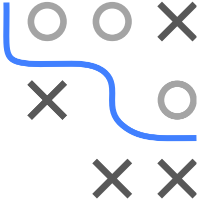
KRONECKER KERNEL RIDGE REGRESSION



- In MTP with target features, we often use kernel methods.
- Consider the following pairwise model representation in the primal:

$$f(\mathbf{x}, \mathbf{t}) = \boldsymbol{\omega}^\top \left(\phi(\mathbf{x}) \otimes \psi(\mathbf{t})\right),$$

  where ϕ is feature mapping for features and ψ is feature mapping for target (features) and $\otimes$ is Kronecker product.
- This yields Kronecker product pairwise kernel in the dual:

$$f(\mathbf{x}, \mathbf{t}) = \sum_{(\mathbf{x}', \mathbf{t}') \in \mathcal{D}} \alpha_{(\mathbf{x}', \mathbf{t}')} \cdot k(\mathbf{x}, \mathbf{x}') \cdot g(\mathbf{t}, \mathbf{t}') = \sum_{(\mathbf{x}', \mathbf{t}') \in \mathcal{D}} \alpha_{(\mathbf{x}', \mathbf{t}')} \Gamma((\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}')),$$

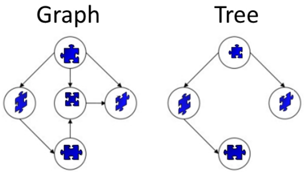
  where k is kernel for feature map ϕ, g kernel for feature map ψ and $\alpha_{(\mathbf{x}', \mathbf{t}')}$ are dual parameters determined by:

$$\min_{\boldsymbol{\alpha}} ||\boldsymbol{\Gamma}\boldsymbol{\alpha} - \boldsymbol{z}||_2^2 + \lambda \boldsymbol{\alpha}^\top \boldsymbol{\Gamma}\boldsymbol{\alpha}, \text{ where } \boldsymbol{z} = \mathrm{vec}(Y)$$

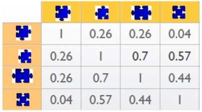
- Commonly used in zero-shot learning.

Stock et al., A comparative study of pairwise learning methods based on kernel ridge regression, Neural Computation 2018.

EXPLOITING RELATIONS IN REGULARIZATION



Similarity



	I	0.26	0.26	0.04
	0.26	I	0.7	0.57
	0.26	0.7	I	0.44
	0.04	0.57	0.44	I

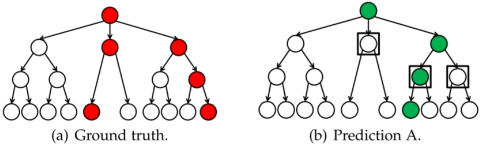
- Graph-based regularization for graph-type relations in targets:

$$\min_{\Theta} \| Y - \Phi\Theta \|_F^2 + \lambda \sum_{m=1}^{l} \sum_{m' \in \mathcal{N}(m)} \| \boldsymbol{\theta}_m - \boldsymbol{\theta}_{m'} \|^2,$$

  where $\mathcal{N}(j)$ is the set of targets related to target j.
- The graph or tree is given as prior information.
- Can be extended to a weighted version aware of the similarities

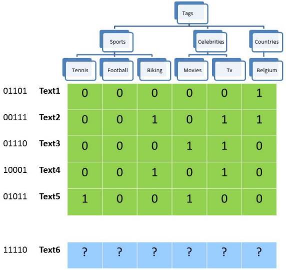
Gopal and Yang, Recursive regularization for large-scale classification with hierarchical and graphical dependencies, KDD 2013.

HIERARCHICAL MULTI-LABEL CLASSIFICATION



(a) Ground truth.

(b) Prediction A.



		Tennis	Football	Biking	Movies	Tv	Belgium
01101	**Text1**	0	0	0	0	0	1
00111	**Text2**	0	0	1	0	1	1
01110	**Text3**	0	0	0	1	1	0
10001	**Text4**	0	0	1	0	1	0
01011	**Text5**	1	0	0	1	0	0
11110	**Text6**	?	?	?	?	?	?

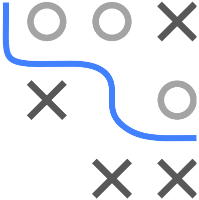
- Hierarchies can also be used to define specific loss functions, such as the Hierarchy-loss:

$$L_{Hier}(\mathbf{y}, f) = \sum_{m: y_m \neq \hat{y}_m} c_m \, \mathbb{1}_{[anc(y_m)=anc(\hat{y}_m)]},$$

- This is rather common in multi-label classification problems.

Bi and Kwok, Bayes-optimal hierarchical multi-label classification, IEEE Transactions on Knowledge and Data Engineering, 2014.

