KRONECKER KERNEL RIDGE REGRESSION

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

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

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:

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

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:

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

- 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

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

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

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

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

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

where k is kernel for feature map ϕ, g kernel for feature map ψ and $(\mathbf{x}, \mathbf{t})$ are the pairwise features defined by $\otimes$.

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

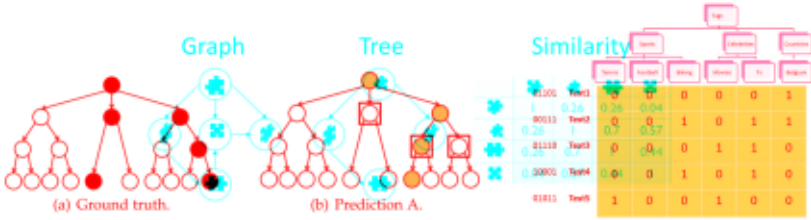
$$\min_{\alpha} \|\Gamma\alpha - \mathbf{z}\|^2 + \lambda \alpha^\top \Gamma \alpha, \text{ where } \mathbf{z} = \text{vec}(Y)$$

- Commonly used in zero-shot learning.

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

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

HIERARCHICAL MULTI-LABEL CLASSIFICATION



- Graph-based regularization for graph-type relations in targets:
- Hierarchies can also be used to define specific loss functions, such as the Hierarchy-loss:

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

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

where $\mathcal{N}(j)$ is the set of targets related to target j.

- The graph or tree is given as prior information.
- This is rather common in multi-label classification problems.
- Can be extended to a weighted version aware of the similarities

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

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

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:

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

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

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

- Learning relies on constructing probabilistic classifiers for

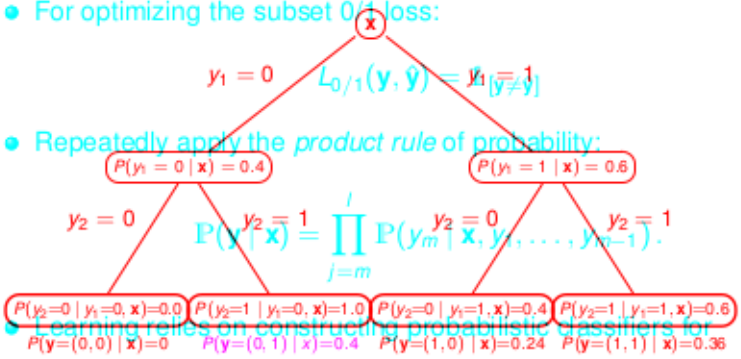
$$\mathbb{P}(y_m \mid \mathbf{x}, y_1, \ldots, y_{m-1}),$$

- This is rather common in multi-label classification problems. independently for each $m = 1, \ldots, l$.

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

PROBABILISTIC CLASSIFIER CHAINS

- Inference relies on exploiting a probability tree: $\mathbb{P}(\mathbf{y} \mid \mathbf{x})$.
- For optimizing the subset 0/1 loss:



- Repeatedly apply the *product rule* of probability:

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

- Learning relies on constructing probabilistic classifiers for

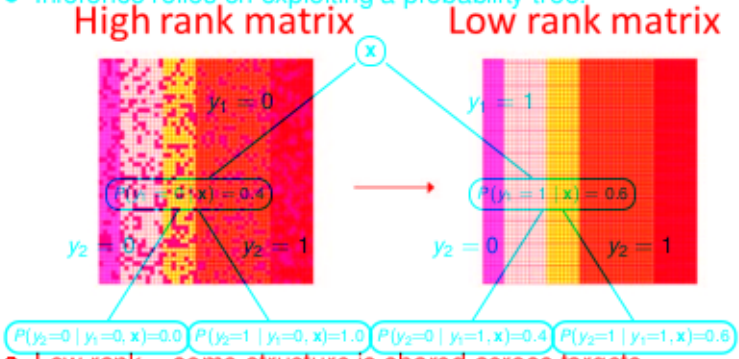
$$\mathbb{P}(y_m | \mathbf{x}, y_1, \ldots, y_{m-1}),$$

- 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 CHAINS

- Inference relies on exploiting a probability tree:



- Low rank = some structure is shared across targets
- Typically perform low-rank approx of param matrix:
- 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 $\mathbb{P}(\mathbf{y} \mid \mathbf{x})$.

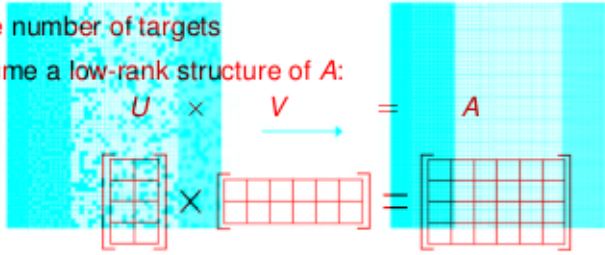
$$\min_{\Theta} \|Y - \Phi\Theta\|_F^2 + \lambda \operatorname{rank}(\Theta)$$

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

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

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:



- We can write $\Theta = UV$ and $\Theta\mathbf{x} = UV\mathbf{x}$
- Low rank = some structure is shared across targets
- V is a $p \times \hat{l}$ matrix
- Typically perform low-rank approx of param matrix:
- U is an $\hat{l} \times l$ matrix
- $\hat{l}$ is the rank of Θ $\min_{\Theta} \|Y - \Phi\Theta\|_F^2 + \lambda \operatorname{rank}(\Theta)$

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:

$$U \times V = A$$

- 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