Chapter 1

Output Kernel Learning Methods

Francesco Dinuzzo
IBM Research, Dublin, Ireland
francesd@ie.ibm.com

Cheng Soon Ong
NICTA, Canberra, Australia
cheng-soon.ong@nicta.com.au

Kenji Fukumizu
The Institute of Statistical Mathematics, Tachikawa, Tokyo, Japan
fukumizu@ism.ac.jp

Simultaneously solving multiple related estimation tasks is a problem known as *multi-task learning* in the machine learning literature. A rather flexible approach to multi-task learning consists in solving a regularization problem where a positive semidefinite *multi-task kernel* is used to model joint relationships between both inputs and tasks. Specifying an appropriate multi-task kernel in advance is not always possible, therefore it is often desirable to estimate one from the data. In this chapter, we overview a family of regularization techniques called Output Kernel Learning (OKL), for learning a multi-task kernel that can be decomposed as the product of a kernel on the inputs and one on the task indices. The kernel on the task indices is optimized simultaneously with the predictive function by solving a joint two-level regularization problem.

1.1 Learning Multi-Task Kernels

Supervised multi-task learning consists in estimating multiple functions $f_j : \mathcal{X}_j \to \mathcal{Y}$ from multiple datasets of input-output pairs

$$(x_{ij}, y_{ij}) \in \mathcal{X}_j \times \mathcal{Y}, \quad j = 1, \dots, m, \quad i = 1, \dots, \ell_j,$$

where m is the number of *tasks* and ℓ_j is the number of data pairs for the j-th task. In general, the input sets $\mathcal{X}_j$ and the output set $\mathcal{Y}$ can be arbitrary non-empty sets. If the input sets $\mathcal{X}_j$ are the same for all the tasks, i.e. $\mathcal{X}_j = \mathcal{X}$, and the power set $\mathcal{Y}^m$ can be given a vector space structure, one can equivalently think in terms of learning a single vector-valued function $f : \mathcal{X} \to \mathcal{Y}^m$ from a dataset of pairs with incomplete output data. The key point in multi-task learning is to exploit relationships between the different components f_j in order to improve performance with respect to solving each supervised learning problem independently.

For a broad class of multi-task (or multi-output) learning problems, a suitable positive semidefinite *multi-task kernel* can be used to specify the joint relationships between inputs and tasks [5]. The most general way to address this problem is to specify a similarity function of the form $K((x_1, i), (x_2, j))$ defined for every pair of input data (x_1, x_2) and every pair of task indices (i, j). In the context of a kernel-based regularization method, choosing a multi-task kernel amounts to designing a suitable Reproducing Kernel Hilbert Space (RKHS) of vector-valued functions, over which the function f whose components are the different tasks f_j is searched. See [13] for details about the theory of RKHS of vector valued-functions.

Predictive performances of kernel-based regularization methods are highly influenced by the choice of the kernel function. Such influence is especially evident in the case of multi-task learning where, in addition to specifying input similarities, it is crucial to correctly model inter-task relationships. Designing the kernel allows to incorporate domain knowledge by properly constraining the function class over which the solution is searched. Unfortunately, in many problems the available knowledge is not sufficient to uniquely determine a good kernel in advance, making it highly desirable to have data-driven automatic selection tools. This need has motivated a fruitful research stream which has led to the development of a variety of techniques for learning the kernel.

There is considerable flexibility in choosing the similarity function K, the only constraint being positive semidefiniteness of the resulting kernel. However, such flexibility may also be a problem in practice, since choosing a good multi-task kernel for a given problem may be difficult. A very common way to simplify such modeling is to utilize a multiplicative decomposition of the form

$$K((x_1, i), (x_2, j)) = K_X(x_1, x_2) K_Y(i, j),$$

where the *input kernel* K_X is decoupled from the *output kernel* K_Y. The same structure can be equivalently represented in terms of a matrix-valued kernel

$$H(x_1, x_2) = K_X(x_1, x_2) \cdot \mathbf{L}, \tag{1.1}$$

where $\mathbf{L}$ is a positive semidefinite matrix with entries $\mathbf{L}_{ij} = K_Y(i, j)$. Since specifying the kernel function K_Y is completely equivalent to specifying the matrix $\mathbf{L}$, we will use the term *output kernel* to denote both of them, with a slight abuse of terminology.

