ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Learning Dot Product Polynomials for multiclass problems

Lauriola Ivano[1], Donini Michele[2] and Aiolli Fabio[1]

1- Department of Mathematics, University of Padova
via Trieste 63, Padova, Italy

2- Computational Statistics and Machine Learning (CSML)
Istituto Italiano di Tecnologia, Via Morego 30, Genova, Italy

Abstract. Several mechanisms exist in the literature to solve a multiclass classification problem exploiting a binary kernel-machine. Most of them are based on problem decomposition that consists on splitting the problem in many binary tasks. These tasks have different complexity and they require different kernels. Our goal is to use the Multiple Kernel Learning (MKL) paradigm to learn the best dot-product kernel for each decomposed binary task. In this context, we propose an efficient learning procedure to reduce the searching space of hyperparameters, showing its empirically effectiveness.

1 Introduction

Kernel machines are a class of machine learning algorithms for classification and pattern analysis by exploiting a vector space, whose the best known member is the Support Vector Machine (SVM). Recently, Kernel Learning and Multiple Kernel Learning have been proposed as two paradigms used to learn the kernel directly from training data, reducing the user's effort on designing a good kernel for a given problem [1, 2].
The kernel obtained by a MKL algorithm is a combination of R *base* kernels $k_1, \dots, k_R$. Different combination mechanisms exist in literature. In this paper, we consider linear non-negative combinations of base kernels in the form: $k_\eta(\boldsymbol{x}, \boldsymbol{z}) = \sum_{r=1}^{R} \boldsymbol{\eta}_r k_r(\boldsymbol{x}, \boldsymbol{z}) = \sum_{r=1}^{R} \boldsymbol{\eta}_r \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{z}) \rangle, \boldsymbol{\eta}_r \geq 0$. In our case, base kernels consist of Homogeneous Polynomial Kernels (HPK) with different degree p, i.e. $k_p(\boldsymbol{x}, \boldsymbol{z}) = \langle \boldsymbol{x}, \boldsymbol{z} \rangle^p$. We refer to their combination as Dot-Product Polynomial (DPP) [3] in the form $k(\boldsymbol{x}, \boldsymbol{z}) = \sum_{r=0}^{\infty} \alpha_r \langle \boldsymbol{x}, \boldsymbol{z} \rangle^r$, $\alpha_r \geq 0$. In literature [4] it is already known that any dot-product kernel can be seen as a DPP, so using the MKL approach to learn the DPP coefficients means giving a method that can learn any dot-product kernel, including RBF and non-homogeneous polynomials.

One of the main limitations of kernel-based algorithms is that they often are designed for pure binary classification problems. There are several mechanisms used to solve a given multiclass problem exploiting a binary classifier; some of these, such as *one-vs-one* (o-v-o) and *one-vs-rest* (o-v-r), are based on problem decomposition. They consist on splitting the original problem in a set of binary tasks (that are easier to solve by using a binary classifier [5]). These approaches are very popular, but generally the selected kernel function is the same for each binary task. Typically, the user chooses the kernel that (on average) works well

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

for each of them, but in the real world applications the tasks can be different and they could require different kernels. In this paper, we learn the coefficients of DPPs for each decomposed task of a multiclass classification problem by using a MKL algorithm and showing the empirical effectiveness of our new methodology.

2 Notation and Background

In this paper we consider multiclass classification problems with training examples $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_L, y_L)\},\ \boldsymbol{x}_i \in \mathbb{R}^m,\ \|\boldsymbol{x}_i\|_2^2 = 1,\ y_i \in \{0, \dots, k-1\}$. We used EasyMKL [6], a scalable MKL algorithm, in order to combine the set of HPKs.

2.1 Multiclass methods based on problem decomposition

Among the different techniques to solve a multiclass problem with a binary classifier, we focus on problem decomposition methods. In particular, we use the o-v-r and o-v-o algorithms. Basically, these methods split the original problem in a set of binary tasks and perform predictions according to a voting procedure.

The easiest decomposition method is known as o-v-r. It generates k binary classifiers where k is the number of classes. The m-th classifier is trained with all the examples of the m-th class as positive labels, and all the other examples as negative labels. After solving the problems, we obtain k decision functions, one for each model. The decision function with the highest value defines the predicted class.

