Multitask Multiple Kernel Learning

Christian Widmer[1], Nora C. Toussaint[2], Yasemin Altun[3], Gunnar Rätsch[1]

1 Friedrich Miescher Laboratory of the Max Planck Society, Tübingen, Germany
2 Center for Bioinformatics Tübingen, Eberhard-Karls-Universität, Tübingen, Germany
3 Max Planck Institute for Biological Cybernetics, Tübingen, Germany

NIPS 2010 MKL workshop

December 11th, 2010

Multitask learning

- Common bottleneck: insufficient training data
- Combine information from several related tasks
- In order to know mutual relevance, task similarity is needed
- We seek task structure

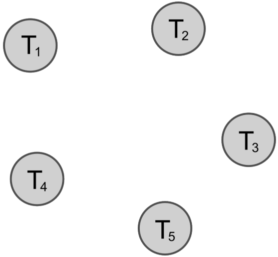


Figure: Task structure in MTL

Multitask learning

- Common bottleneck: insufficient training data
- Combine information from several related tasks
- In order to know mutual relevance, task similarity is needed
- We seek task structure

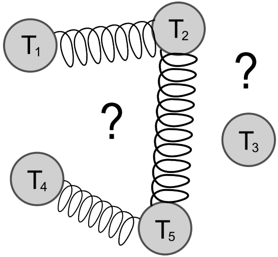


Figure: Task structure in MTL

MTL Framework

- We start from a well established MT-SVM formulation by Evgeniou and Pontil [2004]

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_T} \frac{1}{2} \sum_{t=1}^{T} ||\mathbf{w}_t||^2 + \sum_{t=1}^{T} ||\mathbf{w}_t - \mathbf{w}_0||^2 + C \sum_{t=1}^{T} \sum_{(\mathbf{x},y)\in\mathcal{D}_t} \ell\left(\langle \mathbf{x}, \mathbf{w}_t \rangle, y\right),$$

where ℓ is the hinge loss, $\ell(z,y) = \max\{1 - yz, 0\}$, $\mathbf{w}_0$ is the avrg.

- This corresponds to the dual (leaving out some constants):

$$\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) = \beta_1 K_B(\mathbf{x}_i, \mathbf{x}_j) + \beta_2 \delta_{t(i),t(j)} K_B(\mathbf{x}_i, \mathbf{x}_j),$$

where is $t(i)$ a task indicator function and $\beta_1, \beta_2 \geq 0$

MTL Framework

- We start from a well established MT-SVM formulation by Evgeniou and Pontil [2004]

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_T} \frac{1}{2}\sum_{t=1}^{T} ||\mathbf{w}_t||^2 + \sum_{t=1}^{T} ||\mathbf{w}_t - \mathbf{w}_0||^2 + C \sum_{t=1}^{T} \sum_{(\mathbf{x},y)\in\mathcal{D}_t} \ell\left(\langle \mathbf{x}, \mathbf{w}_t \rangle, y\right),$$

where ℓ is the hinge loss, $\ell(z, y) = \max\{1 - yz, 0\}$, $\mathbf{w}_0$ is the avrg.

- This corresponds to the dual (leaving out some constants):

$$\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) = \beta_1 K_B(\mathbf{x}_i, \mathbf{x}_j) + \beta_2 \delta_{t(i),t(j)} K_B(\mathbf{x}_i, \mathbf{x}_j),$$

where is $t(i)$ a task indicator function and $\beta_1, \beta_2 \geq 0$

MTL Framework

- We start from a well established MT-SVM formulation by Evgeniou and Pontil [2004]

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_T} \frac{1}{2}\sum_{t=1}^{T} ||\mathbf{w}_t||^2 + \sum_{t=1}^{T} ||\mathbf{w}_t - \mathbf{w}_0||^2 + C\sum_{t=1}^{T}\sum_{(\mathbf{x},y)\in\mathcal{D}_t} \ell\left(\langle \mathbf{x}, \mathbf{w}_t\rangle, y\right),$$

where ℓ is the hinge loss, $\ell(z, y) = \max\{1 - yz, 0\}$, $\mathbf{w}_0$ is the avrg.

