Noname manuscript No.
(will be inserted by the editor)

Regularization-based Multitask Learning

With applications to Genome Biology and Biological Imaging

Christian Widmer[1] · Marius Kloft[2] · Xinghua Lou · Gunnar Rätsch

Received: date / Accepted: date

Abstract The aim of multitask learning is to improve the generalization performance of a set of related tasks by exploiting complementary information about the tasks. In this paper, we review established approaches for regularization based Multitask Learning, sketch some recent developments, and demonstrate their applications in Computational Biology and Biological Imaging.

Keywords Multitask Learning · Transfer Learning · Regularized Risk Minimization

1 Introduction

Multitask learning (MTL) is a machine learning technique that considers learning across multiple tasks that are possibly related to each other. Inspired by models of learning in human brain, multitask learning first appeared in the context of Neural Networks [4,5].

In this paper, we specifically address the following three major research questions:

- How can we integrate the information contained in multiple related tasks into the learning process in order to obtain better predictors?
- How can we capture the similarity between tasks and best incorporate this into a learning framework?
- What are examples of applications in Genome Biology and Biological Imaging that profit from learning from multiple tasks simultaneously?

A strong motivation to study MTL in the context of biology stems from complex, coupled inference tasks. For instance, in order to decipher biological processes, biologists often aim at bringing together knowledge that has been obtained in *multiple* experiments, e.g., each experiment performed on a different organism. A challenge here is to take into account the fact that some organisms are more closely related to each other than others (cf. Figure 1). In order to form effective prediction models, several recent MTL learning machines try to respect the differences between these organisms, while exploiting similarities.

From a top level perspective, this is achieved by learning the classifiers for the tasks in a joint learning problem, such that the classifiers of the particular tasks are coupled. This coupling is usually achieved by promoting similar tasks to yields similar prediction models, where the strength of the coupling strongly depends on how closely related the tasks are.

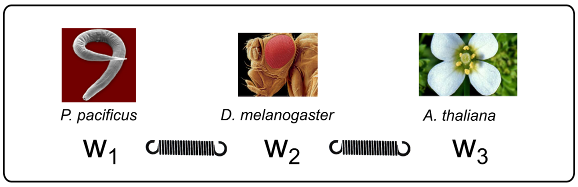

Fig. 1 Examples of biological organism with varying similarity. The associated learning models are denoted by w_t, $t = 1, 2, 3$. The organism shown to the left and right are marginally related, while both of them are closely related to the one shown in the center.

In Section 3, we give a detailed example of a multitask model in the spirit of Figure 1, taken from the application domain of the genome biology. But the range of application scenarios goes beyond the organisms-as-task scenario. For instance, we may want to learn a

All authors are with Computational Biology Center, Memorial Sloan-Kettering Cancer Center, 415 E 68th Street, New York City, NY 10065, USA. E-mail: {widmerc,kloftm,loux,ratschg}@mskcc.org

[1] CW is also with the Machine Learning Group, Technische Universität Berlin, Marchstr. 23, 10587 Berlin, Germany.
[2] MK is also with the Courant Institute of Mathematical Sciences, 251 Mercer St, New York, NY 10012, USA.

model for a biochemical mechanism from several tissues, cell lines - or in the case of cancer biology - tumor types, all of which could be treated as different tasks in a MTL setting.

Another application example is shown in Section 4, where we show how the principle of coupling models can also be extended to applications in Biological Imaging. We show that MTL can be successfully applied to leverage 2D images to facilitate the analysis of 3D images.

2 Multitask Learning

In this section we describe the problem setting of multi-task learning. We also present particular instances of multi-task learning machines, focusing on formulations that are appealing for computational biology. For a detailed overview, see the survey of [14].