Another decomposition method is called o-v-o [7]. This method constructs $k(k-1)/2$ classifiers where each one is trained on data from two classes; these classifiers represent all the possible combinations between pairs of classes (i.e. dichotomies). Different methods are available to do a prediction on a new example. We used the following voting strategy: if the classifier for the pair of classes i and j classifies the instance $\boldsymbol{x}$ in the i-th class, then the vote of i-th class is increased by 1. The class of $\boldsymbol{x}$ is the class with the highest number of votes.

3 Learning kernels in the space of DPPs

As mentioned previously, we consider as base kernels a set of normalized HPKs with different degree p, $k_p(\boldsymbol{x}, \boldsymbol{z}) = \frac{\langle \boldsymbol{x}, \boldsymbol{z} \rangle^p}{\sqrt{\|\boldsymbol{x}\| \cdot \|\boldsymbol{z}\|}}$. We refer to their combinations as Dot Product Polynomial (DPP), that is any linear non-negative combination of HPKs with $k(\boldsymbol{x}, \boldsymbol{z}) = \sum_{r=0}^{\infty} \alpha_r \langle \boldsymbol{x}, \boldsymbol{z} \rangle^r$, $\alpha_r \geq 0\ \ \forall r$. A well-known theorem from harmonic theory [4,8] asserts that any dot-product kernel can be characterized by its Maclaurin expansion with non-negative coefficients (i.e. a DPP). In practice, any choice of non-negative coefficients of DPP induces a valid kernel.

The Radial Basis Function (RBF) is a popular kernel defined as $k(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{z}\|_2^2\right)$, where $\gamma \in \mathbb{R}_+$ is an hyperparameter. If we consider normalized training examples $\|\boldsymbol{x}\|_2^2 = 1$, the kernel can be characterized by a Maclaurin expansion, so it is a DPP [9]. Its DPP coefficients are constrained in a Gaussian curve and they depend strictly on the value of γ. Consequently, the selection

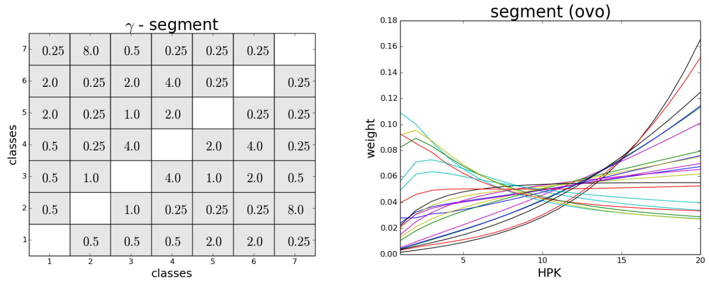
ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

of γ means a fixed choice of this curve. Now we want to take a step forward by making it non-parametric. Specifically, we propose to use the EasyMKL algorithm to learn the coefficients of DPP. In this way, any dot-product kernel, including RBF, anisotropic RBF [10] and non-homogeneous polynomials, can be generalized. Table 1 shows the DPP coefficients for some well-known dot product kernels.

Kernel	Definition	DPP r-th coefficient
Polynomial	$(\boldsymbol{x}^\top \boldsymbol{z} + c)^D$	$\binom{D}{r} c^{D-r}$
RBF	$e^{-\gamma\|\boldsymbol{x}-\boldsymbol{z}\|^2}$	$e^{-2\gamma}\frac{(2\gamma)^{2r}}{r!}$
Rational Quadratic	$1 - \frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{\|\boldsymbol{x}-\boldsymbol{z}\|^2+c}$	$\left(-\frac{2\prod_{j=1}^{r} 2+(j-1)}{(2+c)^{r+1}} + \frac{\prod_{j=1}^{r} 2+(j-1)}{(2+c)^{r}}\right)\frac{1}{r!}$
Cauchy	$\left(1 + \frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{\gamma}\right)^{-1}$	$\frac{r!!}{3^{r+1}\gamma^{r}}\frac{1}{r!}$

Table 1: DPP coefficients for some known dot-product kernels.

