Multiple Subject Learning for Inter-Subject Prediction

Sylvain Takerkart, Liva Ralaivola

Multiple Subject Learning for Inter-Subject Prediction

Sylvain Takerkart*† and Liva Ralaivola†
*Institut de Neurosciences de la Timone UMR 7289, Aix Marseille Université, CNRS, Marseille, France
†Laboratoire d'Informatique Fondamentale UMR 7279, Aix Marseille Université, CNRS, Marseille, France

***Abstract*—Multi-voxel pattern analysis has become an important tool for neuroimaging data analysis by allowing to predict a behavioral variable from the imaging patterns. However, standard models do not take into account the differences that can exist between subjects, so that they perform poorly in the inter-subject prediction task. We here introduce a model called *Multiple Subject Learning* (MSL) that is designed to effectively combine the information provided by fMRI data from several subjects; in a first stage, a weighting of single-subject kernels is learnt using multiple kernel learning to produce a classifier; then, a data shuffling procedure allows to build ensembles of such classifiers, which are then combined by a majority vote. We show that MSL outperforms other models in the inter-subject prediction task and we discuss the empirical behavior of this new model.**

I. Introduction

Background. In recent years, the use of machine learning approaches in neuroimaging has gained in popularity. The most prominent application of machine learning is the so-called *multi-voxel pattern analysis* (MVPA), that consists in predicting a behavioral variable from functional MRI data. The appeal of these multivariate methods relies on their increased sensitivity compared to standard univariate models. However, their generalization power on data recorded in new subjects suffers from the large variability that exists within a population.

Contribution. We specifically focus on the so-called *inter-subject prediction* problem and we propose a method that is precisely aimed at giving reliable predictions for data of any subject for which no data was accessed to during the learning process. Our method builds upon a hierarchical probabilistic setting, and it makes use of two well-known tools from the machine learning artillery to deal with heterogeneous and variance-inducing data : multiple kernel learning (MKL [3], [5], [8]), and data sampling/shuffling [2], [12].

Related work. The standard MVPA paradigm to address the inter-subject prediction task consists in pooling together all samples from the subjects available at training time. A single classifier is estimated in a supervised manner on this dataset to be later tested on data from new subjects. This paradigm, which is used by default in the literature, largely ignores the various sources of inter-subject variability, namely the differences in anatomy and functional organization across subjects, and the fact that all samples are not drawn from the same probability distribution. Therefore there is a crucial need for more elaborate models specifically tuned for the inter-subject prediction task, like those recently proposed in [6], [7], [11] and the new model we introduce in this paper.

II. Methods

A. Setting and Addressed Problems

Training dataset. We consider the following setting. At training time, we have access to data from S subjects, indexed by $s \in \mathcal{S} \doteq \{1 \dots S\}$. For each subject s, we are provided with a training labeled training dataset $D_s \doteq \{(x_n^s, y_n^s)\}_{n=1}^N$, where each pair (x_n^s, y_n^s) is made of an observation vector x_n^s, assumed to be an element of $\mathcal{X} \doteq \mathbb{R}^F$, and a label/category y_n^s assumed to be an element of $\mathcal{C} \doteq \{1, \dots, C\}$. The whole training set is denoted as

$$D \doteq \cup_{s=1}^S D_s.$$

We additionally assume that each subject has participated in exactly the same fMRI protocol, so that $\forall n \in \{1 \dots N\}, y_n^1 = y_n^2 = \dots = y_n^S$ and, in what follows, y_n will be used to denote the category associated with the n-th sample. We will use $\mathbf{y}$ as a compact notation for the vector $\mathbf{y} \doteq [y_1 \cdots y_N] \in \mathcal{C}^N$.

