Multi-task Learning for Computational Biology: Overview and Outlook

Christian Widmer, Marius Kloft, and Gunnar Rätsch

1 Introduction

A series of seminal papers has greatly changed the way we view the field of machine learning. In their 1971 paper *On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities* [22], Vapnik and Chervonenkis laid out the foundations of statistical learning theory, which led to development of support vector machines (SVMs) in the 1990s [5, 8]. Subsequently, with the coming of the information age, machine learning – and computer science in general – has diverged into multiple fascinating and manifold branches, many of which can be traced back to these classical papers; for example, preference learning, multiple kernel learning, structured output learning, and transfer learning, to name just a few subfields. Meanwhile established machine learning methods such as SVMs have matured to a degree that, nowadays, they are frequently employed out-of-the-box in science and technology, for their favorable generalization performance while maintaining computational feasibility. In this article, we present an overview of one of the recent branches of machine learning, that is, *multi-task learning* (MTL) [6, 7], and discuss interesting applications in the domain of computational biology.

In science – and in biology in particular – supervised learning methods are often used to model complex mechanisms in order to describe and ultimately understand

---

Christian Widmer
Computational Biology Center, Memorial Sloan-Kettering Cancer Center, 415 E 68th street, New York, NY 10065, USA, and Machine Learning Group, Technische Universität Berlin, Franklinstr. 28/29, 10587 Berlin, Germany. e-mail: cwidmer@cbio.mskcc.org

Marius Kloft
Computational Biology Center, Memorial Sloan-Kettering Cancer Center, 415 E 68th street, New York, NY 10065, USA, and Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA. e-mail: kloftm@mskcc.org, mkloft@cs.nyu.edu

Gunnar Rätsch
Computational Biology Center, Memorial Sloan-Kettering Cancer Center, 415 E 68th street, New York, NY 10065, USA. e-mail: raetsch@cbio.mskcc.org

them. These models have to be rich enough to capture the considerable complexity of these mechanisms, which requires an enormous amount of training data. We frequently observe that the prediction accuracy does not yet saturate even when employing millions of training data points, which indicates that even using much more data could still help accuracy (cf., e.g., [20]). But how can we obtain this massive amount of training data?

Especially in the biomedical domain, obtaining additional labeled training examples can be either very costly or – e.g., due to technological limitations – even impossible. Multi-task learning overcomes this requirement by incorporating information from several related tasks in order to increase the accuracy of the target task at hand. For example, in genetics, we have a good understanding of how close two organisms are in terms of their evolutionary relationship; this information is summarized in the *tree of life*. Because basic genetic mechanisms tend to be relatively well conserved throughout evolution, we can benefit from combining data from several species for the detection of, for example, splice sites or promoter regions.

The relevance of MTL to Computational Biology goes beyond the setting where we view organism as tasks; we may also view different tasks corresponding to different related protein variants [12], cell lines, pathways [18], tissues, genes [16], technology platforms such as microarrays, experimental conditions, individuals, tumor subtypes, just to name a few. In this paper, we provide an overview of selected MTL approaches that have been successfully applied in computation biology. In this respect, our presentation is based on [24], but goes beyond the latter by covering also some very recent developments that are not yet systematically investigated in biology.

2 Multi-task 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 [17].

2.1 Relation to Transfer Learning

Transfer learning very generally refers to learning methods that transfer information from one or multiple source tasks to a target task with the aim of improving the prediction accuracy of the target task. Multi-task learning is a specific branch of transfer learning that is characterized by *simultaneously* learning the prediction models of all given tasks. Typically this is achieved by optimizing a joint learning criterion with respect to the various prediction functions. There exist several general strategies for multi-task learning [17]:

1. *instance-based transfer*, where data points from different domains are included into the learning problem, typically in combination with some form of re-weighting
2. *feature representation transfer*, where the instances from the various domains are mapped to a joint feature representation
3. *parameter transfer*, a form of parametric learning paradigm assuming that closely related tasks should also yield similar parameters in the learning model.

The main focus of this chapter is on parameter transfer, where the parameters of similar tasks are often coupled by a regularizer. This approach is often called *regularization-based multi-task learning*.

