KEOPS: KERNELS ORGANIZED INTO PYRAMIDS

Marie Szafranski

ENSIIE
IBISC – Université d'Évry Val d'Essonne
Évry, France

*Yves Grandvalet**

Université de Technologie de Compiègne
CNRS UMR 7253 Heudiasyc
Compiègne, France

ABSTRACT

Data representation is a crucial issue in signal processing and machine learning. In this work, we propose to guide the learning process with a prior knowledge describing how similarities between examples are organized. This knowledge is encoded in a tree structure that represents nested groups of similarities that are the *pyramids of kernels*. We propose a framework that learns a Support Vector Machine (SVM) on pyramids of arbitrary heights and identifies the relevant groups of similarities groups are relevant for classifying the examples. A weighted combination of (groups of) similarities is learned jointly with the SVM parameters, by optimizing a criterion that is shown to be an equivalent formulation regularized with a mixed norm of the original fitting problem. Our approach is illustrated on a Brain Computer Interfaces classification problem.

Index Terms— Classification; Kernel methods; Multiple Kernel Learning; Structured sparsity; Brain Computer Interfaces.

1. INTRODUCTION AND RELATED WORKS

Kernel methods for supervised classification. Supervised classification aims to estimate a decision function able to predict the label y of a pattern $\boldsymbol{x}$. In binary classification, learning relies on a sample $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \{\pm 1\}$. In Support Vector Machines (SVM), the examples are implicitely mapped to a feature space via a mapping $\Phi : \mathcal{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) and $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the corresponding reproducing kernel.

The primary role of K is to define the evaluation functional in $\mathcal{H}$: $\forall f \in \mathcal{H},\ f(\boldsymbol{x}) = \langle f, K(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}}$. However, K also defines

- $\mathcal{H}$ itself, since $\forall f \in \mathcal{H}$, $f(\boldsymbol{x}) = \sum_{i=1}^{\infty} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$;
- a metric, and hence a smoothness functional in $\mathcal{H}$: $||f||_{\mathcal{H}}^2 = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_j)$;
- a similarity between pairs of examples, via the mapping Φ: $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}')\rangle_{\mathcal{H}}$.

Hence, the kernel participates to the success of the method and its choice is a crucial issue. This motivates works that may help to learn an appropriate kernel, such as *filters*, *wrappers* and *embedded* methods (see respectively [1], [2, 3, 4], and [5, 6] for instance). The Multiple Kernel Learning (MKL) framework introduced in [7] belongs to the family of *embedded* methods. It builds on standards SVM which minimize the following optimisation problem

$$(f^\star, b^\star) = \arg\min_{(f,b)} \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C\sum_{i=1}^{n}[1 - y_i(f(\boldsymbol{x}_i) + b)]_+ \ ,$$

where $[u]_+ = \max(0, u)$ is the hinge loss function and $C > 0$ controls the trade-off between the complexity of the model and the proportion of non-separable examples. The decision function of the resulting classification problem takes the shape of $\mathrm{sign}(f^\star(\boldsymbol{x}) + b^\star)$.

Learning with multiple unstructured kernels. In MKL, we are provided with M candidate kernels, $\{K_m\}_{m=1}^M$, and we wish to estimate the parameters of the SVM classifier together with the weights of a convex combination of the M kernels that defines the *effective kernel*. In [8], the authors propose to solve

$$(1)\quad \begin{cases} \displaystyle\min_{\substack{f_1,\ldots,f_M,\\ b,\,\boldsymbol{\sigma}}} & \displaystyle\frac{1}{2}\sum_{m=1}^{M}\frac{1}{\sigma_m}\|f_m\|_{\mathcal{H}_m}^2 \\ & \displaystyle + C\sum_{i=1}^{n}[1 - y_i(\sum_{m=1}^{M} f_m(\boldsymbol{x}_i) + b)]_+ \ , \\ \text{s. t.} & \displaystyle\sum_{m=1}^{M}\sigma_m \le 1\ ,\ \sigma_m \ge 0\ ,\ \forall m \in \{1,\ldots,M\}\ , \end{cases}$$

where $\forall m$, $\mathcal{H}_m$ is a RKHS with reproducing kernel K_m and σ_m is the coefficient applied to K_m. [1] In Problem (1), the constraints on coefficients σ_m favor sparse solutions regarding f_m and thus K_m.