2.1 Regularization-based Multitask Learning

From a historical perspective, regularization-based MTL is based on regularized risk minimization [17] and supervised learning methods such as the Support Vector Machine (SVM) [3,6] or Logistic Regression. In regularized risk minimization, we aim at computing a model $\boldsymbol{w}$ by minimizing an objective $J(\boldsymbol{w})$ consisting of a loss-term L that captures the error with respect to the training data (X, Y) and a regularizer $\boldsymbol{\Omega}$ that penalizes model complexity:

$$J(\boldsymbol{w}) = L(\boldsymbol{w}|X, Y) + \boldsymbol{\Omega}(\boldsymbol{w}).$$

This formulation can easily be generalized to the MTL setting, where we are interested in obtaining several models parametrized by $\boldsymbol{w}_1, ..., \boldsymbol{w}_T$, where T is the number of tasks. The above formulation can be extended by introducing an additional regularization term $\boldsymbol{\Omega}_{\mathsf{MTL}}$ that penalizes the discrepancy between individual models:

$$J(\boldsymbol{w}_1, ..., \boldsymbol{w}_T) = \sum_{t=1}^{T} J(\boldsymbol{w}_t) + \boldsymbol{\Omega}_{\mathsf{MTL}}(\boldsymbol{w}_1, ..., \boldsymbol{w}_T).$$

2.1.1 Common Approaches

In the following, we denote the training examples by (x_i, y_i), $i = 1, \dots, n$, each of which is associated with a task $\tau(i) \in \{1, \dots, T\}$. We denote the set of indices of training points of the tth task by $I_t := \{i \in \{1, \dots, n\} : \tau(i) = t\}$ and their number by $n_t := \#I_t$. One of the first works on regularization-based MTL is by Evgeniou and Pontil [9], where at optimization time all parameter vectors are "pulled" towards their average $\bar{\boldsymbol{w}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{w}_t$,

$$\boldsymbol{\Omega}_{\mathsf{MTL}}(\boldsymbol{w}_1, ..., \boldsymbol{w}_T) = \frac{1}{2} \left|\left|\boldsymbol{w}_t - \bar{\boldsymbol{w}}\right|\right|^2.$$

Note that all tasks are treated equally in the above formulation; however, often we are given the priori information that some tasks are more related to each other than the remaining ones. To penalize the differences between the parameter vectors accordingly, the above setting was extended by [8],

$$\boldsymbol{\Omega}_{\mathsf{MTL}}(\boldsymbol{w}_1, ..., \boldsymbol{w}_T) = \frac{1}{2} \sum_{s=1}^{T} \sum_{t=1}^{T} A_{st} \left|\left|\boldsymbol{w}_s - \boldsymbol{w}_t\right|\right|^2.$$

where the graph adjacency matrix $A = (A_{st})$, captures the task similarities. We can rewrite the above formulation using the graph Laplacian $L = (L_{st})$,

$$\boldsymbol{\Omega}_{\mathsf{MTL}}(\boldsymbol{w}_1, ..., \boldsymbol{w}_T) = \frac{1}{2} \sum_{s=1}^{T} \sum_{t=1}^{T} L_{st} \boldsymbol{w}_s^T \boldsymbol{w}_t,$$

where $L = D - A$, where $D_{s,t} = \delta_{s,t} \sum_k A_{s,k}$. Finally, it can be shown that this gives rise to the following *Multitask* kernel in the respective dual form:

$$K((x, s), (z, t)) = H_{st}^{+} \cdot K_B(x, z),$$

where K_B is a kernel defined on examples and $H^+ = (H_{st}^+)$ denotes the pseudo inverse of $H := I + L$, where I is the identity matrix. A closely related formulation was successfully used in the context of Computational Biology by [10], where a kernel on tasks K_T is used instead of the pseudo-inverse, giving rise to

$$K((x, s), (z, t)) = K_T(s, t) \cdot K_B(x, z). \tag{1}$$

Note that the corresponding joint feature space between task t and feature vector x can be written as a tensor product $\phi(t, x) = \phi_T(t) \cdot \phi_B(x)$ [10]. A special case of (1) is studied in [7] in the context of Domain Adaptation, where $\phi_T(t) = (1, 1, 0)$ was used as the source task descriptor and $\phi_T(t) = (1, 0, 1)$ for the target task, corresponding to $K_T(s, t) = (1 + \delta_{s,t})$.