2.2 Regularization-based Multi-task Learning

From a historical perspective, regularization-based MTL is based on regularized risk minimization [22] and supervised learning methods such as the Support Vector Machine (SVM) [5, 8] or Logistic Regression. In regularized risk minimization, we aim at computing a model Θ minimizing an objective $J(\Theta)$ consisting of a loss-term that captures the error with respect to the training data (X, Y) and a regularizer that penalizes model complexity:

$$J(\Theta) = L(\Theta|X,Y) + R(\Theta).$$

This formulation can easily be generalized to the MTL setting, where we are interested in obtaining several models parametrized by $\Theta_1, ..., \Theta_T$, where T is the number of tasks. The above formulation can be extended by introducing an additional regularization term R_{MTL} that penalizes the discrepancy between the individual models:

$$J(\Theta_1, ..., \Theta_T) = \sum_{t=1}^{T} L(\Theta_t|X,Y) + \sum_{t=1}^{T} R(\Theta_t) + R_{\mathsf{MTL}}(\Theta_1, ..., \Theta_T).$$

A highly relevant and active line of research in this context is finding a good regularizer R_{MTL}. A proven approach to this end is to introduce a parameter matrix Q in the regularizer, giving rise to

$$J(\Theta_1, ..., \Theta_T, Q) = \sum_{t=1}^{T} L(\Theta_t|X,Y) + \sum_{t=1}^{T} R(\Theta_t) + R_{\mathsf{MTL}}(\Theta_1, ..., \Theta_T|Q).$$

Learning Q from the data is often referred to *learning the task similarities*.

2.2.1 Common Approaches

In the following, we denote the training examples by (x_i, y_i), $i = 1, \ldots, n$, each of which is associated with a task $\tau(i) \in \{1, \ldots, T\}$. We denote the set of indices of training points of the tth task by $I_t := \{i \in \{1, \ldots, n\} : \tau(i) = t\}$ and their number by $n_t := \#I_t$. One of the first works on regularization-based MTL is by Evgeniou and Pontil [11], where at optimization time all parameter vectors are "pulled" towards their average $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$,

$$\min_{b,w_1,\ldots,w_T} \frac{1}{2}\sum_{t=1}^{T} ||w_t||^2 + \sum_{t=1}^{T} ||w_t - \bar{w}||^2 + C\sum_{i=1}^{n} \ell\left(\left\langle x_i, w_{\tau(i)}\right\rangle + b, y_i\right).$$

Hereby ℓ denotes the hinge loss $\ell(z,y) = \max\{1 - yz, 0\}$. Note that all tasks are treated equally in the above formulation; however, often we are given the a 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 [10],

$$\min_{b,w_1,\ldots,w_T} \frac{1}{2}\sum_{t=1}^{T} ||w_t||^2 + \frac{1}{2}\sum_{s=1}^{T}\sum_{t=1}^{T} A_{st}\,||w_s - w_t||^2 + C\sum_{i=1}^{n} \ell\left(\left\langle x_i, w_{\tau(i)}\right\rangle + b, y_i\right), \quad (1)$$

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

$$\min_{b,w_1,\ldots,w_T} \frac{1}{2}\sum_{t=1}^{T} ||w_t||^2 + \sum_{s=1}^{T}\sum_{t=1}^{T} L_{st} w_s^T w_t + C\sum_{i=1}^{n} \ell\left(\left\langle x_i, w_{\tau(i)}\right\rangle + b, y_i\right),$$

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 *multi-task* kernel to be used in the corresponding dual:

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

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 [13], where a kernel on tasks K_T is used instead of the pseudo-inverse, giving rise to

$$K((x,s),(z,t)) = K_T(s,t) \cdot K_B(x,z). \quad (2)$$

Note that the corresponding joint feature space between task t and feature vector x can be written as a tensor product $\phi(t,x) = \phi_T(t) \cdot \phi_B(x)$ [13]. A "frustratingly easy" special case of (2) is studied in [9] 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 $K_T(s,t) = (1 + \delta_{s,t})$.