Learning with structured multiple kernels. The selection or removal of kernels between or within predefined groups relies on the definition of a structure among kernels. This kind of structure has been widely investigated among variables in linear models. For instance, mixed norms correspond to groups defined as a partition of the set of variables (see [9] and references therein) while the Composite Absolute Penalties (CAP) introduced in [10] and further studied in [11] may also rely on a set of nested groups of variables $\mathcal{I} = \{\mathcal{G}_k\}_{k=1}^K$, with $\mathcal{G}_1 \subset \ldots \subset \mathcal{G}_k \subset \ldots \subset \mathcal{G}_K$. A CAP can be defined in RKHS as

$$\ell_{(\gamma_0;\gamma_1)} = \sum_{k=1}^{K}\Big(\sum_{m\in\mathcal{G}_k}\|f_m\|_{H_m}^{\gamma_1}\Big)^{\gamma_0/\gamma_1} \ , \qquad (2)$$

with $\gamma_1 = 2$ or ∞ and $\gamma_0 = 1$, in which case it favors the so-called hierarchical selection [10], that is, the selection of groups of kernels in the predefined order $\{\mathcal{I} \setminus \mathcal{G}_K\}, \{\mathcal{G}_K \setminus \mathcal{G}_{K-1}\}, \ldots, \{\mathcal{G}_2 \setminus \mathcal{G}_1\}, \mathcal{G}_1$ according to some heredity principle. The Hierarchical Kernel Learning (HKL) framework extends the CAP to a hierarchy of kernels embedded into a Directed Acyclic Graph [12].

Both HKL and KEOPS (KErnels Organized into PyramidS) are generalizations of MKL but their notion of hierarchy differs radically: HKL is based on a partial order of kernels whereas KEOPS, that will be introduced in Section 2, relies on nested groups of kernels. We extend the Composite Kernel Learning framework [13], which considers a partition of the set of kernels, to nested partitions. Similarities can then be grouped at different levels, enabling more flexibility for learning the effective kernel. We first develop the extension starting from Problem (1), before giving an equivalent formulation based on mixed norms.

*Work carried out in the framework of the Labex MS2T, funded by the French National Agency for Research (ANR-11-IDEX-0004-02).

[1] In here and in what follows, u/v is defined by continuation at zero as $u/0 = \infty$ if $u \neq 0$ and $0/0 = 0$.

Published in *IEEE ICASSP 2014*, May 04-09, 2014, Florence, Italy. This version contains slight modifications.

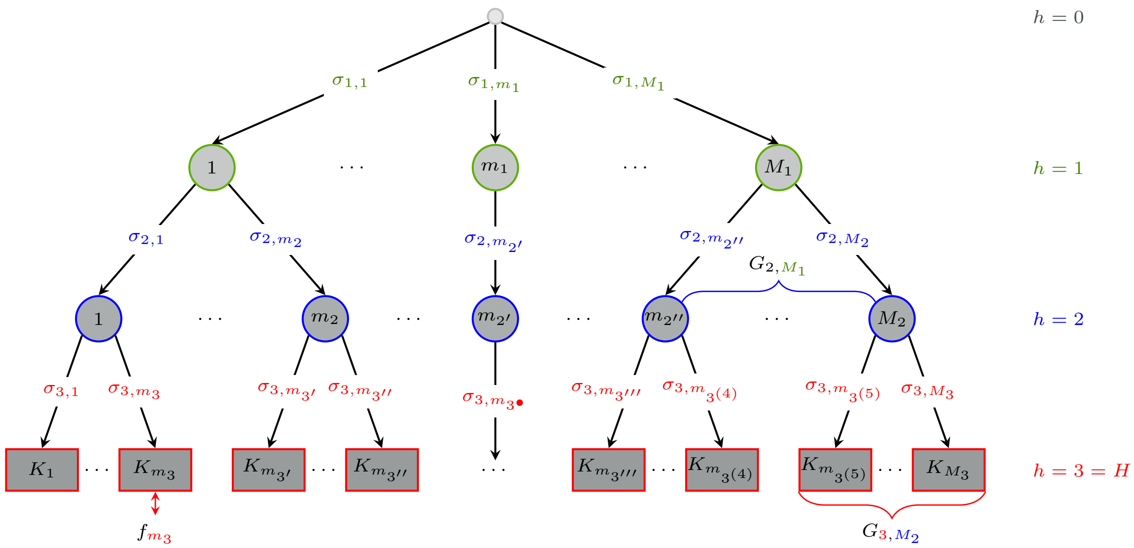

Fig. 1: A pictorial representation of notation for tree structure that symbolizes the pyramid of kernels.