4 Experimental Setup



(a) best RBF parameter γ (b) HPKs coefficients learned

Fig. 1: a) the entry ij represents the best RBF kernel with parameter γ selected a posteriori using the classes i and j with an SVM(C=1); b) the DPP coefficients obtained by EasyMKL for each binary task using 20 HPKs with degrees $1, \ldots, 20$.

First of all, we performed a set of preliminary experiments as sanity check of our methodology. Fig. 1 highlights the importance of using a different kernel for each different pair of tasks in the o-v-o methodology. To learn the o-v-o and o-v-r multiclass models, we divided the training phase in two steps. In the first step, we choose the kernel in the following ways: (1) K_{MKL} (our method), the kernel obtained using EasyMKL and HPKs; (2) K_{SUM}, the average of base kernels; (3) K_{RBF}, RBF kernel. In the latter step, we use an SVM as a base learner with kernels from the previous step. The hyperparameters that need to be validated are λ for K_{MKL}, γ for K_{RBF} and C for all the models[1]. For comparison, we tested several multiclass problems from UCI repository [11] (iris, wine, glass, vowel and pendigits), from StatLog (vehicle, segment, dna, satimage and letter), usps from [12], digits from scikit-learn and keystrokes.

[1] λ, γ and C are hyperparameters of EasyMKL algorithm, RBF kernel and SVM respectively.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

For keystrokes, dna, satimage, usps, pendigits, and letter, test-sets are available, and the hyperparameters are chosen according to the hold-out pattern by splitting the training set in train (70%) and validation (30%). Then, another model is learned using all the training examples and the hyperparameters which obtained the best result in validation. For the other datasets, a stratified nested 10-fold cross validation (CV) has been used. The accuracy is computed by predicting the labels of the test set for the datasets with the hold-out pattern. Otherwise, the accuracy is the average of each fold in the nested CV procedure. For each model, we have evaluated $C \in \{2^i : i = -2, -1, \dots, 8\}$, $\lambda \in \{0.0, 0.1, \dots, 1.0\}$ and $\gamma \in \{2^i : i = -2, -1, \dots, 4\}$. Fig. 2 contains the results obtained from K_{MKL}, K_{SUM} and K_{RBF} varying the number of HPKs used. Note that in our approach we have another hyperparameter, that is the polynomial degree D, e.g. the number of HPKs used in combination.
In our experiments we considered only normalized examples due to different reasons: 1) RBF can be defined as a DPP only when examples are normalized, 2) with non-normalized data, HPKs receive different weights from EasyMKL due to its combination mechanism (see [6]).

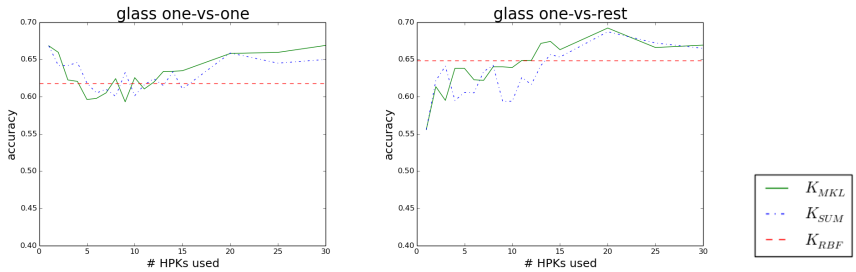


Fig. 2: Accuracy of K_{MKL}, K_{SUM} (with different number of HPKs) and K_{RBF}.

4.1 Heuristic selection of the hyperparameter D

We tried to reduce the number of hyperparameters from the tuple (D, λ, C). λ and C are two parameters of EasyMKL and SVM that aim to solve the non-separability of the data in two different learning steps. We tried to fix $\lambda = 0$ ($K^{\lambda=0}_{MKL}$) and validate the pair (C, D). Fig. 3 describes the influence of the λ value in the final results. A further optimization is based on the ratio $\frac{r^2}{\rho^2}$ between the radius of the Minimal Enclosing Ball (MEB) of the examples r and the margin ρ. Fig. 4 shows a strong correlation between the ratio and the generalization error in the context of DPP with different polynomial degrees. The best degree D* for our DPP is the one with the minimal ratio. In particular, we used the following algorithm: for each iteration i, we calculate r as the average radius of MEB and ρ as the average margin considering all the binary tasks in the space defined by the i-degree DPP. Then, if $\frac{r^2}{\rho^2}$ is less than the previous step, we continue the loop, else we return $i - 1$.