2.3 Learning Task Similarities

The above exposition assumes that we are a priori given a task similarity measure; but how can we access the relatedness of tasks? Although we are often provided with external information on task relatedness (e.g., an evolutionary history of organisms), the given task similarity measure is not necessarily informative of how tightly tasks should be coupled in the MTL algorithm in order to achieve better predictive performance; therefore we are in need of strategies to automatically learn or adjust the degree of coupling between tasks. In the following, we discuss several approaches to this problem – including our own method, Multi-task Multiple Kernel Learning (MT-MKL) [25].

2.3.1 A Simple Approach

Very recently, Blanchard et al. [4] presented a simple method to very generally compute a task similarity matrix A from the data at hand only. Their approach is based on the concept of Hilbert space embedding of probability distributions [21] and consists of two steps: first, computing an average similarity of the examples of a pair tasks,

$$\tilde{A}_{st} = \frac{1}{n_s n_t} \sum_{i \in I_s, j \in I_t} k(x_i, x_j),$$

and then applying a non-linear transformation such as

$$A_{st} = \left(1 + \tilde{A}_{st}\right)^d.$$

The authors show that under a hierachical frequentist i.i.d. setup this method enjoys favorable theoretical guarantees such as consistency when $\forall t = 1, \ldots, T : n_t \to \infty$ and $T \to \infty$.

We would like to remark at this point that the parameter d may be selected by cross validation. Generally, we may choose non-linear transformations of the form $A_{st} = \lambda_1 \cdot exp(\tilde{A}_{s,t}/\lambda_2)$ and select the parameters λ_1, λ_2 per cross validation.

2.3.2 Multi-task Relationship Learning (MTRL)

The authors in [26] propose a convex method of jointly learning a task similarity measure along with the individual parameter vectors. Their method extends the graph-regularized MTL Formulation given in (1).

$$\min_{W=(w_1,\ldots,w_t),\Omega} \quad \mathrm{tr}(W\Omega^{-1}W^T) + C\sum_{i=1}^{n} \ell\left(\left\langle x_i, w_{\tau(i)}\right\rangle + b, y_i\right), \tag{3}$$
$$\text{s.t.} \quad \mathrm{tr}(\Omega) = 1, \Omega \succcurlyeq 0$$

In [26] this formulation is solved by alternatingly optimizing the objective with respect to W and Ω, where in the each optimization step Ω is updated according to

$$\Omega = (W^\top W)^{1/2}/\mathrm{tr}(W^\top W).$$

The above approach is especially appealing when only little a priori information about the task similarities is present. An advantage over the method described in the previous section is that the task similarities and the weights w_t are learned simultaneously so that interactions can be captured well.

2.3.3 Multi-task Multiple Kernel Learning (MT-MKL)

In the following section, we present our own approach MT-MKL. The formulation given below is an extension of the ideas presented in [25]. An in-depth presentation, including a theoretical and empirical analysis as well as details on our large-scale implementation will be presented in a forthcoming journal publication.

Problem 1 (Primal MTL problem). Solve

$$\inf_{\theta:\|\theta\|_p\leq 1, W} \quad \frac{1}{2}\sum_{m=1}^{M} \frac{\|W_m\|_{Q_m}^2}{\theta_m} + C\sum_{i=1}^n l\left(y_i \sum_{m=1}^{M} \left\langle w_{m\tau(i)}, \varphi_m(x_i)\right\rangle\right).$$

where $W = (W_m)_{1\leq m\leq M}$, $W_m = (w_{m1},\ldots,w_{mT})$ and

$$\|W_m\|_{Q_m} := \mathrm{tr}(W_m Q_m W_m^\top) = \sqrt{\sum_{s,t=1}^{T} q_{mst}\left\langle w_{ms}, w_{mt}\right\rangle}$$

□

In the above problem, we assume being given a number M of external task similarity measures Q_m and kernel feature maps φ_m, each being associated with a weight θ_m. As in multiple kernel learning [14, 15], we automatically learn these weights, which allows us to fine-tune the given task similarities. In contrast to MTRL, our method is in need of some external information (which is often a reasonable assumption), but has in turn fewer free parameters to be learnt. Furthermore, the above method lets us associate different feature maps ϕ_m with different task similarity measures Q_m, which gives flexibility in encoding prior information.