2. FRAMEWORK

In MKL, the effective kernel is a flat linear combination of kernels. Our hierarchical combination, still linear, aims to encode richer assumptions about the structure of the solution. For this purpose, we consider a tree structure of kernels such as the one illustrated in Figure 1. Each node represents a kernel: the leaves are the pre-defined elementary kernels, and the kernel of their ancestor's nodes are defined recursively as one proceeds toward the root of the tree, by a weighted combination of their children. The root node itself thus represents the effective kernel.

At this point, we need to introduce some notation for the tree structure (see Fig. 1). The tree height is indexed by h, with $h = 0$ at the root and $h = H$ for the leaves. At height h, there are M_h nodes, indexed by $m \in \{1, \dots, M_h\}$. The set of children of node m at height h is denoted by $G_{h+1,m}$ (children are at height $h+1$). A coefficient $\sigma_{h,m}$ is attached to each edge of the tree: it weights the contribution of the mth kernel at height h to the computation of its parent kernel. That is, denoting $K_{h,m}$ the equivalent kernel of node m at height h, we have $K_{H,m} = K_m$ and $K_{h-1,m} = \sum_{m_h \in G_{h,m}} \sigma_{h,m_h} K_{h,m_h}$. For example, in Figure 1, the kernel at node m_1 at height 1 is defined as $\sum_{m_2 \in G_{2,m_1}} \sigma_{2,m_2} K_{m_2}$ and its contribution to the effective kernel is weighted by σ_{1,m_1}. In what follows, index m_h always runs in $G_{h,m}$.

In this framework, learning consists in assigning values to the all coefficients $\sigma_{h,m}$, which define the effective kernel, and to the SVM parameters. This learning problem is formalized as:

$$
(3)\quad \begin{cases}
\displaystyle\min_{\substack{f_1,\dots,f_{M_H},\\ b,\boldsymbol{\sigma}}} \ \frac{1}{2} \sum_{m_1} \frac{1}{{\sigma_{1,m_1}}^{p_1}} \ \dots \sum_{m_H} \frac{1}{{\sigma_{H,m_H}}^{p_H}} \|f_{m_H}\|^2_{\mathcal{H}_{m_H}} \\
\displaystyle\qquad + C \sum_{i=1}^{n} \Big[1 - y_i\Big(\sum_{m=1}^{M_H} f_m(\boldsymbol{x}_i) + b\Big)\Big]_+ \\
\displaystyle\text{s. t.} \quad \sum_{m=1}^{M_h} \sigma_{h,m} \le 1\,, \quad \forall h \in \{1, \dots, H\} \\
\qquad \sigma_{h,m} \ge 0\,, \ \forall (h,m) \in \{1, \dots, H\} \times \{1, \dots, M_h\}\,,
\end{cases}
$$

where the parameters p_h introduced here allow to control the sparsity at level h. The choice of these free parameters is discussed below.

The objective function of Problem (3) explicitely relies on the weights $\sigma_{h,m}$ that define the effective kernel. For analysis purposes, it is interesting to reformulate this objective function in terms of a mixed norm on functions f_m, so as to exhibit the penalty applied at each level of the tree structure.

Proposition 1. *From the optimality conditions of $\boldsymbol{\sigma}$, the objective function of Problem* (3) *reads:*

$$
\frac{1}{2}\Bigg(\sum_{m_1}\Big(\sum_{m_2}\dots\big(\sum_{m_H}\|f_{m_H}\|^{\gamma_H}_{\mathcal{H}_{m_H}}\big)^{\frac{\gamma_{H-1}}{\gamma_H}}\dots\Big)^{\frac{\gamma_1}{\gamma_2}}\Bigg)^{\frac{2}{\gamma_1}}
+ C\sum_{i=1}^{n}\Big[1 - y_i\big(\sum_{m=1}^{M_H} f_m(\boldsymbol{x}_i) + b\big)\Big]_+ \,, \qquad (4)
$$

with $\gamma_k = 2\Big(1 + \displaystyle\sum_{h=k}^{H} p_h\Big)^{-1}$.

This nested mixed norm contrasts with the CAP-like penalty (2): when CAP relies on nested groups, levels correspond to penalty strengths, whereas we use levels to shape the penalty structure.