In each iteration, we considered the average values of ρ and r to reduce outlier values due to non-separable cases, and we used EasyMKL($\lambda = 0$) to

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1. Available from http://www.i6doc.com/en/.

dataset	$K_{MKL}^{D=30}$		$K_{MKL}^{\lambda=0}$		K_{MKL}^{C}		$K_{SUM}^{D=30}$		K_{RBF}	
	ovo	ovr	ovo	ovr	ovo	ovr	ovo	ovr		
iris	0.973 ±0.044	0.967 ±0.054	**0.980** ±0.043	0.973 ±0.053	0.827 ±0.068	0.807 ±0.047	0.973 ±0.053	0.967 ±0.054	0.973 ±0.053	0.967 ±0.054
wine	0.973 ±0.027	0.967 ±0.027	0.984 ±0.025	**0.989** ±0.022	0.978 ±0.027	0.983 ±0.026	0.961 ±0.035	0.978 ±0.027	0.973 ±0.027	0.983 ±0.025
glass	0.669 ±0.094	0.669 ±0.117	0.674 ±0.097	**0.682** ±0.105	0.648 ±0.104	0.662 ±0.141	0.650 ±0.100	0.665 ±0.128	0.618 ±0.04	0.649 ±0.124
vowel	0.850 ±0.052	0.845 ±0.048	0.850 ±0.052	0.845 ±0.048	**0.944** ±0.027	0.927 ±0.031	0.844 ±0.054	0.840 ±0.061	0.865 ±0.043	0.860 ±0.046
vehicle	0.795 ±0.030	0.817 ±0.029	0.845 ±0.026	0.854 ±0.031	0.780 ±0.029	0.778 ±0.029	0.793 ±0.024	0.800 ±0.024	0.843 ±0.022	**0.858** ±0.023
digits	0.981 ±0.015	0.984 ±0.010	0.983 ±0.010	0.984 ±0.010	0.982 ±0.012	0.984 ±0.011	0.981 ±0.015	0.984 ±0.010	0.983 ±0.011	0.983 ±0.011
segment	0.974 ±0.012	0.974 ±0.012	0.973 ±0.009	0.972 ±0.012	**0.977** ±0.014	0.973 ±0.013	0.973 ±0.013	0.972 ±0.012	0.968 ±0.014	0.973 ±0.010
keystrokes	0.215	0.235	0.208	0.212	**0.242**	0.208	0.208	0.239	0.204	0.208
dna	0.948	0.953	0.952	0.950	0.952	0.954	0.906	0.912	0.954	**0.955**
satimage	**0.920**	0.919	0.919	0.903	0.919	0.865	0.918	0.919	0.911	0.909
usps	0.950	0.9450	0.953	0.950	0.955	0.953	0.948	0.950	**0.956**	0.953
pendigits	0.996	0.996	0.996	0.996	0.996	0.995	0.996	0.996	0.996	0.995
letter	0.973	-	0.973	-	0.973	-	0.972	0.973	0.972	0.973

Table 2: Accuracy of $K_{MKL}^{D=30}$, $K_{MKL}^{\lambda=0}$, $K_{MKL}^{C} K_{SUM}^{D=30}$ and K_{RBF}.