2.3.4 Hierarchical MT-MKL

There are many ways in which the set of Q_m may be chosen. One valid strategy is to define a set of task groups, where information is shared within each group. In the setting of hierarchical task relations (e.g. the evolutionary history between different organisms), these groups come naturally from the inner nodes of our tree. Tasks corresponds to leaf nodes, or *taxa*, in this context, whereas each inner nodes defines a task group (see Figure 1). Let $G_m =: \{l | l \text{ is descendant of } m\}$ be the set of leaves below the sub-tree rooted at node m. Then, we can give the following definition for the hierarchically constructed task adjacency matrix

$$A_m(s,t) = \begin{cases} 1 & \text{if } s \in G_m \text{ and } t \in G_m \\ 0 & \text{else}\,. \end{cases}$$

As an example, consider the kernel defined by a hierarchical decomposition according to Figure 1. We seek a non-sparse weighting of the task sets defined by the hierarchy and will therefore use ℓ_2-norm MKL [14].

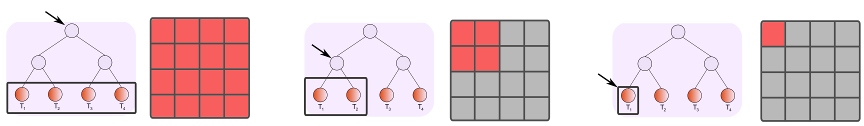

Fig. 1 Example of a hierarchical decomposition. According to a simple binary tree, it is shown that each node defines a subset of tasks (a block in the corresponding adjacency matrix on the left). Here, the decomposition is shown for only one path: the subset defined by the root node contains all tasks, the subset defined by the left inner node contains t_1 and t_2 and the subset defined by the leftmost leaf only contains t_1. Accordingly, each of the remaining nodes defines a subset S_i that contains all descendant tasks.

2.4 When does MTL pay off?

In this section, we give some practical guide lines about when it is promising to use MTL algorithms. First, the tasks should neither be too similar nor too different [23]. If the tasks are too different one will not be able to transfer information, or even end up with *negative transfer* [17]. On the other hand, if tasks are almost identical, it might suffice to pool the training examples and train a single combined classifier. Another integral factor that needs to be considered is whether the problem is *easy* or *hard* with respect to the available training data. In this context, the problem can be considered *easy* if the performance of a classifier saturates as a function of the available training data. In that case using more out-of-domain information will not improve classification performance.

In order to investigate the problem difficulty in the sense defined above, we can compute a learning curve (e.g., auROC as a function of the number of training examples). If the curve saturates when n is large, this indicates that multi-task learning will not considerably help performance, as model performance is most likely limited only by label noise. The same methodology can be employed to empirically measure the similarity between two tasks: we can compute saturation curves for various pairs of tasks, giving us a useful measure of whether or not transferring information between two tasks may be beneficial.

3 Application in Computational Biology

In this section, we give a brief example for an application in Computational Biology, where we have successfully employed Multitask Learning. The recognition of splice sites is an important problem in genome biology. By now it is fairly 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 transfered between source and target organisms in different evolutionary distances (i.e. one-to-many) and training set sizes [19]. We identified TL 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 [23]. 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.

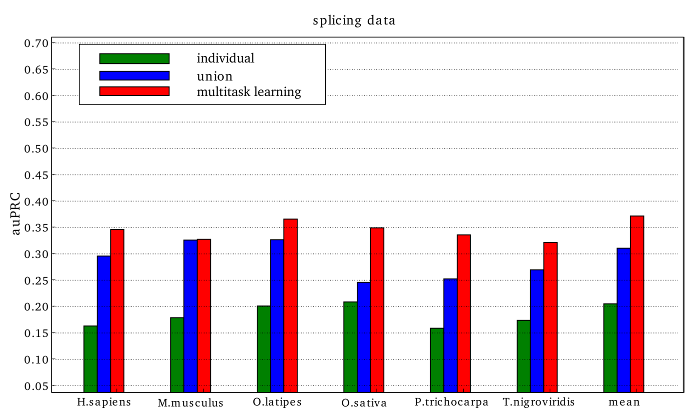

Fig. 2 Results of the RNA splicing experiment. Figure taken from [23].