The convexity and sparsity properties related to the mixed norm in (4) are well-known:

- function (4) is convex if and only if $\gamma_h \ge 1 \ \forall h \in \{1, \dots, H\}$;
- the minimizers of (4) are expected to be sparse at height h if and only if $\gamma_h \le 1$.

Hence $\gamma_h = 1 \ \forall h \in \{1, \dots, H\}$ is required to have a convex problem whose solution will be sparse at each level. In this case, KEOPS reduces de facto to MKL and the group structure plays no role. Convexity, sparsity and effective group-structure can all concur when the latter aims at ensuring the joint selection of the elements of groups at some levels. As a result, we will consider non-convex structured penalties in order to encourage sparsity at different levels.

3. ALGORITHM

The direct minimization of (4) is difficult even for MKL [8], which is the simplest possible case we may consider, with $H = 1$. We thus resort to Problem (3), whose resolution follows the simpleMKL scheme, with two nested problems. The outer problem defines the optimal effective kernel as follows:

$$
(5)\quad \begin{cases}
\displaystyle\min_{\boldsymbol{\sigma}} \ J(\boldsymbol{\sigma}) \\
\displaystyle\text{s. t.} \quad \sum_{m=1}^{M_h} \sigma_{h,m} \le 1\,, \quad \forall h \in \{1, \dots, H\} \\
\qquad \sigma_{h,m} \ge 0\,, \ \forall (h,m) \in \{1, \dots, H\} \times \{1, \dots, M_h\}\ ,
\end{cases}
$$

where $J(\boldsymbol{\sigma})$ is the optimal value of the objective function of a standard SVM problem, with a kernel set to the effective kernel defined by $\boldsymbol{\sigma}$. This SVM problem defines the inner problem:

$$J(\boldsymbol{\sigma}) = \min_{f_1,\dots,f_{M_H},\, b} \ \frac{1}{2} \sum_{m_1} \frac{1}{\sigma_{1,m_1}{}^{p_1}} \ \cdots \sum_{m_H} \frac{1}{\sigma_{H,m_H}{}^{p_H}} \, \|f_{m_H}\|^2_{\mathcal{H}_{m_H}}$$

$$+ C \sum_{i=1}^{n} \left[1 - y_i \Big(\sum_{m=1}^{M_H} f_m(\boldsymbol{x}_i) + b \Big) \right]_+ . \quad (6)$$

The inner problem (6) solves Problem (3) with respect to $\{f_m\}$ and b, for fixed $\boldsymbol{\sigma}$ parameters, thereby defining the value function $J(\boldsymbol{\sigma})$ for the outer problem. The outer problem (5) optimize Problem (3) with respect to the weights $\boldsymbol{\sigma}$ for fixed $\{f_m\}$ and b values.

Problem (6) is a standard SVM problem, while Problem (5) is solved exactly from the optimality conditions of $\boldsymbol{\sigma}$. For lack of space, we limit the detailed exposure to a hierarchy with 3 levels. [2]

$$\begin{aligned}
\sigma_{1,m_1} &= c \times (s_{m_1})^{\frac{\gamma_1}{\gamma_2}} \\
\sigma_{2,m_2} &= c \times (s_{m_1})^{-\frac{p_1\gamma_1}{2}} \times (s_{m_2})^{\frac{\gamma_2}{\gamma_3}} \\
\sigma_{3,m_3} &= c \times (s_{m_1})^{-\frac{p_1\gamma_1}{2}} \times (s_{m_2})^{-\frac{p_2\gamma_2}{2}} \times \|f_{m_3}\|^{\gamma_3}_{\mathcal{H}_{m_3}} \ ,
\end{aligned}$$

where $s_{m_2} = \sum_{m_3} \|f_{m_3}\|^{\gamma_3}_{\mathcal{H}_{m_3}}$, $s_{m_1} = \sum_{m_2} (s_{m_2})^{\frac{\gamma_2}{\gamma_3}}$ and $c = \Big(\sum_{m_1} (s_{m_1})^{\frac{\gamma_1}{\gamma_2}} \Big)^{-1}$. The overall procedure is summarized below.