3 Application in Computational Biology

In this section, we show an application of MTL to the recognition of splice sites – an important problem in genome biology. By now it is well understood and there exist experimentally confirmed labels for a broad range of organisms. In previous work, we have investigated how well information can be transferred between source and target organisms in different evolutionary distances (i.e. one-to-many) and training set sizes [15]. We identified MTL algorithms that are particularly well suited for this task. In a follow-up project we investigated how our results generalize to the MTL scenario (i.e. many-to-many) and showed that exploiting prior information about task similarity provided by taxonomy can be very valuable [18]. An example how MTL can improve performance compared to baseline methods *individual* (i.e. learn a classifier for each task independently) and *union* (i.e. pool examples from all tasks and obtain a global classifier) is given in Figure 2. The figure shows results for 6 out of 15 organisms for the baseline methods *individual* and *union* and the MTL algorithm described in Section 2.1. The mean performance is shown in the last column. For each task, we obtained 10000 training examples and an additional test set of 5000 examples. We normalized the data sets such that there are 100 negative examples per positive example. We report the area under the precision recall curve (auPRC), which is an appropriate measure for unbalanced classification problems (i.e. detection problems). For an elaborate discussion of our experiments with splice site prediction, please consider the original publications [15, 18].

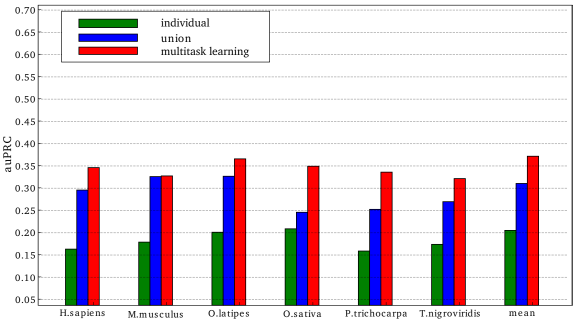

Fig. 2 Results of the RNA splicing experiment comparing *MTL* to baseline methods *individual* and *union*. Figure taken from [18].

4 Application to Biological Imaging

Here, we briefly describe another application of MTL to Biological Imaging. In this example, we jointly learn prediction models for well annotated 2D data and crude 3D data. The goal is to transfer the knowledge learned from the 2D data to regularize the 3D model such that the 3D model is trained robustly even with very limited annotations.

Current biological research is exhibiting a significant trend towards using 3D imaging techniques to monitor complex biological activities at the molecular and cellular level. This has catalyzed the emergence of a new field which is referred to as bioimage informatics, which aims at advancing image analysis to cope with the increasing complexity and quantity of data from 3D imaging. Despite the prevailing employment of 3D imaging and the heavy focus on advancing 3D image analysis, a question is raised: *should we just ignore all the available 2D data?* In this application, we show that early 2D data can be used to facilitate the processing of the emerging 3D data using MTL by treating 2D and 3D as two tasks.

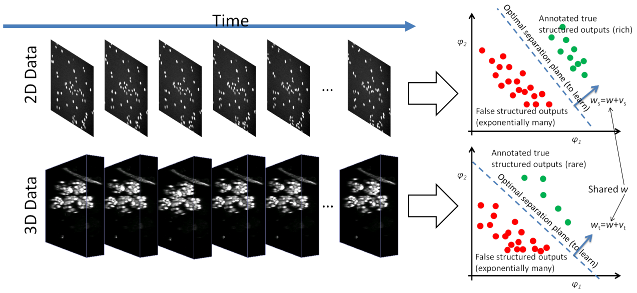
Time
2D Data
3D Data
...
φ_2
φ_1
Optimal separation plane (to learn)
Annotated true structured outputs (rich)
False structured outputs (exponentially many)
$w_s=w+v_s$
Annotated true structured outputs (rare)
Shared w
$w_t=w+v_t$

Fig. 3 2D and 3D image sequences exhibit different yet cognate distributions in the joint feature space. Prediction models share a component and are trained jointly.