- This corresponds to the dual (leaving out some constants):

$$\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) = \beta_1 K_B(\mathbf{x}_i, \mathbf{x}_j) + \beta_2 \delta_{t(i),t(j)} K_B(\mathbf{x}_i, \mathbf{x}_j),$$

where is $t(i)$ a task indicator function and $\beta_1, \beta_2 \geq 0$

Meta-tasks

Figure: Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Meta-tasks

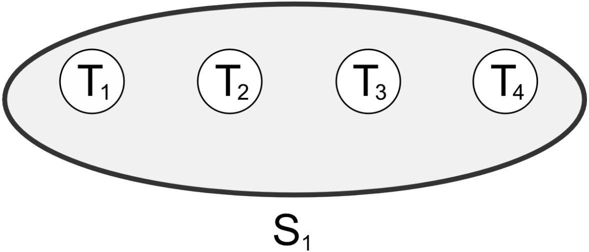


Figure: Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Meta-tasks

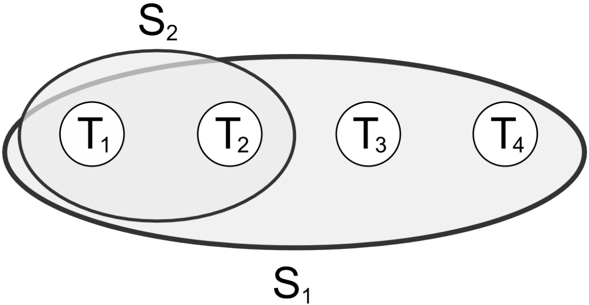


Figure: Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Meta-tasks

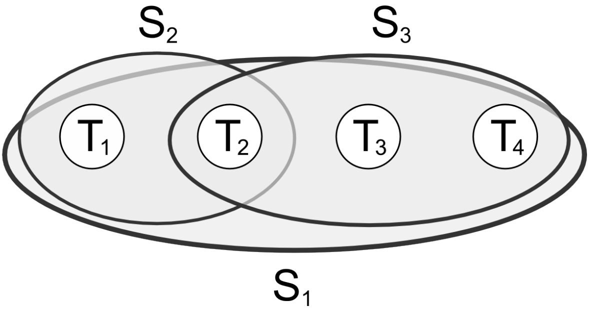


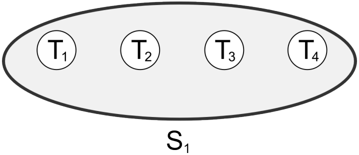
Figure: Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

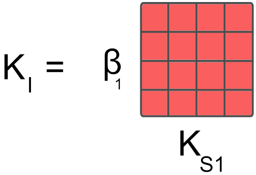
Decomposition of kernel matrix

$$K_S(x,y) = \begin{cases} K_B(x,y), & \text{if } \mathrm{task}(x) \in S \wedge \mathrm{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



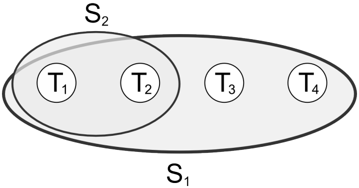
Example for collection of meta-tasks:



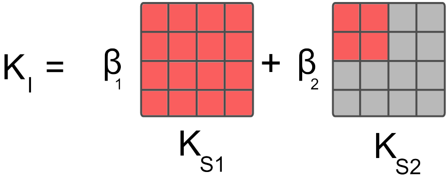
Decomposition of kernel matrix

$$K_S(x, y) = \begin{cases} K_B(x, y), & \text{if task}(x) \in S \wedge \text{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



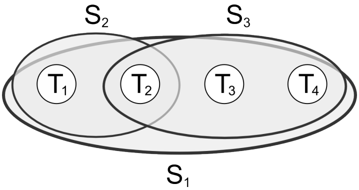
Example for collection of meta-tasks:



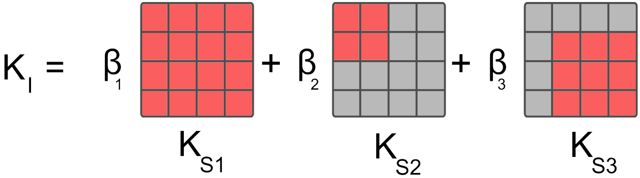
Decomposition of kernel matrix

$$K_S(x,y) = \begin{cases} K_B(x,y), & \text{if } \mathrm{task}(x) \in S \wedge \mathrm{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



