$ tissue cylinders are extracted from primary tumor blocks of hundreds of different patients, and are subsequently embedded into a recipient paraffin block. Such array blocks can then be used for simultaneous analysis of primary tumors on DNA, RNA, and protein level.

In this work, we consider the computer based classification of tissue from renal cell carcinoma (RCC) after such a workflow has been applied. The tissue has been transferred to an array and stained to make the morphology of cells and cell nuclei visible. Current image analysis software for TMAs requires extensive user interaction to properly identify cell populations on the TMA images, to select regions of interest for scoring, to optimize analysis parameters and to organize the resulting raw data. Because of these drawbacks, pathologists typically collect tissue microarray data by manually assigning a composite staining score for each spot. Such manual scoring can result in serious inconsistencies between data collected during different microscopy sessions. Manual scoring also introduces a significant bottleneck that limits the use of tissue microarrays in high-throughput analysis.

[⋆] Corresponding author. †Equal contributors

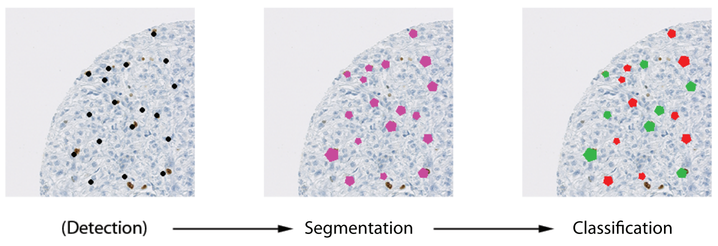

Fig. 1. One keypoint in the automatic TMA analysis for renal cell carcinoma is the nucleus classification. Nuclei are eosin stained and visible in the TMA image as dark blue spots. We want to simulate the classification of cell nuclei into cancerous or benign, which is recently done by trained pathologists by eye. The automatic approach comprises nucleus detection on the image, the segmentation of the nuclei and the classification, all based on training data labeled by two human experts.

The manual rating and assessment of TMAs under the microscope by pathologists is quite unconsistent due to the high variability of cancerous tissue and the subjective experience of humans, as shown in [4]. Therefore, decisions for grading and/or cancer therapy might be inconsistent among pathologists. With this work, we want to contribute to a more generalized and reproducible system that automatically processes TMA images and thus helps pathologists in their daily work.

For various classification tasks, SVM formulations involve using one data set and maximizing the margin between different classes. This poses a restriction on some problems, where different data representations are used. Combining the contribution of different properties is important in discriminating between cancerous and healthy cells. Multiple Kernel Learning (MKL) is a recent and promising paradigm, where the decisions of multiple kernels are combined to achieve better accuracies [1]. The advantage of this idea is to be able to utilize data from multiple sources. In MKL, multiple kernels are combined (see Section 3) globally. We also compare this idea with the usual classifier combination where outputs of multiple classifiers are combined [7, 9].

In previous work, an automated pipeline of TMA processing was already proposed, concentrating on the investigation of various image features and associated kernels on the performance of a support vector machine classifier for cancerous cells [12]. In this work, we follow this workflow and extend the nucleus classification (Figure 1) by using MKL that combines information from multiple sources (in our case different representations). By considering different types of features, we show in Section 4 the importance of using shape features; our results show that MKL reaches significantly better accuracies than SVM and voting (VOTE) using the combination of multiple kernels.

Our contribution is to show how information from different representations can make this classification task easier: the MKL algorithm inherently combines data from different representations to get better classification accuracies. Instead of combining outputs of multiple classifiers, MKL uses an optimization procedure where data from all sources are seen during training and optimization is done accordingly. Our experiments demonstrate that although it is more costly to use MKL, the increase in accuracy is worth its cost.

The paper is organized as follows: in Section 2, we introduce the data set used in this study. We explain the methods applied in Section 3, and show our experiments in Section 4. We conclude in Section 5.

2 Data Set

2.1 Tissue Micro Arrays

Small round tissue spots of cancerous tissue are attached to TMA glass plate. The diameter of the spots is 1mm and the thickness corresponds to one cell layer. Eosin staining made the morphological structure of the cells visible, so that cell nuclei appear bluish in the TMAs. Immunohistochemical staining for the proliferation protein MIB-1 (Ki-67 antigen) makes nuclei in cell division status appear brown.

For computer processing, the TMA slides were scanned with a magnification of 40x, resulting in a per pixel resolution of $0.23\mu m$. The final spots of single patients are separately extracted as three channel color images of size 3000x3000px.