Even after imposing such simplified model, specifying the inter-task similarities in advance is typically impractical. Indeed, it is often the case that multiple learning tasks are known to be related, but no precise information about the structure or the intensity of such relationships is available. Simply fixing $\mathbf{L}$ to the identity, which amounts to share no information between the tasks, is clearly suboptimal in most of the cases. On the other hand, wrongly specifying the entries may lead to a severe performance degradation. It is therefore clear that, whenever the task relationships are subject to uncertainty, learning them from the data is the only meaningful way to proceed.

1.1.1 Multiple Kernel Learning

The most studied approach to automatic kernel selection, known as Multiple Kernel Learning (MKL), consists in learning a conic combination of N basis kernels of the form

$$K = \sum_{k=1}^{N} d_k K_k, \qquad d_k \geq 0, \qquad k = 1, \ldots, N.$$

Appealing properties of MKL methods include the ability to perform selection of a subset of kernels via sparsity, and tractability of the associated optimization problem, typically (re)formulated as a convex program. Although most of the works on MKL focus on learning similarity measured between inputs, the approach can be clearly also used to learn a multi-task kernel of the form

$$K((x_1, i), (x_2, j)) = \sum_{k=1}^{N} d_k K_X^k(x_1, x_2) K_Y^k(i, j),$$

which includes the possibility of optimizing the matrix $\mathbf{L}$ in (1.1) as a conic combination of basis matrices, by simply choosing the input kernels K_X^k to be equal. In principle, proper complexity control allows to combine an arbitrarily large, even infinite [1], number of kernels. However, computational and memory constraints force the user to specify a relatively small dictionary of basis kernels to be combined, which again calls for a certain amount of domain knowledge. Examples of works that employ a MKL approach to address multi-output or multi-task learning problems include [17, 11, 16].

1.1.2 Output Kernel Learning

A more direct approach to learn inter-task similarities from the data consists in searching the output kernel K_Y over the whole cone of positive semidefinite kernels, by optimizing a suitable objective functional. Equivalently, the corresponding matrix $\mathbf{L}$ can be searched over the cone of positive semidefinite matrices.

This can be accomplished by solving a two-level regularization problem of the form

$$\min_{\mathbf{L}\in\mathbb{S}_+} \min_{f\in\mathcal{H}_\mathbf{L}} \left(\sum_{j=1}^{m} \sum_{i=1}^{\ell_j} V(y_{ij}, f_j(x_{ij})) + \lambda \left(\|f\|^2_{\mathcal{H}_\mathbf{L}} + \Omega(\mathbf{L}) \right) \right), \tag{1.2}$$

where (x_{ij}, y_{ij}) are input-output data pairs for the j-th task, V is a suitable loss function, $\mathcal{H}_\mathbf{L}$ is the RKHS of vector-valued functions associated with the reproducing kernel (1.1), Ω is a suitable matrix regularizer, and $\mathbb{S}_+$ is the cone of symmetric and positive semidefinite matrices. The regularization parameter $\lambda > 0$ should be properly selected in order to achieve a good trade-off between approximation of the training data and regularization. This can be achieved by hold-out validation, cross-validation, or other methods. We call such an approach Output Kernel Learning (OKL). By virtue of a suitable representer theorem [13], the inner regularization problem in (1.2) can be shown to admit solutions of the form

$$\hat{f}_k(x) = \sum_{j=1}^{m} \mathbf{L}_{kj} \left(\sum_{i=1}^{\ell_j} c_{ij} K_X(x_{ij}, x) \right), \tag{1.3}$$

under mild hypothesis on V.

From the expression (1.3), we can clearly see that when $\mathbf{L}$ equals the identity, the external sum decouples and each optimal function $\hat{f}_k$ only depends on the corresponding dataset (*independent single task-learning*). On the other hand, when all the entries of the matrix $\mathbf{L}$ are equal, all the functions $\hat{f}_k$ are the same (*pooled single task-learning*). Finally, whenever $\mathbf{L}$ differs from the identity, the datasets from multiple tasks get mixed together and contribute to the estimates of other tasks.