Example for collection of meta-tasks:



Optimization strategy: q-norm MKL

We use the MKL formulation by Kloft et al. [2009]:

$$
\begin{aligned}
\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \quad & \mathbf{1}^T \boldsymbol{\alpha} - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \sum_{t=1}^{|\mathcal{I}|} \beta_t K_{S_t}(x_i, x_j) \\
\text{s.t.} \quad & ||\boldsymbol{\beta}||_q^q \leq 1, \boldsymbol{\beta} \geq 0 \\
& \mathbf{Y}^T \boldsymbol{\alpha} = 0, 0 \leq \boldsymbol{\alpha} \leq C
\end{aligned}
$$

- We learn the weights $\sum_{t=1}^{|\mathcal{I}|} \beta_t K_{S_t}$
- q lets us choose the appropriate norm (sparse/non-sparse)

$\Rightarrow$ How to define collection $\mathcal{I}$ of meta-tasks?

Optimization strategy: q-norm MKL

We use the MKL formulation by Kloft et al. [2009]:

$$
\begin{aligned}
\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \quad & \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}(x_i, x_j) \\
\text{s.t.} \quad & ||\boldsymbol{\beta}||_q^q \leq 1, \boldsymbol{\beta} \geq 0 \\
& \mathbf{Y}^T\boldsymbol{\alpha} = 0, 0 \leq \boldsymbol{\alpha} \leq C
\end{aligned}
$$

- We learn the weights $\sum_{t=1}^{|\mathcal{I}|} \beta_t K_{S_t}$
- q lets us choose the appropriate norm (sparse/non-sparse)

$\Rightarrow$ How to define collection $\mathcal{I}$ of meta-tasks?

Power-set based approach

If no prior information available:

$$\mathcal{I}_{\mathcal{P}} = \{S | S \in \mathcal{P}(\mathcal{T}) \land S \neq \emptyset\}$$

- Consider Powerset $\mathcal{P}(\mathcal{T})$
- Most meta-tasks in power-set will be meaningless
  $\rightarrow$ learn sparse weights: $q = 1$
- Approach can be used to identify task structure *ab-initio*
- 2^M meta-tasks $\rightarrow$ computationally expensive

Hierarchical decomposition

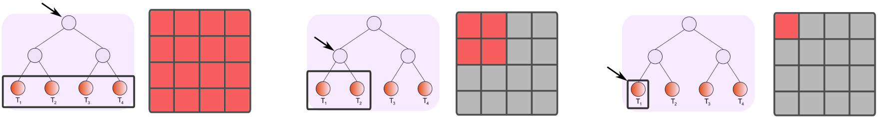
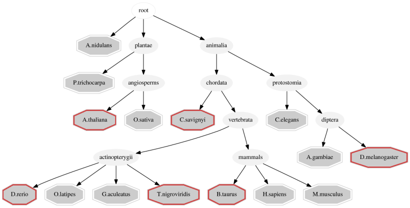


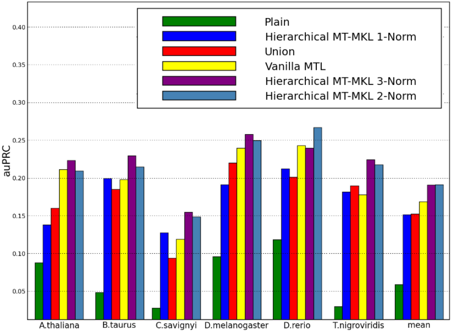
Figure: Example of taxonomy-based decomposition.

$$\mathcal{I}_{\mathcal{G}} = \{\mathit{leaves}(\mathit{node}) | \mathit{node} \in \mathcal{G}\}$$

- Meta-tasks are defined by taxonomy $\mathcal{G}$
- Taxonomy $\mathcal{G}$ gives us reasonable groups
  - Idea is to refine structure
  - Non-sparse combination ($q > 1$) for groupings

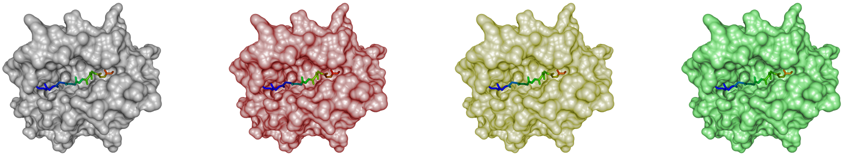
Experiments (a): Splice-site recognition