In practice, the early 2D data is well studied and contains rich annotation, but the fresh 3D dataset is crude and manually annotating it is particularly exhausting and time-consuming. Therefore, we want to train a high-quality prediction model for the 3D data using as little annotation as possible. A concrete example is shown in Fig. 3 in the context of cell tracking from time-lapse experiments. These two experiments capture similar biological processes (cell movement, division, etc.), yielding cognate distributions of tracking features (we adopted the segmentation in [13] and the tracking features described in [11]). However, due to differences in the underlying biological entities and the experiment conditions, they also exhibit a certain degree of variations (Fig. 3, right). To explore the connection and also capture the difference, we try to learn prediction models for 2D and 3D jointly, which share a common component $\boldsymbol{w}$ but also have distinct, domain-dependent parts (i.e., $\boldsymbol{w}_s - \boldsymbol{w}$ and $\boldsymbol{w}_t - \boldsymbol{w}$, the deviation of domain-specific parameters to their shared base).

Since images are intrinsically structured data, the models we discuss here are structured prediction models [16]. We can extend the model described in Section 2.1 to the domain of structured prediction as follows:

$$\min_{\boldsymbol{w},\boldsymbol{w}_{\mathrm{s}},\boldsymbol{w}_{\mathrm{t}}} \frac{\lambda_1}{2}\|\boldsymbol{w}-\boldsymbol{w}_{\mathrm{s}}\|^2 + \frac{\lambda_1}{2}\|\boldsymbol{w}-\boldsymbol{w}_{\mathrm{t}}\|^2 + \frac{\lambda_2}{2}\|\boldsymbol{w}\|^2$$
$$+\boldsymbol{\Omega}_{\mathrm{s}}(\boldsymbol{w}_{\mathrm{s}}) + \boldsymbol{\Omega}_{\mathrm{t}}(\boldsymbol{w}_{\mathrm{t}}),$$

Table 1 Comparison among several training settings. The baseline is a model trained using all fully annotated 3D data – the most expensive one in terms of annotation cost. Training on 2D data only, though being the cheapest approach, yields a significant increase of errors (39.8% w.r.t. the baseline). The same applies to training on partially annotated 3D data (21.9% increased error w.r.t. the baseline (much cheaper, though). By combining the datasets from these two approaches into a *MTL* setting, we reduced the relative error to only 3.6% at the same annotation cost.

Methods	Test loss	Relative to baseline
Trained on fully annotated 3D data (as baseline)	3.69%	0.0%
Trained on fully annotated 2D data	5.16%	+39.8%
Trained on partially (25%) annotated 3D data	4.41% ± 0.33	+21.88%±9.0%
Jointly trained on 2D (full) and 3D (partial, 25%)	3.82% ± 0.09	+3.59%±2.57%

where $\boldsymbol{\Omega}_\mathrm{s}(\boldsymbol{w}_\mathrm{s})$ and $\boldsymbol{\Omega}_\mathrm{t}(\boldsymbol{w}_\mathrm{t})$ are the empirical loss from the source and target domain, respectively. The two hyper parameters λ_1 and λ_2 have two uses: firstly, they control the regularization to avoid over-fitting; secondly, the ratio $\frac{\lambda_1}{\lambda_2}$ controls the similarity between the two domains (higher means more related) [9]. We assume that data is completely annotated in the source domain (2D). Therefore, $\boldsymbol{\Omega}_\mathrm{s}(\boldsymbol{w}_\mathrm{s})$ is the convex hinge loss for structured prediction [16]. On the other hand, we only require partial annotations for the target domain (3D). Accordingly, the "best" full annotation has to be inferred during the training, which leads to the bridge loss introduced in [12]. The resulting objective function is a convex-concave function and we adopt the fast CCCP procedure [12] to solve it. For more details, we refer the users to [12].