1.1.2.1 Frobenius Norm Output Kernel Learning

A first OKL technique was introduced in [4] for the case where V is a square loss function, Ω is the squared Frobenius norm, and the input data x_{ij} are the same for all the output components f_j, leading to a problem of the form

$$\min_{\mathbf{L}\in\mathbb{S}_+} \min_{f\in\mathcal{H}_\mathbf{L}} \left(\sum_{i=1}^{\ell} (y_i - f_j(x_i))^2 + \lambda \left(\|f\|^2_{\mathcal{H}_\mathbf{L}} + \|\mathbf{L}\|^2_F \right) \right), \tag{1.4}$$

Such special structure of the objective functional allows to develop an effective block coordinate descent strategy where each step involves the solution of a

Sylvester linear matrix equation. A simple and effective computational scheme to solve (1.4) is described in [4]. Regularizing with the squared Frobenius norm ensures that the sub-problem with respect to $\mathbf{L}$ is well-posed. However, one may want to encourage different types of structures for the output kernel matrix, depending on the application.

1.1.2.2 Low-Rank Output Kernel Learning

When the output kernel is low-rank, the estimated vector-valued function maps into a low-dimensional subspace. Encouraging such low-rank structure is of interest in several problems. Along this line, [3, 2] introduce low-rank OKL, a method to discover relevant low dimensional subspaces of the output space by learning a low-rank kernel matrix. This method corresponds to regularizing the output kernel with a combination of the trace and a rank indicator function, namely

$$\Omega(\mathbf{L}) = \mathrm{tr}(\mathbf{L}) + I(\mathrm{rank}(\mathbf{L}) \leq p).$$

For $p = m$, the hard-rank constraint disappears and Ω reduces to the trace, which still encourages low-rank solutions. Setting $p < m$ gives up convexity of the regularizer but, on the other hand, allows to set a hard bound on the rank of the output kernel, which can be useful for both computational and interpretative reasons. The optimization problem associated with low-rank OKL is the following:

$$\min_{\mathbf{L} \in \mathbb{S}_+} \min_{f \in \mathcal{H}_{\mathbf{L}}} \left(\sum_{j=1}^{m} \sum_{i=1}^{\ell_j} (y_{ij} - f_j(x_{ij}))^2 + \lambda \left(\|f\|^2_{\mathcal{H}_{\mathbf{L}}} + \mathrm{tr}(\mathbf{L}) \right) \right), \text{ s.t. } \mathrm{rank}(\mathbf{L}) \leq p. \tag{1.5}$$

The optimal output kernel matrix can be factorized as $\mathbf{L} = \mathbf{B}\mathbf{B}^T$, where the horizontal dimension of $\mathbf{B}$ is equal to the rank parameter p. Problem (1.5) exhibits several interesting properties and interpretations. Just as sparse MKL with a square loss can be seen as a nonlinear generalization of (grouped) Lasso, low-rank OKL is a natural kernel-based generalization of reduced-rank regression, a popular multivariate technique in statistics [9]. When $p = m$ and the input kernel is linear, low-rank OKL reduces to multiple least squares regression with nuclear norm regularization. Connections with reduced-rank regression and nuclear norm regularization are analyzed in [3].

For problems where the inputs x_{ij} are the same for all the tasks, optimization for low-rank OKL can be performed by means of a rather effective procedure that iteratively computes eigendecompositions, see Algorithm 1 in [3]. Importantly, the size of the involved matrices such as $\mathbf{B}$, the low rank factor of $\mathbf{L}$, can be controlled by selecting the parameter p. However, more general multi-task learning problems where each task is sampled in correspondence with different inputs require completely different methods. It turns out that an effective strategy to approach the problem consists in iteratively applying inexact Preconditioned Conjugate Gradient (PCG) solvers to suitable linear operator equations that arise from the optimality conditions. Such linear operator equations are derived and analyzed in [2].