- Taxonomy is used to define collection of meta-tasks $\mathcal{I}$
- Baselines: Plain, Union, Vanilla
- Best performance for norm $q = 2$

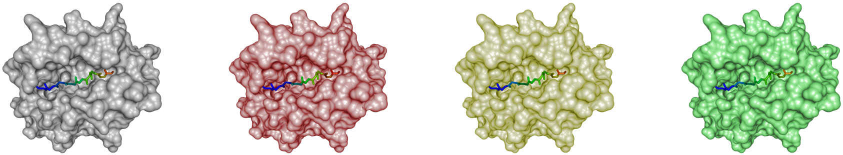
Experiments (b): MHC-I binding prediction



Method	Plain	Union	Vanilla MTL	Powerset MT-MKL
auPRC	67.1%	57.6%	67.9%	69.9%

- No task structure (used): Powerset MT-MKL
- Question: Can we identify meaningful structure?

Experiments (b): MHC-I binding prediction

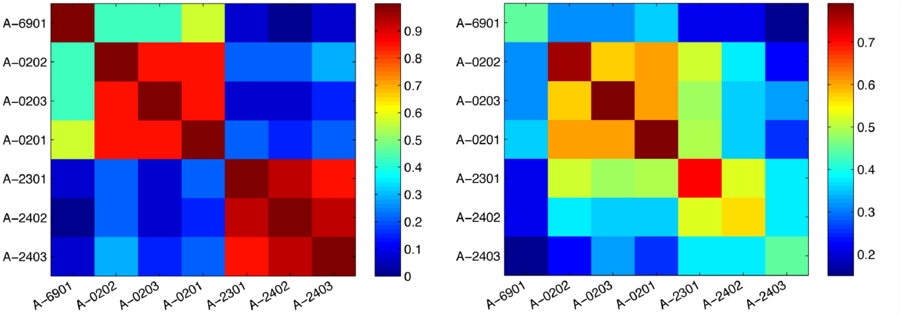


Method	Plain	Union	Vanilla MTL	Powerset MT-MKL
auPRC	67.1%	57.6%	67.9%	69.9%

- No task structure (used): Powerset MT-MKL
- Question: Can we identify meaningful structure?

Experiments (b): MHC-I binding prediction

Learned weights can also be used for interpretation purposes:



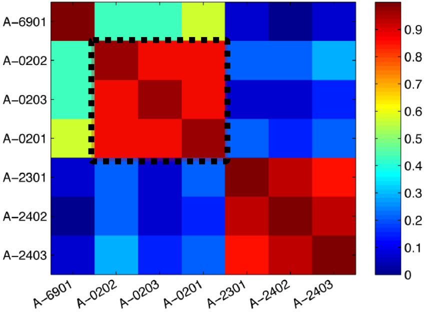
(a) Sequence similarity between alleles

(b) Task similarity computed from weights

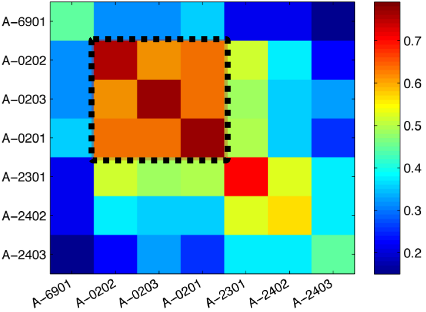
- Similarity computed from meta-task weights
- Comparison to similarity between peptide sequences
- Successfully identifies biological meaningful structure

Experiments (b): MHC-I binding prediction

Learned weights can also be used for interpretation purposes:



