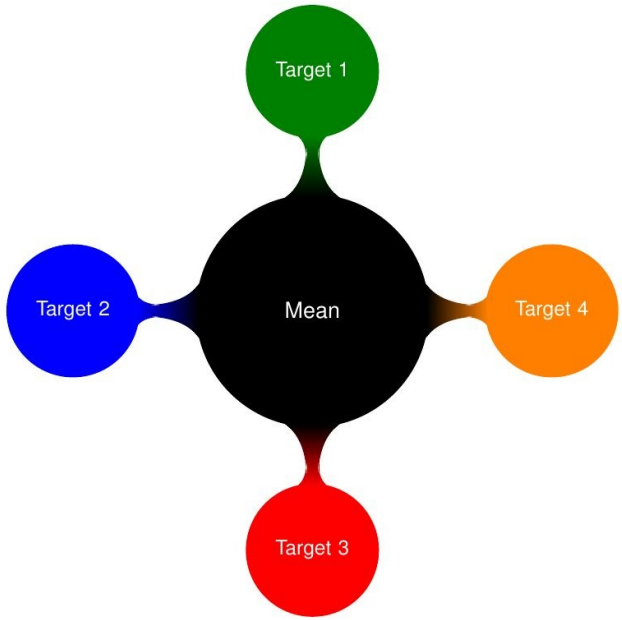
Advanced Machine Learning

Methods for Multi-Target Prediction

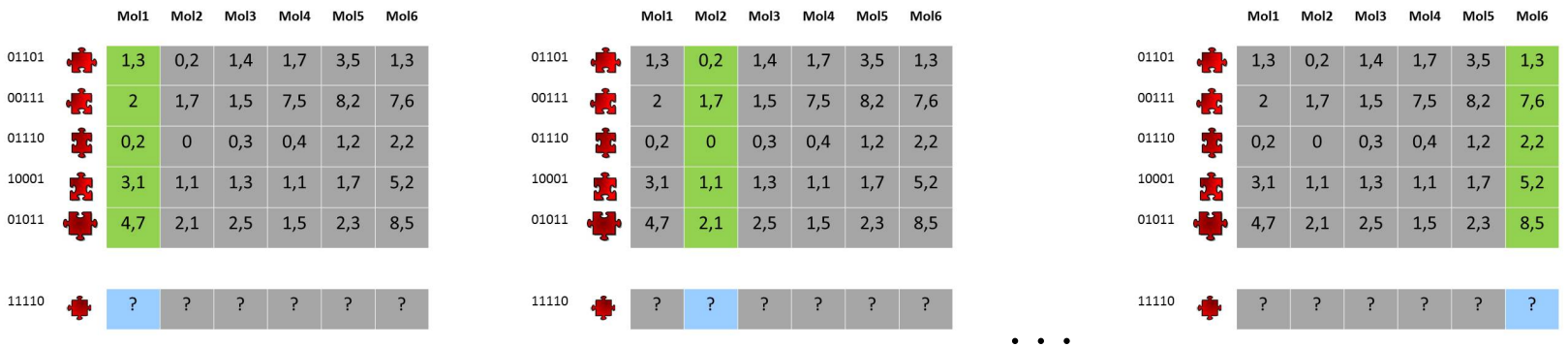


Learning goals

- Get an overview of the existing groups of methods for MTP
- Know that treating targets independently is often sub-optimal

INDEPENDENT MODELS

- The most naive way to make multi-target predictions is by learning a model for each target independently , i.e., for each target one uses one model to make the predictions for (only) that target.



- In multi-label classification this approach is also known as *binary relevance learning.*
- The advantage of this approach is that it is quite easy to realize, as for single-target prediction we have a wealth of methods available.

INDEPENDENT MODELS

- We illustrate the typical approach by means of linear basis function model for the j-th target:

$$f_j(\mathbf{x}) = \mathbf{a}_j^\mathsf{T} \phi(\mathbf{x}) \,,$$

  where $\mathbf{a}_j$ is a target-specific parameter vector and ϕ some feature mapping.