1.1.2.3 Sparse Output Kernel Learning

In many multitask learning problems it is known that some of the tasks might be related while some others are independent, but it is unknown in advance which of the tasks are related. In such cases, it may make sense trying to encourage sparsity in the output kernel by means of suitable regularization. For instance, by choosing an entry-wise ℓ_1 norm regularization $\Omega(\mathbf{L}) = \|\mathbf{L}\|_1$, one obtains the problem

$$\min_{\mathbf{L}\in\mathbb{S}_+} \min_{f\in\mathcal{H}_\mathbf{L}} \left(\sum_{j=1}^{m} \sum_{i=1}^{\ell_j} (y_{ij} - f_j(x_{ij}))^2 + \lambda \left(\|f\|^2_{\mathcal{H}_\mathbf{L}} + \|\mathbf{L}\|_1 \right) \right).$$

Encouraging a sparse output kernel may allow to automatically discover clusters of related tasks. However, some of the tasks may be already known in advance to be unrelated. Such information can be encoded by also enforcing a hard constraint on the entries of the output kernel, for instance by means of the regularizer $\Omega(\mathbf{L}) = \|\mathbf{L}\|_1 + I(P_S(\mathbf{L}) = 0)$, where I is a indicator function, P_S selects a subset S of the non-diagonal entries of $\mathbf{L}$ and projects them into a vector, yielding the additional constraint

$$\mathbf{L}_{ij} = 0, \qquad \forall (i,j) \in S.$$

The subproblem with respect to $\mathbf{L}$ is a convex nondifferentiable problem, also when hard sparsity constraints are present. Effective solvers for sparse output kernel learning problems are currently under investigation.

1.2 Applications

Multi-task learning problems where it is important to estimate the relationships between tasks are ubiquitous. In this section, we provide examples of such problems where OKL techniques have been applied successfully.

1.2.1 Collaborative Filtering And Preference Estimation

Estimating preferences of several users for a set of items is a typical instance of multi-task learning problem where each task is the preference function of one of the users, and exploiting similarities between the tasks matters. Preference estimation is a key problem addressed by collaborative filtering systems and recommender systems, that find wide applicability on the web.

In the context of collaborative filtering , techniques such as low-rank matrix approximation are considered state of the art. In the following, we present some results from a study based on the MovieLens datasets (see Table 1.1), three popular collaborative filtering benchmarks containing collections of ratings in the range $\{1, \ldots, 5\}$ assigned by several users to a set of movies, for more details, see [2]. The study shows that, by exploiting additional information about the inputs (movies), OKL techniques are superior to plain low-rank matrix approximation.

TABLE 1.1: `MovieLens` datasets: total number of users, movies, and ratings.

Dataset	Users	Movies	Ratings
`MovieLens100K`	943	1682	10^5
`MovieLens1M`	6040	3706	10^6
`MovieLens10M`	69878	10677	10^7

The results reported in Table 1.2 correspond to a setup where a random test set is extracted, containing about the 50% of the ratings for each user, see also [15, 10]. Results under different test settings are also available, see [2]. The 25% of the remaining training data are used as a validation set to tune the regularization parameter. Performance is evaluated according to the root mean squared error (RMSE) on the test set. Regularized matrix factorization (RMF) corresponds to choosing the input kernel equal to $K_X(x_1, x_2) = \delta_K(x_1, x_2)$, where δ_K denotes the Kronecker delta (non-zero only when the two arguments are equal), so that no information other than the movie Id is exploited to express the similarity between the movies. The pooled and independent baselines correspond to choosing $\mathbf{L}_{ij} = 1$ and $\mathbf{L}_{ij} = \delta_K(i, j)$, respectively. The last method employed is low-rank OKL with rank parameter $p = 5$ fixed a priori for all three datasets, and input kernel designed as

$$K(x_1, x_2) = \delta_K(x_1^{id}, x_2^{id}) + \exp\left(-d_H(x_1^g, x_2^g)\right),$$

by taking into account movie Ids x_1^{id}, x_2^{id} and meta-data about genre categorization of the movies x_1^g, x_2^g available in all three datasets.

TABLE 1.2: `MovieLens` datasets: test RMSE for low-rank OKL, RMF, pooled and independent single-task learning.