In this study, we used the top left quarter of eight tissue spots from eight patients. Therefore, each image shows a quarter of the whole spot, i.e. 100-200 cells per image (see Figure 2).

For training our models, the TMA images were independently labeled by two pathologists [4]. From such eight labeled TMA images, we extracted 1633 nuclei-patches of size 80x80 pixels. Each patch shows a cell nucleus in the center (see Figure 3). 1273 (78 %) from the nuclei form our datase, where the two pathologists agree on the label: 891 (70 %) benign and 382 (30 %) malignant nuclei.

2.2 Image Normalization and Patching

The eight images were adjusted in contrast to minimize illumination variances among the scans. To classify the nuclei individually, we extracted patches from the whole image such that each 80x80px patch has one nucleus in the center (see Figure 3). The locations of the nuclei were known from the labels of the pathologists. Both procedures drastically improved the following segmentation of cell nuclei.

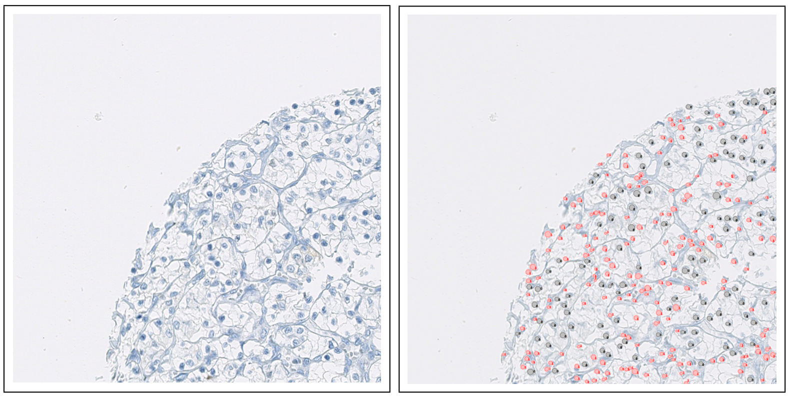

Fig. 2. Left: One 1500x1500px quadrant of a TMA spot from a RCC patient. **Right:** A pathologist exhaustively labeled all cell nuclei and classified them into malignant (black) and benign (red).

2.3 Segmentation

The segmentation of cell nuclei was performed with graphcut [3]. The gray intensities were used as unary potentials. The binary potentials were linearly weighted based on their distance to the center to prefer roundish objects lying in the center of the patch (see Figure 3). The contour of the segmented object was used to calculate several shape features as described in the following section.

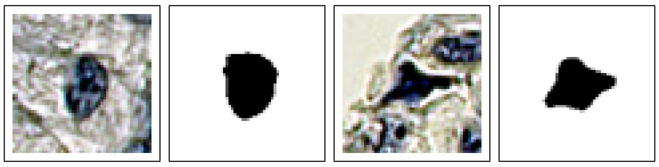

Fig. 3. Two examples of nucleus segmentation. The original 80x80 pixel patch are shown, each with the corresponding nucleus shape found with graphcut.

2.4 Feature extraction

For training and testing the various classifiers we extracted several histogram-like features from the patches (see Table 1).

Table 1. Features extracted from patch images for training and testing. Except the PROP feature, all features are histograms normalized to sum up to one.

Shortcut	Feature Description
ALL	**Patch Intensity**: A 16-bin histogram of gray scaled patch
FG	**Foreground Intensity**: A 16-bin histogram of nucleus
BG	**Background Intensity**: A 16-bin histogram of background
LBP	**Local Binary Patterns**: This local feature has been shown to bring considerable performance in face recognition tasks. It benefits from the fact that it is illumination invariant.
COL	**Color feature**: The only feature comprising color information. The colored patch (RGB) is rescaled to size 5x5. The 3x25 channel intensities are then concatenated to feature vector of size 75.
FCC	**Freeman Chain Code**: The FCC describes the nucleus' boundary as a string of numbers from 1 to 8, representing the direction of the boundary line at that point ([5]). The boundary is discretized by subsampling with grid size 2. For rotational invariance, the first difference of the FCC with minimum magnitude is used. The FCC is represented in a 8-bin histogram.
SIG	**1D-signature**: Lines are considered from the object center to each boundary pixel. The angles between these lines form the signature of the shape ([5]). As feature, a 16-bin histogram of the signature is generated.
PHOG	**Pyramid histograms of oriented gradients**: PHOGs are calculated over a level 2 pyramid on the gray-scaled patches ([2]).
PROP	**Shape descriptors** derived from MATLAB's `regionprops` function: `Area BoundingBox(3:4)`, `MajorAxisLength`, `MinorAxisLength`, `ConvexArea`, `Eccentricity`, `EquivDiameter`, `Solidity`, `Extent`, `Perimeter`, `MeanIntensity`, `MinIntensity`, `MaxIntensity`;