- The parameter vectors are found by solving a (regularized) optimization problem:

$$\min_A \| Y - \mathbf{X}A \|_F^2 + \sum_{j=1}^m \lambda_j \left\| \mathbf{a}_j \right\|^2 ,$$

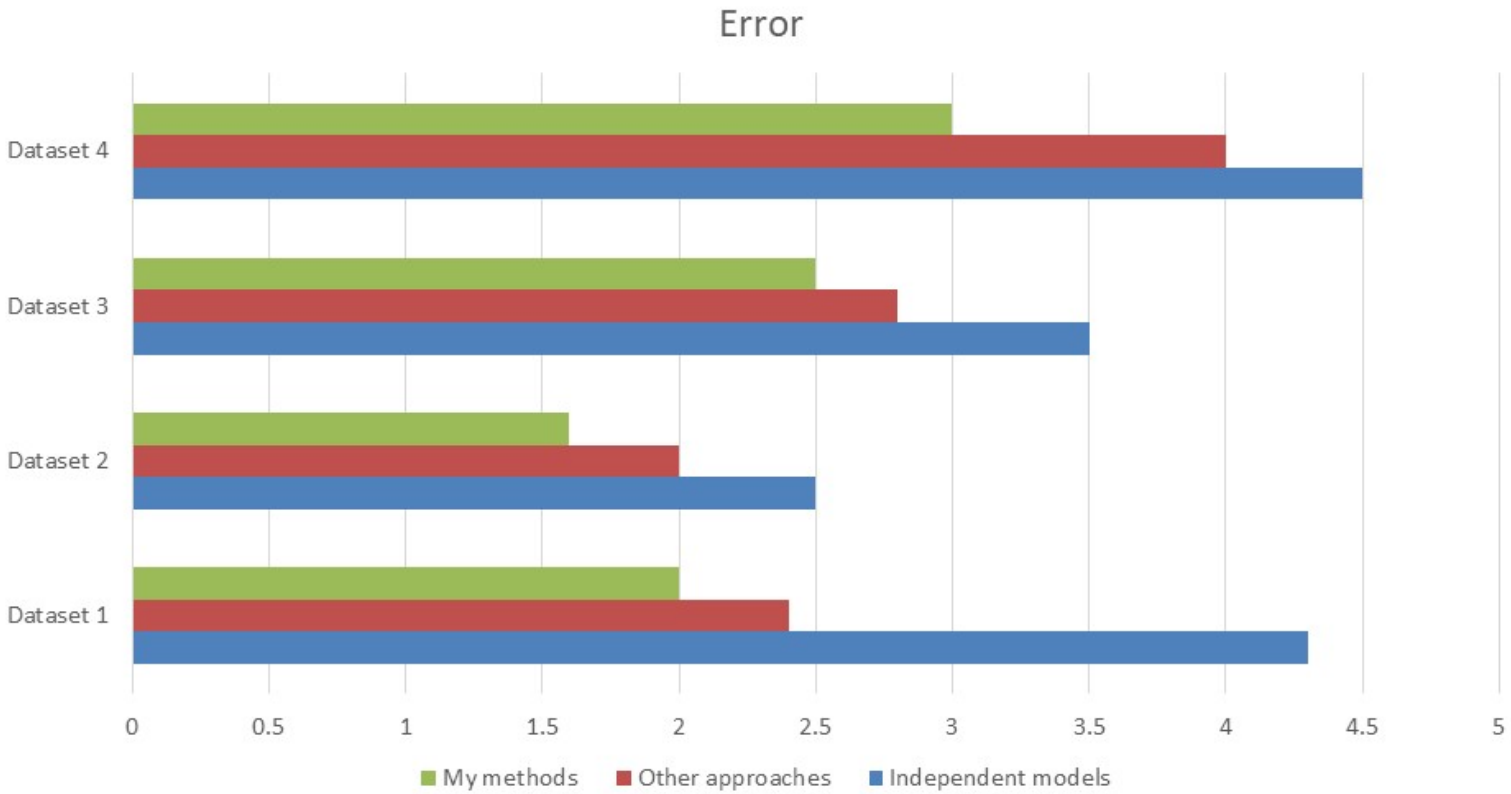
  where $\|B\|_F^2 = \sqrt{\sum_{i=1}^n \sum_{j=1}^m B_{i,j}^2}$ is the Frobenius norm for a matrix $B \in \mathbb{R}^{n \times m}$ and