Probabilistic setting. A *hierarchical* —hierarchical, because distributions on distributions are considered— probabilistic setting that may be associated with the generation of D is as follows. There is an unkown and fixed distribution $\mathcal{L}^s$ defined on the space of distributions on $\mathcal{X} \times \mathcal{C}$. A realization $\mathcal{L}^o_{|s} : \mathcal{X} \times \mathcal{C} \to \mathbb{R}$ of a random variable distributed according to $\mathcal{L}^s$ is the unknown and fixed distribution associated with subject s which governs training set D_s. More precisely, if $\mathcal{L}^o_{|s;y} : \mathcal{X} \to \mathbb{R}$ is the conditional law defined by

$$\mathcal{L}^o_{|s;y}(\cdot) \doteq \mathcal{L}^o_{|s}(\cdot, y),$$

then, D_s is a random sample distributed according to

$$D_s \sim \bigotimes_{n=1}^N \mathcal{L}^o_{|s;y_n}. \quad (1)$$

The probabilistic setting we have just described allows us to say that training sample D (given $\mathbf{y}$) is so that:

$$D \sim \bigotimes_{s=1}^S \left[\mathcal{L}^s \bigotimes_{n=1}^N \mathcal{L}^o_{|s;y_n} \right],$$

and the core random pair $(s, (X^s, Y^s))$ that defines our learning problem is therefore distributed according to[1]

$$\mathcal{L} \doteq \mathcal{L}^s \otimes \mathcal{L}^o_{|s}. \quad (2)$$

Addressed problems. The main problem that we address is then to *learn* from D a predictor $f : \mathcal{X} \to \mathcal{C}$ with risk

$$R(f) \doteq \mathbb{P}_{(s,(X^s,Y^s))\sim\mathcal{L}}(f(X^s) \neq Y^s)$$

[1]Note that the work in [1], theoretically shows how learning from data sampled according to a distribution conditioned on $\mathbf{y}$ provides results regarding learning from the corresponding unconditional distribution.

as small as possible. It is essential to understand that the *inter-subject* nature of the problem stems from s being an independent random variable (of law $\mathcal{L}^s$): being able to make reliable predictions for subjects not seen during training is ultimately the goal conveyed by the minimization of $R(f)$. This general problem may be tackled by considering two subproblems: i) that of identifying, among the training subjects, those that provide the most representative behaviors so as to more heavily rely on them in the learning of f and, ii) that of reducing the variance of the learned predictor that may be due to the scarceness of data and the natural inter-subject variability of the data [4]. In II-C, we propose a learning model to address both subproblems (and thus, the general problem) which echoes the hierarchical decomposition of $\mathcal{L}$ as in (2).

B. Simple Strategies for Inter-Subject Prediction with SVC

We here briefly recall naive strategies to deal with learning from data coming from various subjects. In order to lighten the exposition, we now consider that $\mathcal{C}$ is reduced to the set $\mathcal{C} = \{-1, +1\}$ of two labels. A real-valued function $f : \mathcal{X} \to \mathbb{R}$ is readily associated with the thresholded decision function $x \mapsto \text{sign}(f(x))$, which predicts the label of x according to the sign, -1 or $+1$, of $f(x)$; with a slight abuse, any function f will also denote its associated thresholded predictor. The specific class $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}}$ of functions that we will consider throughout, is that of *kernel-based* functions: we consider that we have at hand some positive definite kernel (see [10]) $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and that $\mathcal{F}$ is defined as

$$\mathcal{F} \doteq \left\{ f : f(\cdot) = \sum_{i=1}^{m} \alpha_i k(x_i, \cdot), (x_1, \ldots, x_m) \in \mathcal{X}^m \right\}.$$

The core predictor that we will use are Support Vector Classifiers (SVC). If trained on D^s, an SVC f^s writes as:

$$f^s(x) = \sum_{n=1}^{N} \alpha_n^s k(x_n^s, x), \tag{3}$$

where the kernel used are centered on the training data x_n^s. However, this is intuitive that if a classifier is aimed at the inter-subject prediction task it should take advantage of all available data of D. This can be achieved by combining single subject classifiers $(f_s)_{s=1}^{s=S}$ through a vote, to give classifier

$$f^{\texttt{vote}}(x) = \sum_{s=1}^{S} \beta_s f^s(x) = \sum_{s=1}^{S} \beta_s \sum_{n=1}^{N} \alpha_n^s k(x_n^s, x), \tag{4}$$

where the $(\beta_s)_{s=1}^{s=S}$ is a sequence of (nonnegative) voting weights. Among the various ways to learn those weights, we will simply use the most trivial with: $\beta_1 = \ldots = \beta_S = 1/S$.