PROBABILISTIC CLASSIFIER CHAINS



- Estimate the joint conditional distribution $\mathbb{P}(\mathbf{y} \mid \mathbf{x})$.
- For optimizing the subset 0/1 loss:

$$L_{0/1}(\mathbf{y}, \hat{\mathbf{y}}) = \mathbb{1}_{[\mathbf{y} \neq \hat{\mathbf{y}}]}$$

- Repeatedly apply the *product rule* of probability:

$$\mathbb{P}(\mathbf{y} \mid \mathbf{x}) = \prod_{j=m}^{l} \mathbb{P}\left(y_m \mid \mathbf{x}, y_1, \ldots, y_{m-1}\right).$$

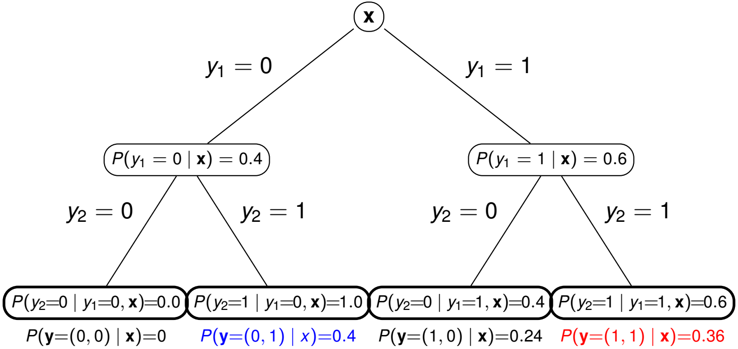
- Learning relies on constructing probabilistic classifiers for

$$\mathbb{P}\left(y_m | \mathbf{x}, y_1, \ldots, y_{m-1}\right),$$

independently for each $m = 1, \ldots, l$.

PROBABILISTIC CLASSIFIER CHAINS

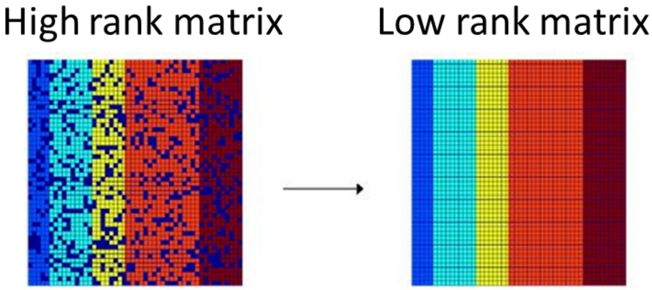
- Inference relies on exploiting a probability tree:


$\mathbf{x}$
$y_1 = 0$
$y_1 = 1$
$P(y_1 = 0 \mid \mathbf{x}) = 0.4$
$P(y_1 = 1 \mid \mathbf{x}) = 0.6$
$y_2 = 0$
$y_2 = 1$
$y_2 = 0$
$y_2 = 1$
$P(y_2{=}0 \mid y_1{=}0, \mathbf{x}){=}0.0$
$P(y_2{=}1 \mid y_1{=}0, \mathbf{x}){=}1.0$
$P(y_2{=}0 \mid y_1{=}1, \mathbf{x}){=}0.4$
$P(y_2{=}1 \mid y_1{=}1, \mathbf{x}){=}0.6$
$P(\mathbf{y}{=}(0,0) \mid \mathbf{x}){=}0$
$P(\mathbf{y}{=}(0,1) \mid x){=}0.4$
$P(\mathbf{y}{=}(1,0) \mid \mathbf{x}){=}0.24$
$P(\mathbf{y}{=}(1,1) \mid \mathbf{x}){=}0.36$

- For subset 0/1 loss one needs to find $h(\mathbf{x}) = \arg\max_{\mathbf{y}} \mathbb{P}(\mathbf{y} \mid \mathbf{x})$.
- Greedy and approximate search techniques with guarantees exist.
- Other losses: compute the prediction on a sample from $\mathbb{P}(\mathbf{y} \mid \mathbf{x})$.

Dembczynski et al., An analysis of chaining in multi-label classification, ECAI 2012.

LOW-RANK APPROXIMATION



- Low rank = some structure is shared across targets
- Typically perform low-rank approx of param matrix:

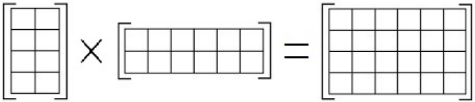
$$\min_{\Theta} \|Y - \Phi\Theta\|_F^2 + \lambda\,\mathrm{rank}(\Theta)$$

Chen et al., A convex formulation for learning shared structures from multiple tasks, ICML 2009.

LOW-RANK APPROXIMATION

- Θ: parameter matrix of dimensionality $p \times l$
- p: the number of (projected) features
- l: the number of targets
- Assume a low-rank structure of A:

$$U \quad \times \quad V \quad = \quad A$$



- We can write $\Theta = UV$ and $\Theta \mathbf{x} = UV\mathbf{x}$
- V is a $p \times \hat{l}$ matrix
- U is an $\hat{l} \times l$ matrix
- $\hat{l}$ is the rank of Θ