$$\mathbf{X} = \begin{bmatrix} \phi(\mathbf{x}^{(1)})^\top \\ \vdots \\ \phi(\mathbf{x}^{(n)})^\top \end{bmatrix} \qquad A = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_m \end{bmatrix} .$$

- The norm for regularizing the target-specific parameters can vary:
  - L2-norm $\rightsquigarrow$ Multivariate Ridge Regression.
  - L1-norm $\rightsquigarrow$ Multivariate Lasso Regression.

INDEPENDENT MODELS: PRACTICAL PERFORMANCE

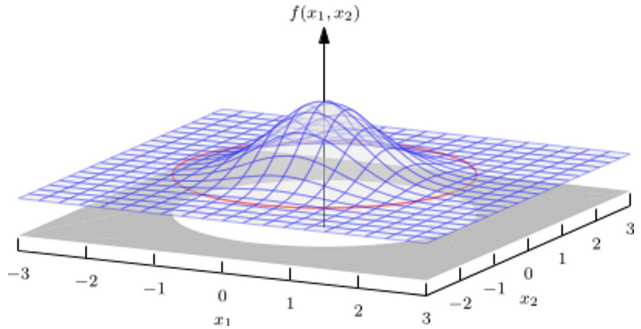
The experimental results section of a typical MTP paper:



$\rightsquigarrow$ Independent models do not exploit target dependencies compared to more sophisticated methods, which seems to be a key for better performance in MTP problems.

JAMES-STEIN ESTIMATION

- Consider a sample of a multivariate normal distribution $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma^2 \mathbf{I})$, i.e., the components $y_1, \ldots, y_m$ of $\mathbf{y}$ are independent of each other.



- What is the best estimator of the mean vector $\boldsymbol{\theta}$ w.r.t. mean-squared error (MSE): $\mathbb{E}[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^2]$?
- The single-observation maximum likelihood estimator (which is also the least-squares estimate in this case) is $\hat{\boldsymbol{\theta}}^{\mathrm{ML}} = \mathbf{y}$.
- The James-Stein estimator [James & Stein (1961)]:

$$\hat{\boldsymbol{\theta}}^{\mathrm{JS}} = \left(1 - \frac{(m-2)\sigma^2}{\|\mathbf{y}\|_2^2}\right) \mathbf{y}$$

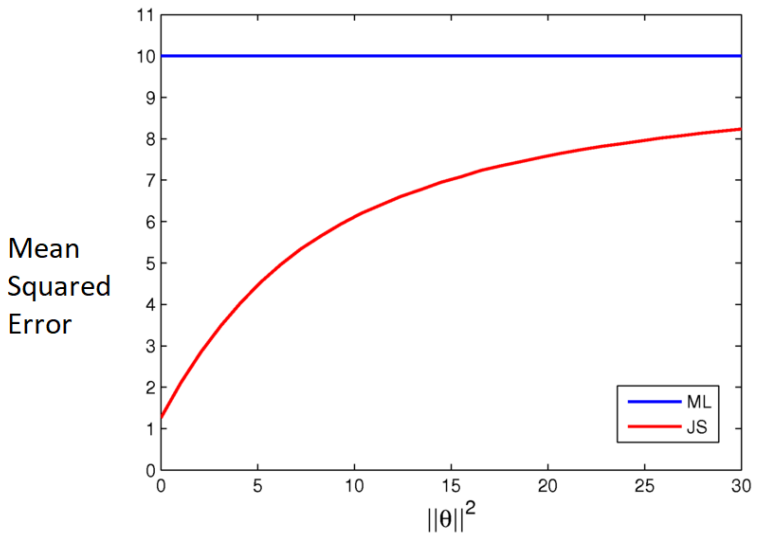
  has a smaller MSE than the maximum likelihood estimator!