combine i different HPKs with degree from 1 to i. In this way, C and the latter learning step are not required, making the complexity to find D* linear respect to the maximum degree D_{max}. Then, we used D* to fit the complete model with parameters ($\lambda = 0, D = \mathrm{D}^*, C$ selected by validation). The global search complexity is $O(D_{max} + |\mathcal{C}|)$, where $\mathcal{C}$ is the set of C values used in validation. Table 2 shows the accuracy scores obtained in each dataset.
This method, that we call K_{MKL}^{C}, and the average of base kernels $K_{SUM}^{D=30}$ have only C as hyperparameter, so they are faster than $K_{MKL}^{D=30}$, $K_{MKL}^{\lambda=0}$ and K_{RBF}, where the hyperparameters are respectively (λ, C), (D, C) and (γ, C). In $K_{MKL}^{D=30}$ and $K_{SUM}^{D=30}$ we fixed $D = 30$ a priori to reduce the search complexity. Due to different implementations of SVM (libsvm) and EasyMKL(Python) used, we consider only the asymptotic complexity and not the computational time. Our code is available online on the Python Package Index repository (PyPI) with the name "MKLpy: *a scikit-compliant framework for Multiple Kernel Learning*"[2].

5 Conclusions and future work

In the context of multiclass, we showed that each task obtained by a decomposition method has different complexity and it requires a different kernel. We tried to learn different DPPs coefficients for each task using MKL and we showed a procedure to make the searching complexity linear compared to the quadratic complexity of validating with respect to the classic pair (γ, C), holding an high accuracy score. In the future we want to improve the mechanisms to learn the D value using an approximation of the radius of MEB.

[2]Our MKLpy code: `https://pypi.python.org/pypi/MKLpy`

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

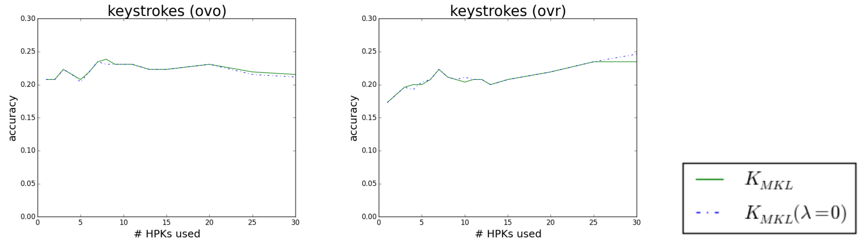


Fig. 3: Accuracy for K_{MKL} and $K_{MKL}^{\lambda=0}$.

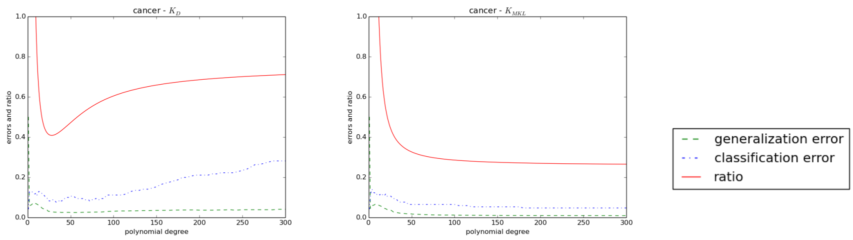


Fig. 4: Classification error, generalization error and ratio on single HPK (K_D) and $K_{MKL}$with different degree for cancer dataset (UCI, binary).

References

[1] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12(Jul):2211–2268, 2011.

[2] Verónica Bolón-Canedo, Michele Donini, and Fabio Aiolli. Feature and kernel learning. In *Proceedings*, page 173. Presses universitaires de Louvain, 2015.

[3] Michele Donini and Fabio Aiolli. Learning deep kernels in the space of dot product polynomials. *Machine Learning*, pages 1–25, 2016.

[4] IJ Schoenberg. Positive definite functions on spheres. *Duke Math. J*, 9(1):96–108, March 1942.

[5] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[6] Fabio Aiolli and Michele Donini. Easymkl: a scalable multiple kernel learning algorithm. *Neurocomputing*, 169:215–224, 2015.

[7] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing*, pages 41–50. Springer, 1990.

[8] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *AISTATS*, volume 22, pages 583–591, 2012.

[9] Andrew Cotter, Joseph Keshet, and Nathan Srebro. Explicit approximations of the gaussian kernel. *arXiv preprint arXiv:1109.4603*, 2011.

[10] Fabio Aiolli and Michele Donini. Learning anisotropic rbf kernels. In *International Conference on Artificial Neural Networks*, pages 515–522. Springer, 2014.

[11] D.J. Newman A. Asuncion. UCI machine learning repository, 2007.

[12] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.