3 Methodology

In this section, we summarize the MKL framework behind our experiments. The main idea behind support vector machines [14] is to transform the input feature space to another space (possibly with a greater dimension) where the classes are linearly separable. After training, the discriminant function of SVM becomes $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle + b$, where $\boldsymbol{w}$ are the weights, b is the threshold and $\Phi(\boldsymbol{x})$ is the mapping function. Using dual formulation and the kernels one does not have to define this mapping function $\Phi(\boldsymbol{x})$ explicitly and the discriminant becomes as in (1) where $K(\boldsymbol{x}_i, \boldsymbol{x})$ is the kernel.

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \ . \tag{1}$$

Using SVM with a single kernel would restrict us to use one feature set (or a concatenation of all feature sets) and complicates the possibility to exploit the information coming from different sources. As in classifier combination [7], we can combine multiple kernels using different feature sets and use this information to come up with more accurate classifiers [9]. The simplest way for this is to use an unweighted sum of kernel functions [10]. Lanckriet et al. [8] have formulated this semidefinite programming problem which allows finding the combination weights and support vector coefficients together. Bach et al. [1] reformulated the problem and proposed an efficient algorithm using sequential minimal optimization (SMO). Using Bach's formulation, with P kernels, the discriminant function becomes as in (2) where m indexes the kernel:

$$f(\boldsymbol{x}) = \sum_{m=1}^{P} \eta_m \sum_{i=1}^{N} \alpha_i y_i K_m(\boldsymbol{x}, \boldsymbol{x}_i) + b \,. \tag{2}$$

This allows us to combine different kernels in different feature spaces and this is the formulation we apply in this work. Here, kernels are combined globally, namely the kernels are assigned the same weights for the whole input space.

It has been shown by many researchers that using a subset of given classification algorithms increases accuracy rather than using all the classifiers [11, 13]. Keeping this in mind, we apply the same idea to incrementally adding kernels to the MKL framework and compare the results.

The incremental algorithm works as follows: It starts with the most accurate kernel (classifier) on the validation folds (leave-the-other-fold-out), and adds kernels (classifiers) to the combination one by one. This procedure continues until all kernels (classifiers) are used or the average validation accuracy does not increase [13]. The algorithm starts with $E^0 \leftarrow \emptyset$, then at each step t, all the kernels (classifiers) $M_j \notin E^{(t-1)}$ are combined with $E^{(t-1)}$ to form S_j^t ($S_j^t = E^{(t-1)} \cup M_j$). We select S_{j*}^t which is the ensemble with the highest accuracy. If accuracy of S_{j*}^t is higher than $E^{(t-1)}$, we set $E^t \leftarrow S_{j*}^t$ and continue, else the algorithm stops and returns $E^{(t-1)}$.

4 Experiments

4.1 Experiment Setup

The data of 1273 nuclei samples is divided into ten folds (with stratification). We then train support vector machines (*svl, sv2, svg*, see below) and MKL using these folds. We also combine the support vector machines using voting and report average accuracies using 10-fold CV. For the Gaussian kernel, σ is chosen using a rule of thumb: $\sqrt{D}$ where D is the number of features of the data representation. We compare our results using 10-fold CV t-test at $p = 0.05$. In the incremental learning part, we apply leave-the-other-fold-out cross validation (used for validation) to estimate which kernel and classifier should be added.

As a summary, we have 9 representations (All, Bg, Col, Fcc, Fg, Lbp, Phog, Sig and Prop), three different kernels (linear kernel: *svl*, polynomial kernel with degree 2: *sv2*, and Gaussian kernel: *svg*), and two combination algorithms (MKL, VOTE).

The SVM accuracies with each individual kernel are reported in Table 2. The best accuracy using a single SVM is 76.9 %. For most representations (except Phog and Col), the accuracies of different kernels are comparable.