$\rightsquigarrow$ Improvements over independent predictions can be achieved even for problems without any statistical dependence between the targets!

- Explanation: The variance is reduced by introducing a bias.

JAMES-STEIN ESTIMATION

- Works best when the norm of the mean vector is close to zero:



- Regularization towards other directions, say $\boldsymbol{v}$, is also possible:

$$\hat{\boldsymbol{\theta}}^{\mathrm{JS}}(\boldsymbol{v}) = \left(1 - \frac{(m-2)\sigma^2}{\|\mathbf{y} - \boldsymbol{v}\|^2}\right)(\mathbf{y} - \boldsymbol{v}) + \boldsymbol{v}$$

- Only outperforms the maximum likelihood estimator w.r.t. the sum of squared errors over all components, and only when $m \geq 3$.
- The result can be generalized to the case with n iid observations of $\mathcal{N}(\boldsymbol{\theta}, \sigma^2\mathbf{I})$ as well as for the case of iid samples of $\mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$ for a general covariance matrix $\boldsymbol{\Sigma}$.

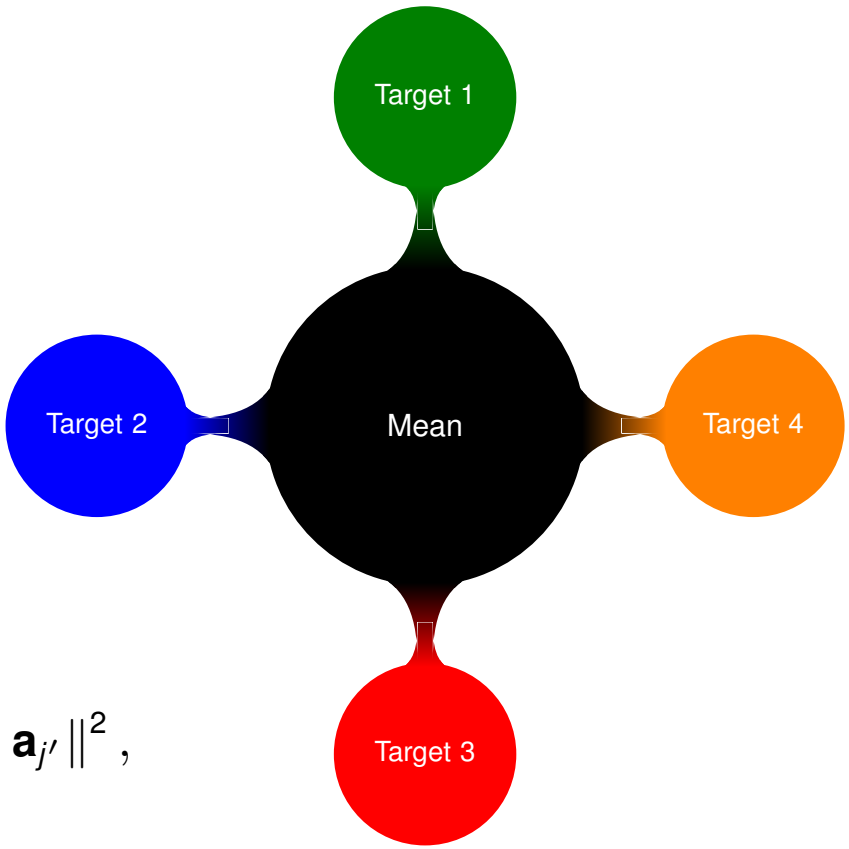
Similarity-enforcing methods

MEAN-REGULARIZED MULTI-TASK LEARNING

- *Idea*: The models for the different targets should behave similar. How?
- *Simple solution*: The parameters of these models should have similar values.
- *Approach*: Bias the parameter vectors towards their overall mean vector:

$$\min_A \| Y - \mathbf{X}A \|_F^2 + \lambda \sum_{j=1}^{m} \| \mathbf{a}_j - \frac{1}{m} \sum_{j'=1}^{m} \mathbf{a}_{j'} \|^2 ,$$

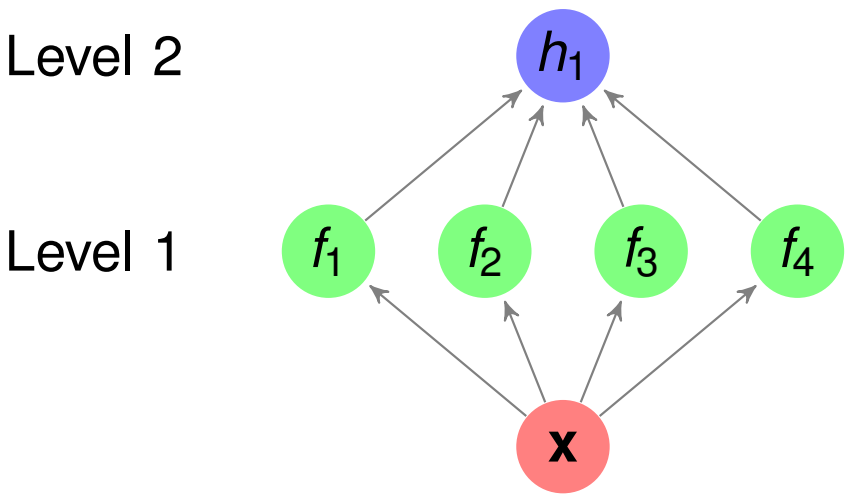
- *Disadvantage*: The assumption of all target models being similar might be invalid for many applications.



Evgeniou and Pontil, Regularized multi–task learning, KDD 2004

STACKING (STACKED GENERALIZATION)

- Originally introduced as a general ensemble learning or blending technique.
- Level 1 learners: apply a series of ML methods on the same dataset (or, one ML method on bootstrap samples of the dataset)
- Level 2 learner: apply an ML method to a new dataset consisting of the predictions obtaining at Level 1



Wolpert, Stacked generalization. Neural Networks 1992.