Algorithm 1: KEOPS

```
initialize σ ;
repeat
    solve the SVM problem → J(σ) ;
    for h = {1, ..., H} and m = {1, ..., M_h} do
        update σ_{h,m}; // according to
        // optimality conditions of (3)
until convergence;
```

4. EXPERIMENTS

Dataset. This experiment comes from Brain-Computer Interface (BCI) and deals with single trial classification of EEG signals. We use the dataset from the BCI 2003 competition for the task of interfacing the P300 Speller [14]. The 7560 EEG signals, recorded from 64 electrodes (or channels) and paired with positive or negative visual stimuli responses, are processed as in [15]. It leads to 7560 examples of dimension 896 with 14 time frames for each of the 64 channels. The 896 features extracted from the EEG signals are not transformed before classification and are used as linear kernels.

Structure. We consider a tree structure of 3 levels with brain regions at the top level, channels at the intermediate level and time frames at the leaf level. Sparsity is expected at each level. The channels are grouped into different cerebral cortex areas to encourage localisation in the brain functional regions. Indeed, some regions may be more involved than others to solve a task related to a paradigm. In particular, the strongest activity for the P300 Speller is expected to occur over the parietal brain area [16]. Furthermore, an automated channel selection for each single subject is of primary importance for BCI real-life applications since it makes the acquisition system easier to use and to set-up and may lead to better performances [17]. Finally, the most salient frames for the P300 Speller are expected to be centered around 300 ms which corresponds to frames 7 and 8, so that feature selection may be also carried out within each channel to eliminate irrelevants frames. Therefore, we have to learn different coefficients $\{\boldsymbol{\sigma}_h\}_{h=1}^3$ according to $M_3 = 896$ frames divided into $M_2 = 64$ channels organized into $M_1 = 17$ regions.

Methods. We aim at classifying the EEG trials with as few channels and time frames as possible. To induce a sparse solution through the different levels, we test a non-convex parametrization of KEOPS, which corresponds to a $\ell_{(1/2;2/3;1)}$ penalty, by setting $p_h = 1, \forall h$. We compare our approach to MKL and SPAMS which implements a classification method for linear models with tree structured penalties as those presented in Section 1 [11]. Note that SPAMS has been tested with all the penalties available though the reported results only concern the $\ell_{(1;2)}$ mixed norm that achieves the best performance.

Protocol. We have randomly picked 567 training examples from the datasets and used the remaining as testing examples. Using a small part of the examples for training can be justified by the use of ensemble of SVM (not considered here) on a latter stage of the EEG classification procedure [15]. The hyperparameter C has been selected by 5-fold cross validation. The performance is measured by the AUC. This overall procedure has been repeated 10 times.

Numerical results. Table 1 reports the average AUC for KEOPS, SPAMS and MKL together with the number of regions, channels and frames selected. The prediction performances are similar for the 3 methods, with a slight advantage for KEOPS. Regarding sparsity, KEOPS has a clear edge at all levels, with much less features involved than SPAMS or MKL. In terms of brain areas, KEOPS focusses on half of the regions while SPAMS needs more than three quarters of them. MKL, which does not take any structure into account, keeps all the regions. At the channels level, KEOPS is still extremely sparse, retaining less than a quarter of the electrodes whereas MKL and SPAMS solutions require almost three quarters of them. Finally, KEOPS keeps at the very most a tenth of the frames and is two times sparser than MKL and six times sparser than SPAMS despite the ℓ_1 norm applied to the frames level (each frame has been considered as a group in SPAMS).

Graphical results. Figure 2 represents the median relevance computed over the 10 experiments at the different levels. The results for KEOPS are particularly neat and show the highest relevances in the parietal lobe with the lateral electrodes PO_7, PO_8 and P_8. The primary motor and somatosensory cortices are also significantly involved with the central electrodes FC_Z, C_Z, and CP_Z as well as the left part of the temporal lobe with the electrode T_9. The maps for MKL and SPAMS identify the importance of the same regions and channels but also highlight numerous frontal electrodes that are not likely to be relevant for the BCI P300 Speller. At the frames level, all the methods show a slow rise with a sudden peak at frames 7 and 8 followed by a slow decline, with a more drastic elimination process regarding the frames outside this bandwidth for KEOPS.

5. CONCLUSION

KEOPS is at the crossroad of kernel learning and structured feature selection. It extends the MKL framework to encode nested groups of similarities in a tree structure allowing flexible formulations that transcribe richer assumptions about the solution. This behavior is illustrated in a BCI problem where KEOPS reaches the prediction performances of the competing approaches with much less features providing interpretable solutions at different scales. A further improvement in our approach would be to introduce penalties that can encourage sparsity according to some kind of neighborhood such as in [18] or [19]. Indeed, regarding the BCI application for instance, such penalties could induce some persistency between contiguous regions or time frames.