Table 2. Single support vector accuracies (± std) in %.

	svl	*sv2*	*svg*
All	70.0±0.2	71.6±2.9	72.0±3.2
Bg	70.0±0.2	71.2±2.6	68.9±2.3
Col	70.1±0.2	63.6±3.5	66.2±2.3
Fcc	70.0±0.2	70.0±0.2	67.4±1.6
Fg	70.0±0.2	70.0±3.2	70.5±3.5
Lbp	70.0±0.2	66.9±3.0	68.7±4.4
Phog	76.5±3.7	72.0±3.3	**76.9±3.6**
Sig	70.0±0.2	68.6±2.5	66.6±2.6
Prop	75.7±2.3	75.6±2.6	74.1±1.8

Next, we use the same kernel and combine all the feature sets we extracted. As shown in Table 3 (top), we can achieve an accuracy of 81.3 % using the linear kernel, by combining all representations using the same kernel. This shows that the combination of information from multiple sources might be important and, by using MKL, the accuracy can be increased around 5 %. We observe from the table also that when we use all kernels with *sv2*, we have a decrease in accuracy compared to the single best support vector machine. This is analogous to combining all classifiers in classifier combination. If one has relatively inaccurate classifiers, combining all may decrease accuracy. Instead, it might be better to choose a subset. This also shows that medically, all the information is complementary and should be used to achieve better accuracy. In Figure 4, we plotted the weights of MKL when we use the linear kernel. As expected, the two best representations Phog and Prop have high weights. But the representation Lbp that has very low accuracy when considered as a single classifier increases the accuracy when considered in combination. This shows that when considering combinations, even a representation which is not very accurate alone may contribute to the combination accuracy. From this, we also deduce that these three features are useful in discriminating between healthy and cancerous cells and we may focus our attention on these properties.

On the bottom part of Table 3, the results using the incremental algorithm are shown. We can see that we do not have an increase in accuracy compared to the best single support vector machine. In fact, the incremental algorithm cannot find a second complementary kernel which will increase accuracy when added to the single best. In principle, we expect the incremental algorithm to have better accuracies than combining all classifiers. We see this behavior for *sv2*. When we consider *svl*, combining all kernels seems to be better than the subset selection strategy. This might partially result from the fact that the incremental algorithm could not find a complementary kernel, and partially from the optimization formulation of MKL. In the incremental search, we discard kernels which do not improve the overall accuracy. On the other hand, in MKL, every kernel is given a weight and all kernels contribute to the solution of the problem. From

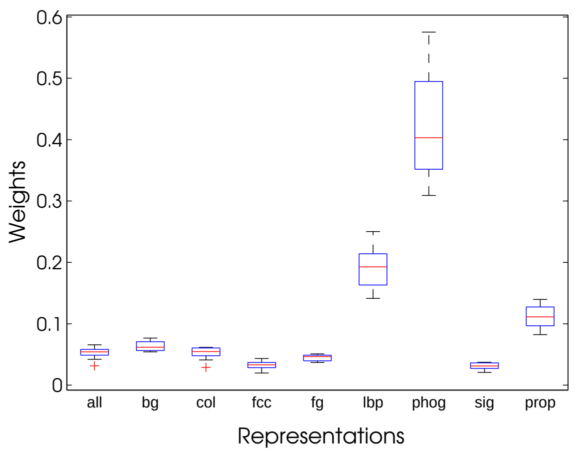

Fig. 4. Combination weights in MKL using the linear kernel.

this, we can say that it is better to use MKL instead of combining outputs of support vector machines using voting. We can also see the support of this claim in Table 3. When we use voting, combining all classifiers is always worse than the single best and always worse than MKL because the optimization procedure does not see the data, but only combines outputs of all classifiers. On the other hand, when we apply the incremental paradigm, we achieve better results than MKL because there are complementary classifiers which increase the accuracy.

Table 3. MKL accuracies (in %). Top: accuracy (± std) of combining all kernels. Bottom: accuracies calculated using the incremental algorithm, the number of kernels/classifiers selected.

	svl	*sv2*	*svg*
MKL	**81.3±3.6**	72.0±3.3	76.9±3.6
VOTE	70.0±0.2	71.3±1.7	72.4±1.2
MKL		76.9±3.6, 1	
VOTE		78.9±2.5, 4	

4.2 Discussion

We have seen that MKL performs better than VOTE and SVMs with single kernel, when all kernels are combined. This is because the optimization procedure takes into account all data and gives weights to all kernels, so it can use all representations. On the other hand, when we apply the incremental algorithm, classifier combination achieves better accuracies than combining all classifiers. MKL combines the underlying feature sets to make a better combination. In this work, we used three different kernels and two combination schemes to see how the change of each parameter effects the classification accuracy. We see that, when we use single support vector machines, all the kernels have comparable accuracies. The importance of each kernel function increases when the combination is considered, and combining outputs is less effective than combining the kernels themselves using optimization.