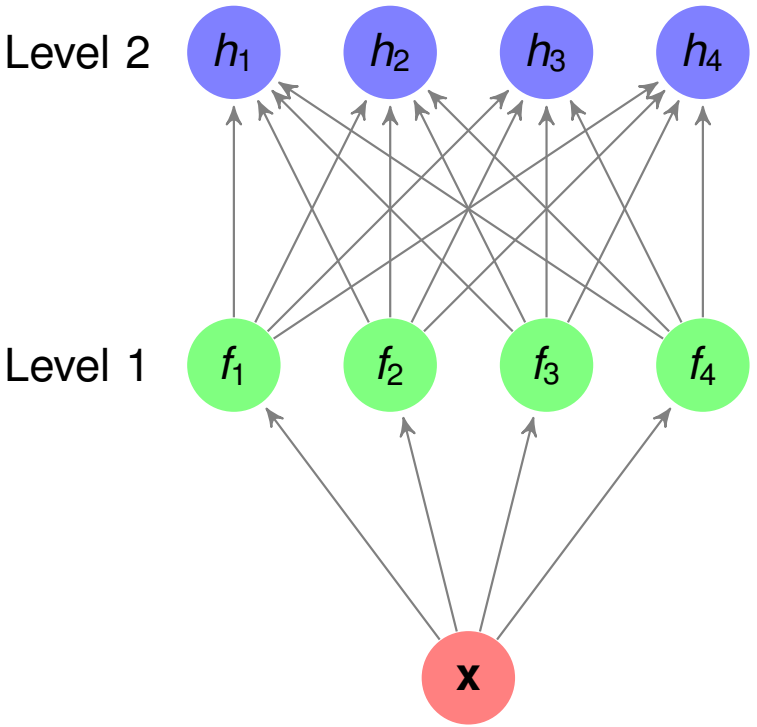
STACKING APPLIED TO MTP

- Level 1 learners: learn a model for every target independently

$\rightsquigarrow$ $f_1, \ldots, f_m$

- Level 2 learner: learn again a model for every target independently, using the predictions of the first step as features

$\rightsquigarrow$ $f(\mathbf{x}) = h(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$
Or alternatively:
$f(\mathbf{x}) = h(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}), \mathbf{x})$