Another strategy that uses all the data from D consists in *pooling* all the training samples together, i.e assuming that they are independent realizations of the same fixed random variable. This way, an SVC is learned on the whole training set D, with no regards to the hierarchical probabilistic decomposition discussed previously, and in particular, no distinction between data from different subjects. The SVC classifier $f^{\texttt{pool}}$ obtained with such an approach takes the form

$$f^{\texttt{pool}}(x) = \sum_{n=1}^{N} \sum_{s=1}^{S} \alpha_n^s k(x_n^s, x). \tag{5}$$

Note that if the form of $f^{\texttt{pool}}$ encompasses that of $f^{\texttt{vote}}$ (see (4)), which is recovered for $\alpha_n^s = \beta_n \alpha^s$, the loss of the hierarchical structure suggested by the probabilistic model is here lost, as both indices s and n play the same role.

C. Multiple Subject Learning

We now describe the approaches that we promote: *Multiple Subject Learning* (MSL) which, as indicated by its name, is built upon the idea of Multiple Kernel Learning [5].

MSL. When using data from all available subjects to train a classifier, it is implicitely hoped that the subjects are a representative sample of the distribution $\mathcal{L}^s$ of subjects. However, it seems intuitive that some subjects in D might contribute in a stronger manner to a good generalization, i.e. a small risk. Weighting the contribution of each subject in the learning process is therefore a natural way to account for this belief, as already mentioned previously, when the voting scheme was discussed (see $f^{\texttt{vote}}$). Here, we take advantage of the fact that the prediction functions under consideration are kernel-based classifiers to have the weighting intervene at the kernel level, and we propose to look for a classifier $f^{\texttt{msl}}$ of the form

$$f^{\texttt{msl}}(x) = \sum_{n=1}^{N} \alpha_n \sum_{s=1}^{S} \beta_s k(x_n^s, x), \tag{6}$$

where the sequence $(\beta_s)_{1 \leq s \leq S}$ of *nonnegative* coefficients, which weighs the contributions of data from the different subjects, has to be learned from the training examples D together with the sequence $(\alpha_n)_{1 \leq n \leq N}$. Introducing the following notation will help us make a clearer connection with MKL: i) $\underline{x}_n \doteq [x_n^1 \cdots x_n^S] \in \mathcal{X}^S$ denotes the vector made of the concatenation of the x_n^s's, ii) likewise, for $x \in \mathcal{X}$, $\underline{x} \in \mathcal{X}^S$ is the vector $\underline{x} \doteq [x \cdots x]$ of S concatenations of x, iii) $\Pi^s : \mathcal{X}^S \to \mathcal{X}$ is the orthogonal projector such that $\Pi^s \underline{x}$ extracts the s-th block (of size F —recall that $\mathcal{X} = \mathbb{R}^F$) of coordinates from $\underline{x}$, so that $\Pi^s \underline{x}_n = x_n^s$ and $\Pi^s \underline{x} = x$, and iv) $k^s : \mathcal{X}^S \times \mathcal{X}^S \to \mathbb{R}$ is the positive definite kernel such that $k^s(\underline{x}, \underline{x}') \doteq k(\Pi^s \underline{x}, \Pi^s \underline{x}')$. Classifier $f^{\texttt{msl}}$ of (6) can now be rewritten as:

$$f^{\texttt{msl}}(x) = \sum_{n=1}^{N} \alpha_n \sum_{s=1}^{S} \beta_s k^s(\underline{x}_n, \underline{x}) \doteq \underline{f}^{\texttt{msl}}(\underline{x}) \tag{7}$$

(where $\underline{f}^{\beta} \in \mathbb{R}^{\mathcal{X}^S}$), or,

$$f^{\texttt{msl}}(x) = \sum_{n=1}^{N} \alpha_n K^{\beta}(\underline{x}_n, \underline{x}),$$

if $K^{\beta} : \mathcal{X}^S \times \mathcal{X}^S \to \mathbb{R}$ is the positive kernel defined by $K^{\beta} \doteq \sum_{s=1}^{S} \beta_s k^s$ (K^{β} is positive definite because it is a nonnegative combination of positive kernels).