Also, we have seen that when we use the multiple kernel learning algorithm, we gain 5 % in accuracy compared to SVMs with single kernel. Combining all kernels here comes with a drawback. We have to use all kernels and extract all the features when we have to use this model but the increase in accuracy might be worth the cost. We see that when we use the incremental algorithm, we cannot add any kernels, so we are stuck in a local minimum. When we combine classifiers on the other hand, the incremental algorithm achieves more accurate results. Nevertheless, the best results are obtained when we use all representations using *svl* and this accuracy is the best result we have reached so far.

5 Conclusion

In this paper, we propose the use of the multiple kernel learning paradigm for the classification of nuclei in TMA images of renal clear cell carcinoma. We used support vector machines extensively through different feature sets in our previous work. This study extends those works by using several feature sets in a multiple kernel learning paradigm and compares the results with single support vector machines and combining outputs of support vector machines using voting.

We have seen that MKL performs better than SVMs and VOTE in most of the experiments. MKL exploits the underlying contribution of each feature set and heterogeneity of the problem, and by using multiple kernels, achieves better results than single kernels and voting of classifiers.

In this work, we used image based feature sets for creating multiple features. In a further application of this scenario, the use of other modalities or other features (e.g. SIFT) extracted from these images, as well as the incorporation of complementary information of different modalities to achieve better classification accuracy is possible. The incremental algorithm as implemented in this scenario does not work as well as combining all kernels using MKL. As a future work, we would like to implement other heuristics (decremental search, two step look-ahead incremental search, floating search etc.) so that we can achieve better accuracies without imposing too much cost on the system and using only a few kernel combinations. We also would like to apply a local multiple kernel combination framework which is analogous to classifier selection in ensemble framework where the combination also depends on the input which puts forward the inherent localities of the data sets and automatically divides the data set into subsets within the optimization procedure.

Acknowledgements

We thank Dr. Cheng Soon Ong very much for helpful discussions. Also, we acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

1. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the smo algorithm. In: Proceedings of the twenty-first international conference on Machine learning, ICML '04. pp. 41–48 (2004)
2. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval. pp. 401–408. ACM, New York, NY, USA (2007)
3. Boykov, Y., Veksler, O., Zabih, R.: Efficient approximate energy minimization via graph cuts. IEEE transactions on Pattern Analysis and Machine Intelligence 20(12), 1222–1239 (November 2001)
4. Fuchs, T.J., Wild, P.J., Moch, H., Buhmann, J.M.: Computational pathology analysis of tissue microarrays predicts survival of renal clear cell carcinoma patients. MICCAI (2008)
5. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital image processing using matlab (2003), 993475
6. Kononen J, Bubendorf L, e.a.: Tissue microarrays for high-throughput molecular profiling of tumor specimens. Nat Med. Jul;4(7), 844–7 (1998)
7. Kuncheva, L.I.: Combining pattern classifiers: methods and algorithms. Wiley-Interscience (2004)
8. Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research 5, 27–72 (December 2004)
9. Lee, W.J., Verzakov, S., Duin, R.P.W.: Kernel combination versus classifier combination. In: Proceedings of the 7th international conference on Multiple classifier systems, MCS'07. pp. 22–31 (2007)
10. Moguerza, J.M., Muoz, A., de Diego, I.M.: Improving support vector classification via the combination of multiple sources of information. In: Fred, A., Caelli, T., Duin, R.P.W., Campilho, A., Ridder, D.d. (eds.) Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science, vol. 3138, pp. 592–600. Springer Berlin / Heidelberg (2004)
11. Ruta, D., Gabrys, B.: Classifier selection for majority voting. Information Fusion 6(1), 63–81 (2005)
12. Schüffler, P.J., Fuchs, T.J., Ong, C.S., Roth, V., Buhmann, J.M.: Computational tma analysis and cell nucleus classification of renal cell carcinoma. In: Proceedings of the 32nd DAGM conference on Pattern recognition, DAGM '10. pp. 202–211. Springer-Verlag, Berlin, Heidelberg (2010)
13. Ulaş, A., Semerci, M., Yıldız, O.T., Alpaydın, E.: Incremental construction of classifier and discriminant ensembles. Information Sciences 179(9), 1298–1318 (April 2009)
14. Vapnik, V.N.: Statistical learning theory. John Wiley and Sons (1998)