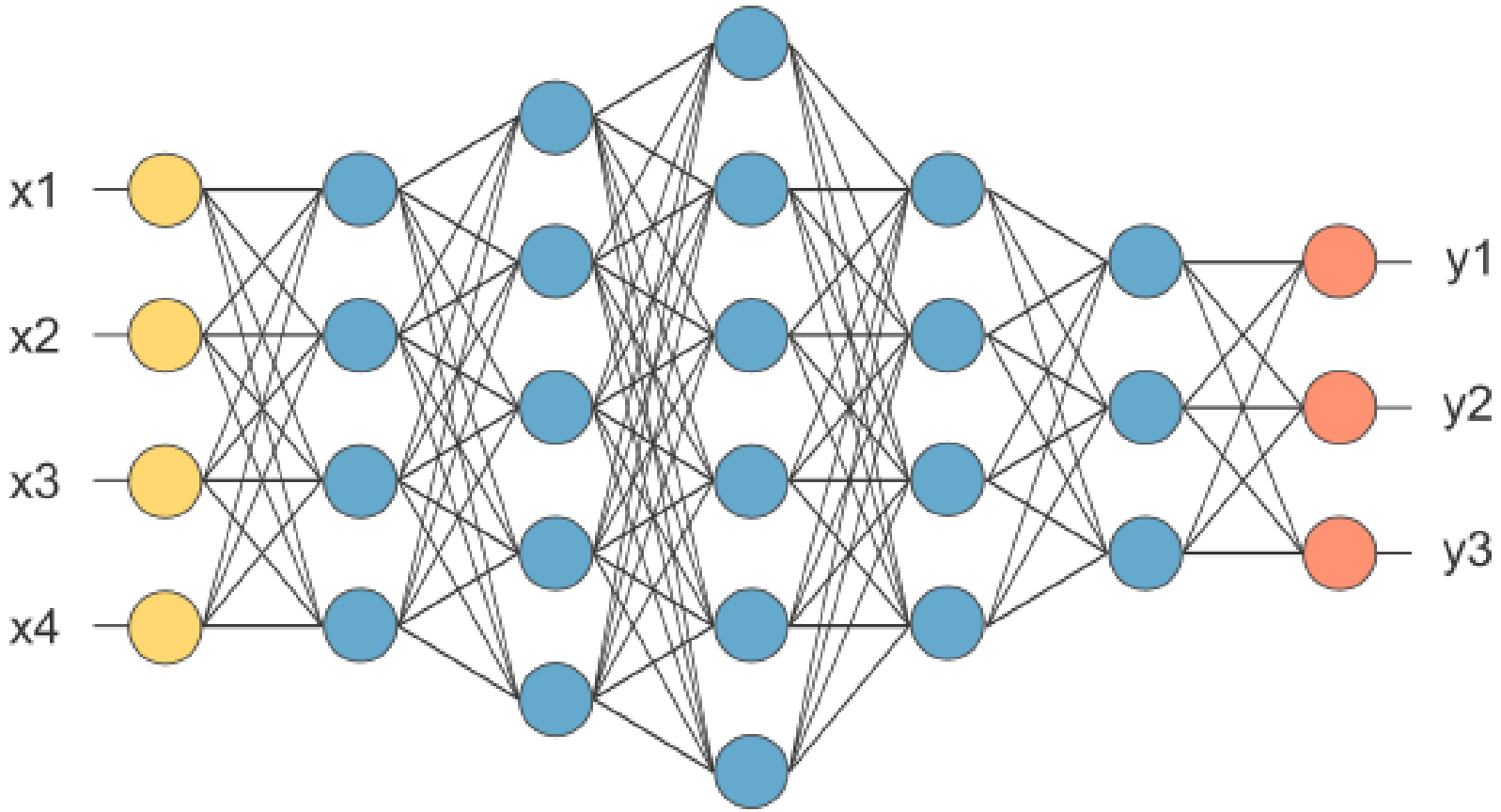


- Advantages: Easy to implement and general (usable with any level-1 learner).
- Has been shown to avoid overfitting in multivariate regression.
- If level 2 learner uses regularization $\rightsquigarrow$ models are forced to learn similar parameters for different targets.

Cheng and Hüllermeier, Combining Instance-based learning and Logistic Regression for Multi-Label classification, Machine Learning, 2009.

ENFORCING SIMILARITY IN (DEEP) NEURAL NETWORKS

Commonly-used architecture: weight sharing in the final layer with *m* nodes, i.e., weight sharing among the targets



Caruana, Multitask learning: A knowledge-based source of inductive bias. Machine Learning 1997.

Similarity-exploiting methods

KRONECKER KERNEL RIDGE REGRESSION

- In the case of multi-target prediction with target features one typically uses kernel methods for learning.
- In particular, one considers 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 some feature map for the features and ψ is a feature map for the target (features) and $\otimes$ is the Kronecker product.
- This leads to the 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 the kernel for the feature map ϕ, g the kernel for the feature map ψ and $\alpha_{(\mathbf{x}', \mathbf{t}')}$ are the dual parameters, which can be found by least-squares minimization:

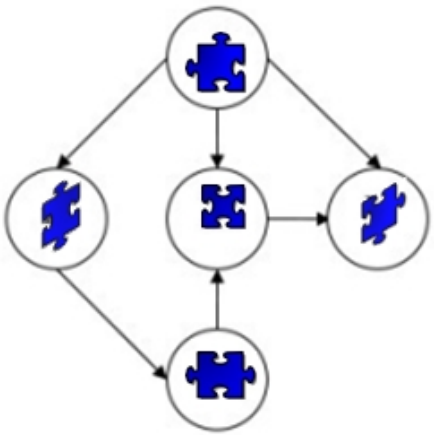
$$\min_{\boldsymbol{\alpha}} \; ||\boldsymbol{\Gamma}\boldsymbol{\alpha} - \boldsymbol{z}||_2^2 + \lambda \boldsymbol{\alpha}^\top \boldsymbol{\Gamma} \boldsymbol{\alpha},$$

  where $\boldsymbol{z} = \mathrm{vec}(Y)$.
- This approach is commonly used in the zero-shot learning framework.

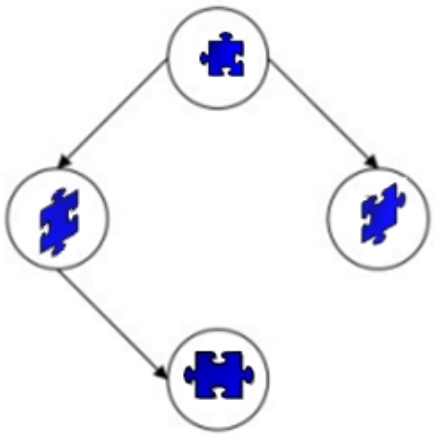
Stock et al., A comparative study of pairwise learning methods based on kernel ridge regression, Neural Computation 2018.