(a) Sequence similarity between alleles

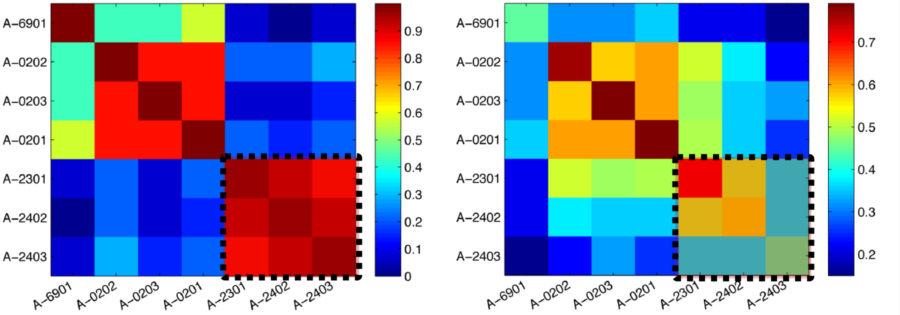


(b) Task similarity computed from weights

- Similarity computed from meta-task weights
- Comparison to similarity between peptide sequences
- Successfully identifies biological meaningful structure

Experiments (b): MHC-I binding prediction

Learned weights can also be used for interpretation purposes:



(a) Sequence similarity between alleles (b) Task similarity computed from weights

- Similarity computed from meta-task weights
- Comparison to similarity between peptide sequences
- Successfully identifies biological meaningful structure

Conclusion

- Summary
  - Idea: Learning/Refining meta-tasks with MKL
  - Hierarchical Decomposition
  - Power-set approach
  - Two applications from Computational Biology
- Future work
  - Efficient optimization strategy for Powerset
  - Use other variants of MKL?

Acknowledgments

- Nora C. Toussaint [4]
- Yasemin Altun [2]
- Jose Leiva [1,2]
- Sören Sonnenburg [3]
- Klaus Robert Müller [3]
- Gunnar Rätsch [1]

1 Friedrich Miescher Laboratory of the Max Planck Society

2 Max Planck Institute for Biological Cybernetics

3 Berlin Institute of Technology

4 Center for Bioinformatics Tübingen

Thank you for your attention.

Acknowledgments

- Nora C. Toussaint [4]
- Yasemin Altun [2]
- Jose Leiva [1,2]
- Sören Sonnenburg [3]
- Klaus Robert Müller [3]
- Gunnar Rätsch [1]

1 Friedrich Miescher Laboratory of the Max Planck Society

2 Max Planck Institute for Biological Cybernetics

3 Berlin Institute of Technology

4 Center for Bioinformatics Tübingen

Thank you for your attention.

References I

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In Won Kim, Ron Kohavi, Johannes Gehrke, and William DuMouchel, editors, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 109–117. ACM, 2004. ISBN 1-58113-888-1.

Marius Kloft, Ulf Brefeld, Sören Sonnenburg, Pavel Laskov, Klaus-Robert Müller, and Alexander Zien. Efficient and accurate lp-norm multiple kernel learning. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 997–1005. MIT Press, 2009.

Proof of positive semi-definiteness I

$$\hat{K}(x,z) = M(x,z) \cdot K_{\text{base}}(x,z)$$

$$\mathbf{a}^T M\mathbf{a} \geq 0 \quad \forall \mathbf{a} \in R^n$$

$$a^T M\mathbf{a} = \mathbf{a}^T \cdot \left[\begin{array}{ccc|cc} a_1 & \dots & a_1 & & \\ & \ddots & & & 0 \\ a_k & \dots & a_k & & \\ \hline & & & \ddots & \\ & 0 & & & 0 \end{array} \right] = \sum_{i=1}^{k} (\mathbf{a}_i \cdot (\sum_{j=1}^{k} \mathbf{a}_j))$$

$$= (\sum_{i=1}^{k} \mathbf{a}_i) \cdot (\sum_{j=1}^{k} \mathbf{a}_j) = (\sum_{j=1}^{k} \mathbf{a}_j)^2 \geq 0. \quad \square$$

Prediction function I

$$f_t(y) = \sum_{i=0}^{N} \alpha_i y_i \sum_{S_j \in \mathcal{I}; t \in S_j} \beta_j K_{S_j}(x_i, y),$$

where N is the total number of training examples of all tasks combined.

Computing task similarity from weights I

We define the collection of task sets containing task t_k as $\mathcal{T}_{t_k} = \{S | t_k \in S \wedge S \in \mathcal{T}\}$. Using this definition, we can define the similarity $\gamma_{k,l}$ between two tasks by summing up the weights of the shared task sets S_i

$$\gamma_{k,l} = \sum_{S_i \in \mathcal{T}_{t_k} \cap \mathcal{T}_{t_l}} \beta_i. \tag{1}$$