Given model (7), the problem we face is therefore to both learn the coefficients $(\alpha_n)_{1 \leq n \leq N}$ of a kernel classifier *and* the weights $(\beta_s)_{1 \leq s \leq S}$ of a relevant combination of kernels: this is exactly the problem of Multiple Kernel Learning (see [5]). Here, each kernel k^s to be combined is specifically associated to subject s, hence the name of *Multiple Subject Learning*.

Ensembles of MSL. We observe from (7) and, in fact, from the definition of $\underline{x}_n$, that we have grouped together the data

$x_n^1, \ldots, x_n^S$. However, standard fMRI protocols are designed to measure repetitions of responses that are stationary in time, such that, from a probabilistic point of view x_n^s and $x_{n'}^s$ are identically distributed provided that $y_n = y_{n'}$ (see how D_s is distributed according to (1)). The association of $x_n^1, \ldots, x_n^S$ into $\underline{x}_n$ we assumed so far is therefore arbitrary and several other matchings are just as valid as long as they group together vectors $x_{n_1}^1, \ldots, x_{n_N}^S$ such that $y_{n_1} = \ldots = y_{n_N}$. We may take advantage of that remark to build *ensembles of MSL classifiers* that get rid of this arbitrary grouping *and* make it possible to lower the variance of the predictor learned. The strategy to do so is based on the use of *permutations*, which allow various groupings of the data that, in turn, give various MSL predictors. Given such ensemble of M MSL classifiers $(f_m^{\texttt{msl}})_{m=1}^{m=M}$, we may compute their combined prediction by

$$f^{\texttt{msl}^*}(x) = \sum_{m=1}^{M} f_m^{\texttt{msl}}(x), \tag{8}$$

Here, the learning of $f_m^{\texttt{msl}}$ is the result of i) a random and uniform draw of S permutations[2] $\sigma_1^m, \ldots \sigma_S^m$ over $\{1, \ldots, N\}$ such that for any n, $y_{\sigma_1^m(n)} = \ldots = y_{\sigma_S^m(n)} = y_n$ and ii) the solving of MSL problem (7) with, for $n = 1, \ldots, N$, the vector $\underline{x}_n \in \mathcal{X}^S$ defined as $\underline{x}_n = [x_{\sigma_1^m(n)}^1 \cdots x_{\sigma_S^m(n)}^S]$, which is where some shuffling of the training data occurs. As announced earlier, the strategy to learn $f^{\texttt{msl}^*}$ better echoes the probabilistic decomposition discussed above.

III. Experiments

A. Dataset

To evaluate our MSL framework, we analyzed data from an fMRI experiment during which nine subjects listened to auditory stimuli centered around five frequencies while performing tapping with a finger of the left hand, matched to the audio frequency, thus defining five classes of trials. The acquisitions comprised a high resolution T1, as well as EPIs (TR = 2.4s, voxel size = 2x2x3mm) recorded during five functional sessions; each session included six trials per condition presented in a pseudo-randomized order.

The functional data was analyzed in *SPM8*, with motion and slice timing corrections, followed by a GLM comprising one regressor per trial. The corresponding beta maps served as estimates of the response size for each trial. *Freesurfer* was then used to perform cortical reconstruction and registration to a spherical atlas. Each beta map was projected onto this atlas, thus providing a vertex to vertex mapping across subjects. Three cortical regions of interest were defined on the atlas to respectively cover the right and left auditory cortices and the right somato-sensory cortex.

The goal of the *inter-subject prediction* task was to guess the class of the stimulus from the response pattern (chance level = 0.2). We examined the different strategies that were defined in Section II i) the SVC classifiers learned on single-subject data (defined in eq. (3) and hereafter called s-SVC); ii) the classifier obtained by a majority vote on the set of s-SVCs (eq. (4), hereafter vote-SVC); iii) the SVC learned by pooling data from all subjects (eq. (5), hereafter pool-SVC); iv) the multiple subject learning classifiers (eq. (6) and eq. (7), hereafter MSL and MSL*). The algorithms were run with the linear kernel and evaluated with a leave-one-subject-out cross-validation. For the first three strategies to perform at their best, a univariate feature selection was performed before learning the classifier; the percentage of selected features and the regularization constant C were chosen in a nested cross-validation scheme. MSL and MSL* were estimated with C = 1. on the full feature set, using the l_2 norm implementation of MKL available in the *Shogun* toolbox.