The figure shows results for 6 out of 15 organisms for the baseline methods *individual* and *union* and the multitask learning algorithm described in Section 2.2. 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 [19, 23].

4 Conclusion

We have presented a brief overview of regularization-based multi-task learning methods and their application in the field of Computational Biology. Especially in the context of biomedical data, where generating training labels can be very expensive, multi-task learning can be viewed as an appealing means to obtain more cost-effective predictors. Accessing – or even learning – the similarity or relatedness of tasks is of central importance when applying multi-task learning methods, especially when are given prior knowledge of the hierarchical task structure, e.g., in form of a taxonomy. To this end, we have discussed several approaches such as multi-task multiple kernel learning to exploit task relationships in multi-task learning. We review some basic insights obtained in our experiments on MTL over the past years and give some practical guidelines for accessing, for a given dataset, whether or not multi-task learning is likely to help performance over more straight-forward baseline approaches. Lastly, we would like to mention that multi-task learning enjoys deep theoretical foundations. This has not been a focus of this article, though, but we refer the interested reader to, e.g., [1, 2, 3]. A common approach in MTL theory is to phrase multi-task learning within a hierarchical frequentist i.i.d. setup. This approach is taken, e.g., in Ando and Zhang [1] and Baxter [2], who extend the classical statistical learning theory of Vapnik & Chervonenkis [22] to multiple tasks.

Acknowledgements We thank 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).

References

1. R.K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.
2. J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

3. S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning. *Lecture notes in computer science*, pages 567–580, 2003.
4. 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.
5. 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.
6. R. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, pages 41–48. Morgan Kaufmann, 1993.
7. R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.
8. C. Cortes and V.N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.
9. H. Daumé. Frustratingly easy domain adaptation. In *Annual meeting-association for computational linguistics*, volume 45, page 256, 2007.
10. T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(1):615–637, 2005.
11. T. Evgeniou and M. Pontil. Regularized multi-task learning. In *International Conference on Knowledge Discovery and Data Mining*, page 109, 2004.
12. D. Heckerman, C. Kadie, and J. Listgarten. Leveraging information across HLA alleles/supertypes improves epitope prediction. *Journal of Computational Biology*, 14(6):736–746, 2007.
13. 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.
14. M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. lp-Norm Multiple Kernel Learning. *Journal of Machine Learning Research*, 12:953–997, 2011.
15. G. Lanckriet, N. Cristianini, L. E. Ghaoui, P. Bartlett, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. *JMLR*, 5:27–72, 2004.
16. F. Mordelet and J.P. Vert. Prodige: Prioritization of disease genes with multitask machine learning from positive and unlabeled examples. *BMC Bioinformatics*, 12:389, 2011.
17. S.J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1345–1359, 2009.
18. C.Y. Park, D.C. Hess, C. Huttenhower, and O.G. Troyanskaya. Simultaneous genome-wide inference of physical, genetic, regulatory, and functional pathway components. *PLoS computational biology*, 6(11):e1001009, 2010.
19. 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.
20. S. Sonnenburg, A. Zien, and G. Rätsch. Arts: Accurate recognition of transcription starts in human. *Bioinformatics*, 22(14):e472–e480, 2006.
21. B.K. Sriperumbudur, A. Gretton, K. Fukumizu, G.R.G. Lanckriet, and B. Schölkopf. Injective hilbert space embeddings of probability measures. In Rocco A. Servedio and Tong Zhang, editors, *Proceedings of the 21st Annual Conference on Learning Theory*, pages 111–122. Omnipress, 2008.
22. 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.
23. 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.
24. C. Widmer and G. Rätsch. Multitask Learning in Computational Biology. *JMLR W&CP. ICML 2011 Unsupervised and Transfer Learning Workshop.*, 27:207–216, 2012.
25. 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.
26. Y. Zhang and D.Y. Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2010.