The 2D data in our experiment were acquired using a Nikon's TE2000 inverted microscope while the 3D data were acquired using a confocal microscope by Zeiss. For more details, we again refer to [12]. Results are shown in Table 1. Our baseline is a model trained using fully annotated 3D data, which is of course very expensive in terms of annotation efforts. This baseline approach yields a test loss of 3.69% (first row), which is significant better than simply training on the 2D data (second row). To address this issue, we tried 25% partial annotation which brings apparent improvement yet not sufficient (third row). By jointly learning on 2D and 3D data (fourth row), we bring the relative error down to 3.59% ± 2.57%.

5 Conclusion and Outlook

We have presented a brief overview of regularization-based MTL methods that allow the joint learning from several tasks, with applications in Genome Biology and Biological Imaging, where joint learning was helpful. Especially in the context of biomedical data, where generating training labels can be very expensive, MTL learning can be viewed as an appealing means to obtain more cost-effective predictors.

We have shown how to incorporate task similarities into the learning framework by means of graph-regularizers. In this context, a great challenge in MTL is how to obtain meaningful task similarities. In some applications this can be provided by domain experts or is naturally available, for instance, in form of a taxonomy. Recently, advances were made to address the question of how the similarity or relatedness of tasks can be learned from data directly. While we did not cover this aspect of transfer learning in this paper, we would like to stress that this question is of central importance when applying MTL methods in practice and point out the methods by [2, 21, 1] and our own work [20, 19].

Acknowledgements We thank Klaus-Robert Müller and Mehryar Mohri for inspiring and helpful discussions. This work was supported by the German Research Foundation (DFG) under MU 987/6-1 and RA 1894/1-1 as well as by the European Community's 7th Framework Programme under the PASCAL2 Network of Excellence (ICT-216886). Marius Kloft acknowledges a postdoctoral fellowship by the German Research Foundation (DFG). We also thank Anna-Katerina Hadjantonakis and Minjung Kang for the 3D image data.

References

1. G. Blanchard, G. Lee, and C. Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems 24*, 2011.
2. E. Bonilla, K.M. Chai, and C. Williams. Multi-task Gaussian Process Prediction. *Advances in Neural Information Processing Systems*, 20(October):153–160, 2008.
3. B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.
4. R. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, pages 41–48. Morgan Kaufmann, 1993.
5. R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.
6. C. Cortes and V.N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.
7. H. Daumé. Frustratingly easy domain adaptation. In *Annual meeting-association for computational linguistics*, volume 45, page 256, 2007.

8. T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(1):615–637, 2005.
9. T. Evgeniou and M. Pontil. Regularized multi-task learning. In *International Conference on Knowledge Discovery and Data Mining*, page 109, 2004.
10. L. Jacob and J. Vert. Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics (Oxford, England)*, 24(3):358–66, February 2008.
11. X. Lou and F. A. Hamprecht. Structured Learning for Cell Tracking. In *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, 2011.
12. X. Lou and F. A. Hamprecht. Structured Learning from Partial Annotations. In *The 29th International Conference on Machine Learning (ICML 2012)*, 2012.
13. X. Lou, U. Koethe, J. Wittbrodt, and F. A Hamprecht. Learning to segment dense cell nuclei with shape prior. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1012–1018. IEEE, 2012.
14. S.J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1345–1359, 2009.
15. G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch. An Empirical Analysis of Domain Adaptation Algorithms for Genomic Sequence Analysis. In D Koller, D Schuurmans, Y Bengio, and L Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1433–1440. NIPS, 2009.
16. I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453, 2006.
17. V. N. Vapnik and A. Ya. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.
18. C. Widmer, J. Leiva, Y. Altun, and G. Rätsch. Leveraging Sequence Classification by Taxonomy-based Multitask Learning. In B. Berger, editor, *Research in Computational Molecular Biology*, pages 522–534. Springer, 2010.
19. C. Widmer and G. Rätsch. Multitask Learning in Computational Biology. *JMLR W&CP. ICML 2011 Unsupervised and Transfer Learning Workshop.*, 27:207–216, 2012.
20. 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(Suppl 8):S5, January 2010.
21. Y. Zhang and D.Y. Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2010.