[2]A permutation over $\{1, \ldots, N\}$ is a bijection from $\{1, \ldots, N\}$ to itself.

B. Results

For each region, 500 MSL classifiers were learnt, each from a random set of within-subject shufflings. To generate an MSL* classifier of size M, we randomly drew M MSL classifiers from these 500. For each value of M, we generated 100 MSL* classifiers and averaged the results.

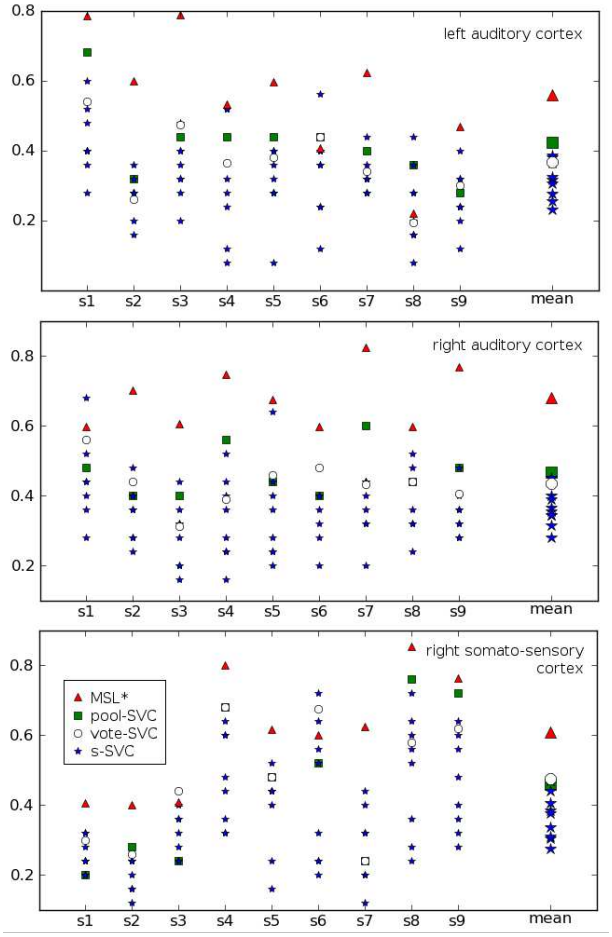

Fig. 1. Comparing classifiers. Accuracy rates on test subject $s1, ..., s9$, or averaged by cross validation (right column). Chance level is 0.2.

1) Comparing learning strategies: Fig. 1 shows the accuracy rate of each classifier type, averaged on all folds of the cross-validation as well as on each single fold.

First, we qualitatively analyse these results. We observe that i) all classifiers perform very differently across folds of the cross-validation; ii) within each fold (i.e each column in Fig. 1), a large variation of the performances of the s-SVCs is also present; and iii) in most cases, at least one of the s-SVCs

outperformed pool-SVC and vote-SVC. These qualitative observations respectively confirm our initial intuitions that i) the inter-subject variability weights heavily on the results; ii) some subjects will be more informative than others to generalize over the population; and iii) the pooling strategy vastly used in the litterature is clearly sub-optimal.

Secondly, we quantitatively assess the performances of the different classifiers by performing t-tests with paired samples corresponding to the performances of classifiers in each fold of the cross-validation. There was no significant differences between vote-SVC and pool-SVC classifiers ($p > 0.2$ for all three regions). Our MSL* classifiers (with $M = 50$) outperformed the other strategies in all cases ($p < 0.05$ for all three regions), which supports the effectiveness of this model.