[2] These expressions, obtained after tedious algebra from the first-order optimality conditions of Problem (3), are available upon request.

Method	AUC	# Regions	# Channels	# Frames
KEOPS	85.8 ± 1.2	8.3 ± 1.8	13.1 ± 2.9	62.3 ± 17.5
SPAMS	84.7 ± 0.7	14.5 ± 2.1	47.5 ± 11.9	394.3 ± 264.6
MKL	85.5 ± 0.9	16.3 ± 0.5	49.7 ± 7.6	139.7 ± 41.2

Table 1: Average results and standard deviations for KEOPS, SPAMS and MKL on the BCI P300 Speller dataset.

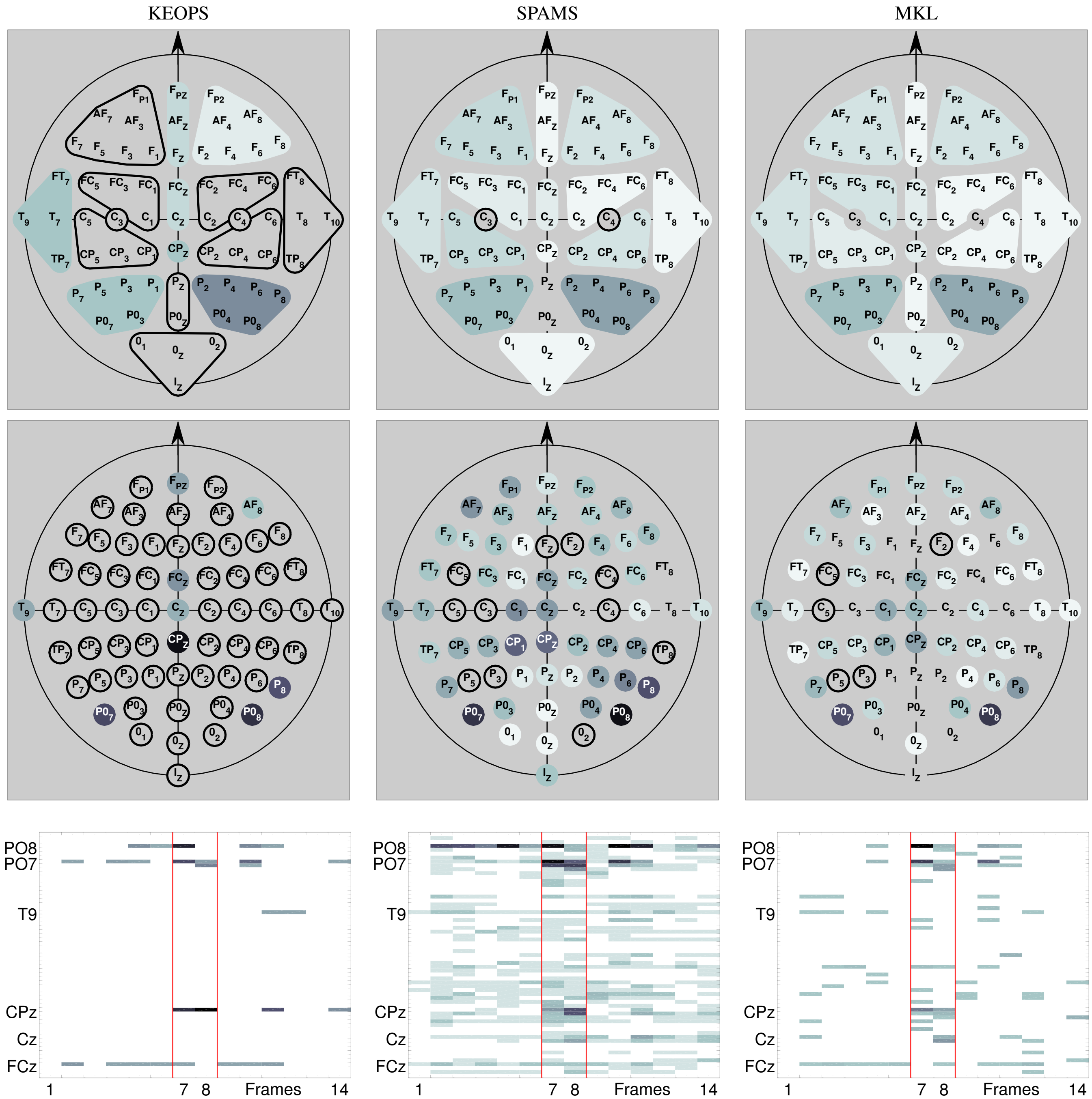

Fig. 2: Median relevance at the regions (top), channels (middle) and frames (bottom) levels for KEOPS, SPAMS and MKL. The darker the color, the higher the relevance. Regions and electrodes with no color and a black boundary as well as frames in white are discarded (the relevance is exactly zero). At each level, a normalization factor has been applied to set the sum of relevances to one. At the frames levels, the channel sequence starting from the bottom is the following: FC_5, FC_3, FC_1, FC_Z, FC_2, FC_4, FC_6, C_5, C_3, C_1, C_Z, C_2, C_4, C_6, CP_5, CP_3, CP_1, CP_Z, CP_2, CP_4, CP_6, FP_1, FP_Z, FP_2, AF_7, AF_3, AF_Z, AF_4, AF_8, F_7, F_5, F_3, F_1, F_Z, F_2, F_4, F_6, F_8, FT_7, FT_8, T_7, T_8, T_9, T_{10} , TP_7, TP_8, P_7, P_5, P_3, P_1, P_Z, P_2, P_4, P_6, P_8, PO_7, PO_3, PO_Z, PO_4, PO_8, O_1, O_Z, O_2, I_Z.

6. REFERENCES

[1] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and K. Kandola, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*, 2002, pp. 367–373.

[2] N. Cristianini, C. Campbell, and J. Shawe-Taylor, "Dynamically adapting kernels in support vector machines," in *Advances in Neural Information Processing Systems 11*, 1999, pp. 204–210.

[3] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature selection for SVMs," in *Advances in Neural Information Processing Systems 13*, 2001, pp. 668–674.

[4] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, pp. 131–159, 2002.

[5] Y. Grandvalet and S. Canu, "Adaptive scaling for feature selection in SVMs," in *Advances in Neural Information Processing Systems 15*, 2003, pp. 553–560.