Dataset	RMF	Pooled	Independent	OKL
`MovieLens100K`	1.0300	1.0209	1.0445	**0.9557**
`MovieLens1M`	0.9023	0.9811	1.0297	**0.8945**
`MovieLens10M`	0.8627	0.9441	0.9721	**0.8501**

1.2.2 Structure Discovery in Multiclass Classification

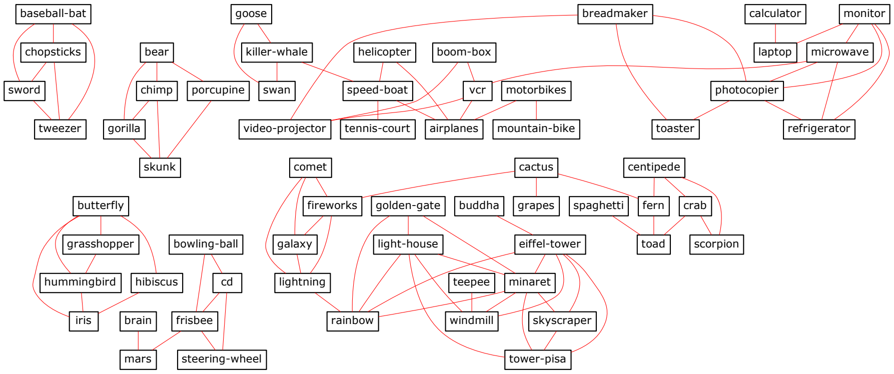

FIGURE 1.1: Caltech 256: learned similarities between classes. Only a subset of the classes is shown.

Multi-class classification problems can be also seen as particular instances of multi-task learning where each real-valued task function f_j corresponds to a score for a given class. The training labels can be converted into sparse real vectors of length equal to the number of classes, and only one component different from zero. Employing a OKL method in this context allows not only to train a multi-class classifier, but also to learn the similarities between the classes.

As an example, Figure 1.1 shows a visualization of the entries of output kernel matrix obtained by applying low-rank OKL to the popular Caltech 256 dataset [6, 7], containing images of several different categories of objects, including buildings, animals, tools, etc. By using 30 training examples for each class, the obtained classification accuracy on the test set (0.44) is close to state of the art results. At the same time, the graph obtained by thresholding the entries of the learned output kernel matrix with low absolute value, reveals clusters of classes that are meaningful and agree with common sense. Output kernel learning methods have been also applied in [8] to solve object recognition problems.

1.2.3 Pharmacological problems

Multi-task learning problems are common in pharmacology, where data from multiple subjects are available. Due to the scarcity of data for each subject, it is often crucial to combine the information from different datasets in order to obtain a good estimation performance. Such combination needs to take into account the similarities between the subjects, while allowing for enough flexibility to estimate personalized models for each of them. Output kernel learning methods have been successfully applied to pharmacological

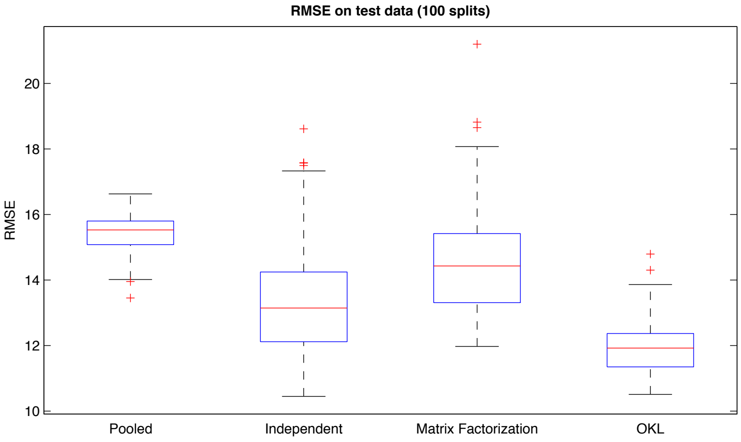

FIGURE 1.2: Experiment on pharmacokinetic data [2]. Root Mean Squared Error averaged over 100 random splits for the 27 subject profiles in correspondence with different methods.

problems in [2], where two different problems are analyzed. Both problems can be seen as multi-task regression problems or matrix completion problems with side information.