2) Influence of the size of MSL ensembles: We further examined the influence of the size of the MSL* ensembles. The results are summarized on Fig. 2, in which each color corresponds to the analysis conducted in one of the three ROIs, and the x axis of all subplots corresponds to varying size M of the MSL* ensembles. Fig. 2.A presents the MSL* accuracy

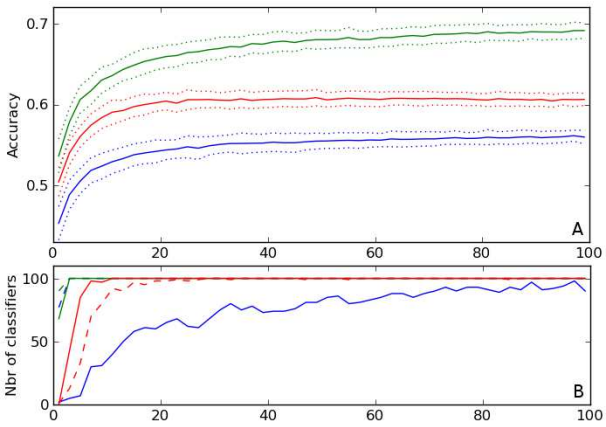

Fig. 2. Influence of the ensemble size M (x axis) on A: the accuracy of MSL*; and B: the proportion of MSL* classifiers that outperform pool-SVC (solid lines) and vote-SVC (dashed lines). Each color refers to one ROI.

rates. One can clearly see the accuracy gain offered by the construction of such ensembles and that their performances reaches an asymptotic maximum when M is of the order of a few dozens. The curves on Fig. 2.B represent the number of MSL* classifiers (amongst the 100 generated for each M) for which the performance was significantly better (paired t-test, $p < 0.05$) than vote-SVC (dashed lines) and pool-SVC (solid lines). Those curves quickly reach 100% (except for the solid blue line), which means that the gain of accuracy offered by MSL* is robust to the choice of the ensemble itself.

IV. Discussion and Conclusion

We have presented several models that combine the information from multiple subjects to learn a classifier aimed at performing predictions on data from subjects not available at training time. We have demonstrated that our *multiple subject learning* framework makes it possible to i) combine information from different subjects in a *multiple kernel learning* fashion, and ii) build ensembles of classifiers by shuffling the ordering of examples in each subjects. The resulting MSL* classifiers vastly outperform other strategies in the inter-subject prediction task in an fMRI experiment designed to study the organization of the auditory and somato-sensory cortices.

Future work will focus on the interpretation of the results provided by MSL. Beyond the use of the outcome of the predictions, our MSL framework provides weights for each subject. We plan on studying their consistency, both across folds of the cross-validation and across the different MSL classifiers used in a MSL* ensemble. Furthermore, similarly to the use of MKL weights, the MSL weights could further be used to detect which subjects contribute the most strongly in such classification task, or even to perform subject selection/rejection: on can imagine setting up a recursive subject elimination by removing the subject with the smallest weight at each iteration to maximize the generalization performance.

We also plan on attempting to improve the construction of the MSL* ensembles to further increase the performances of our framework. Indeed, using a purely random selection of classifiers is known to result in sub-optimal ensembles, in the sense that one can build an equally efficient ensemble of smaller size by appropriately choosing the classifiers. This could be done by maximising the information diversity that is provided by each classifiers of the ensemble [9].

Acknowledgment

Thanks to the CNRS Neuro-IC program for funding, and to the *Centre IRMf de Marseille* and its staff for data acquisition.

References

[1] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth, "Generalization Bounds for the Area under the ROC Curve," *Journal of Machine Learning Research*, vol. 6, pp. 393–425, 2005.

[2] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[3] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "Lp-norm multiple kernel learning," *The Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.

[4] L. I. Kuncheva, J. J. Rodriguez, C. O. Plumpton, D. E. J. Linden, and S. J. Johnston, "Random subspace ensembles for fMRI classification," *IEEE Trans. on Medical Imaging*, vol. 29, no. 2, pp. 531–542, 2010.

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

[6] A. Lorbert and P. J. Ramadge, "Kernel hyperalignment." in *NIPS*, 2012, pp. 1799–1807.

[7] A. F. Marquand, M. Brammer, S. C. Williams, and O. M. Doyle, "Bayesian multi-task learning for decoding multi-subject neuroimaging data," *NeuroImage*, vol. 92, pp. 298 – 311, 2014.

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

[9] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, vol. 6, no. 1, pp. 63–81, 2005.

[10] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[11] S. Takerkart, G. Auzias, B. Thirion, D. Schon, and L. Ralaivola, "Graph-based inter-subject classification of local fmri patterns," in *Machine Learning in Medical Imaging*, LNCS 2012, vol. 7588, pp. 184–192.

[12] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.