[6] A. Argyriou, R. Hauser, C. A. Micchelli, and M. Pontil, "A dc-programming algorithm for kernel selection," in *Proceedings of the twenty-third International Conference on Machine Learning*, 2006, pp. 41–48.

[7] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semi-definite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[8] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[9] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, pp. 303–324, 2009.

[10] P. Zhao, G. Rocha, and B. Yu, "The Composite Absolute Penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, pp. 3468–3497, 2009.

[11] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *Journal of Machine Learning Research*, vol. 12, pp. 2297–2334, 2011.

[12] F. Bach, "Exploring large feature spaces with hierarchical multiple kernel learning," in *Advances in Neural Information Processing Systems 21*, 2009, pp. 105–112.

[13] M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy, "Composite kernel learning," *Machine Learning*, vol. 79, no. 1-2, pp. 73–103, 2010.

[14] B. Blankertz, K.-R. Müller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, and N. Birbaumer, "The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials," *IEEE Trans. Biomed. Eng*, vol. 51, pp. 1044–1051, 2004.

[15] A. Rakotomamonjy, V. Guigue, G. Mallet, and V. Alvarado, "Ensemble of SVMs for improving brain-computer interface P300 speller performances," in *15th International Conference on Artificial Neural Networks*, 2005, vol. 3696, pp. 45–50.

[16] J. Polich, "Updating P300: an integrative theory of P3a and P3b," *Neurophysiology*, vol. 10, pp. 2128–2148, 2007.

[17] M. Schröder, T. N. Lal, T. Hinterberger, M. Bogdan, J. Hill, N. Birbaumer, W. Rosenstiel, and B. Schölkopf, "Robust EEG channel selection across subjects for brain computer interfaces," *EURASIP Journal on Applied Signal Processing*, vol. 19, pp. 3103–3112, 2005.

[18] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society Series B*, vol. 67, pp. 91–108, 2005.

[19] M. Kowalski, K. Siedenburg, and M. Dörfler, "Social sparsity! neighborhood systems enrich structured shrinkage operators," *IEEE Transactions on Signal Processing*, vol. 61, pp. 2498–2511, 2013.