The first problem consists in filling a matrix of drug concentration measurements for 27 subjects in correspondence with 8 different time instants after the drug administration, by having access to only 3 measurements per subject. Standard low-rank matrix completion techniques are not able to solve this problem satisfactorily, since they ignore the available knowledge about the temporal shape of the concentration curves. On the other hand, a OKL method allows to easily incorporate such knowledge by designing a suitable input kernel that takes into account temporal correlation, as done in [2]. Figure 1.2 reports boxplots over the 27 subjects of the root mean squared error, averaged over 100 random selections of the three training measurements, showing a clear advantage of the OKL methodology with respect to both pooled and independent baselines, as well as a low-rank matrix completion technique that does not use side information.

The second problem analyzed in [2] has to do with completing a matrix of Hamilton Depression Rating Scale (HAMD) scores for 494 subjects in correspondence with 7 subsequent weeks, for which only a subset of 2855 entries is available [12]. Performance is evaluated by keeping 1012 properly selected entries for test purposes. In order to automatically select the regularization parameter λ, a further splitting of the remaining data is performed to obtain a validation set containing about 30% of the examples. Such splitting is performed randomly and repeated 50 times. By employing a low-rank OKL approach with a simple linear spline input kernel, one can observe significantly better results (Table 1.3) with respect to low-rank matrix completion and standard baselines, see [2] for further details.

TABLE 1.3: Drug efficacy assessment experiment [2]: best average RMSE on test data (and their standard deviation over 50 splits)

Pooled	Independent	RMF	OKL
6.86 (0.02)	6.72(0.16)	6.66(0.4)	**5.37**(0.2)

1.3 Concluding remarks and future directions

Learning output kernels via regularization is an effective way to solve multi-task learning problems where the relationships between the tasks are uncertain or unknown. The OKL framework that we have discussed in this chapter is rather general and can be further developed in various directions. There are several practically meaningful constraints that could be imposed on the output kernel: sparsity patterns, hierarchies, groupings, etc. Effective optimization techniques for more general (non-quadratic) loss functions are still lacking and the use of a variety of matrix penalties for the output kernel matrix is yet to be explored. Extensions to semi-supervised and online problems are needed in order to broaden applicability of these techniques. Finally, some hybrid methods that combine learning of, possibly multiple, input and output kernels have been recently investigated [14] and are currently still under active investigation.

Bibliography

[1] A. Argyriou, C. A. Micchelli, and M. Pontil. Learning convex combinations of continuously parameterized basic kernels. In Peter Auer and Ron Meir, editors, *Learning Theory*, volume 3559 of *Lecture Notes in Computer Science*, pages 338–352. Springer Berlin / Heidelberg, 2005.

[2] F. Dinuzzo. Learning output kernels for multi-task problems. *Neurocomputing*, 118:119–126, 2013.

[3] F. Dinuzzo and K. Fukumizu. Learning low-rank output kernels. *Journal of Machine Learning Research - Proceedings Track*, 20:181–196, 2011.

[4] F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto. Learning output kernels with block coordinate descent. In *Proceedings of the 28th Annual International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

[5] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[6] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR*, page 178, 2004.

[7] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, Caltech, 2007.

[8] Z. Guo and Z. J. Wang. Cross-domain object recognition by output kernel learning. In *Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on*, pages 372–377. IEEE, 2012.

[9] A. J. Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975.

[10] M. Jaggi and M. Sulovský. A simple algorithm for nuclear norm regularized problems. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 471–478, Haifa, Israel, June 2010. Omnipress.

[11] H. Kadri, A. Rakotomamonjy, F. Bach, and P. Preux. Multiple operator-valued kernel learning. In *NIPS*, pages 2438–2446, 2012.

[12] E. Merlo-Pich and R. Gomeni. Model-based approach and signal detection theory to evaluate the performance of recruitment centers in clinical trials with antidepressant drugs. *Clinical Pharmacology and Therapeutics*, 84:378–384, September 2008.

[13] C. A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.

[14] V. Sindhwani, H.Q. Minh, and A. C. Lozano. Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and granger causality. In *UAI*, 2013.

[15] K Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Optimization Online*, 2009.

[16] C. Widmer, N. C. Toussaint, Y. Altun, and G. Rätsch. Inferring latent task structure for multitask learning by multiple kernel learning. *BMC bioinformatics*, 11(Suppl 8):S5, 2010.

[17] A. Zien and C. S. Ong. Multiclass multiple kernel learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1191–1198, 2007.

Index