EXPLOITING RELATIONS IN REGULARIZATION TERMS

Graph



Tree



Similarity

	1	0.26	0.26	0.04
	0.26	1	0.7	0.57
	0.26	0.7	1	0.44
	0.04	0.57	0.44	1

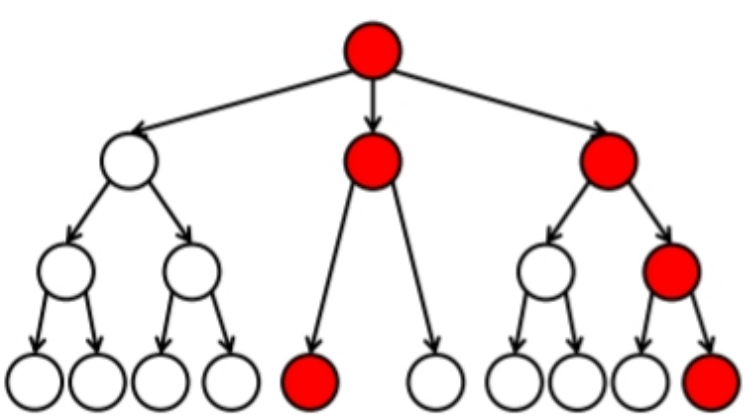
- Graph-based regularization is an approach that can be applied to the tree types of relations in the targets:

$$\min_A \| Y - XA \|_F^2 + \lambda \sum_{j=1}^{m} \sum_{j' \in \mathcal{N}(j)} \| \mathbf{a}_j - \mathbf{a}_{j'} \|^2,$$

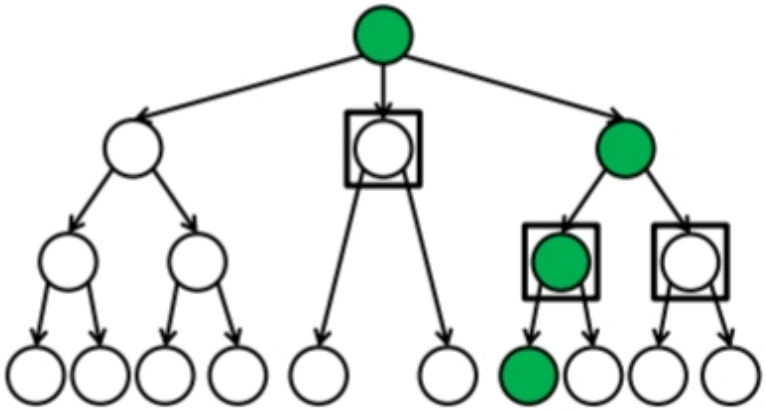
  where $\mathcal{N}(j)$ is the set of targets that are related to target j.
- Can also be used in a weighted version taking the similarities (or correlations) into account.

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

HIERARCHICAL MULTI-LABEL CLASSIFICATION



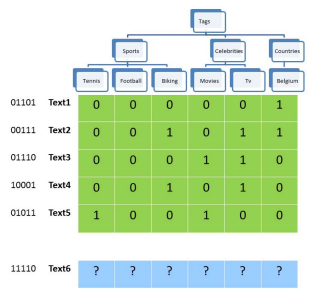
(a) Ground truth.



(b) Prediction A.

- In addition to performance gains in general, hierarchies can also be used to define specific loss functions, such as the Hierarchy-loss:

$$\ell_{Hier}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{j: y_j \neq \hat{y}_j} c_j \, \mathbb{1}_{[anc(y_j) = anc(\hat{y}_j)]},$$



where c_j are costs depending on the depth of node j.

- 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.

Similarity-constructing methods

PROBABILISTIC CLASSIFIER CHAINS

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

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

- Repeatedly apply the *product rule* of probability:

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

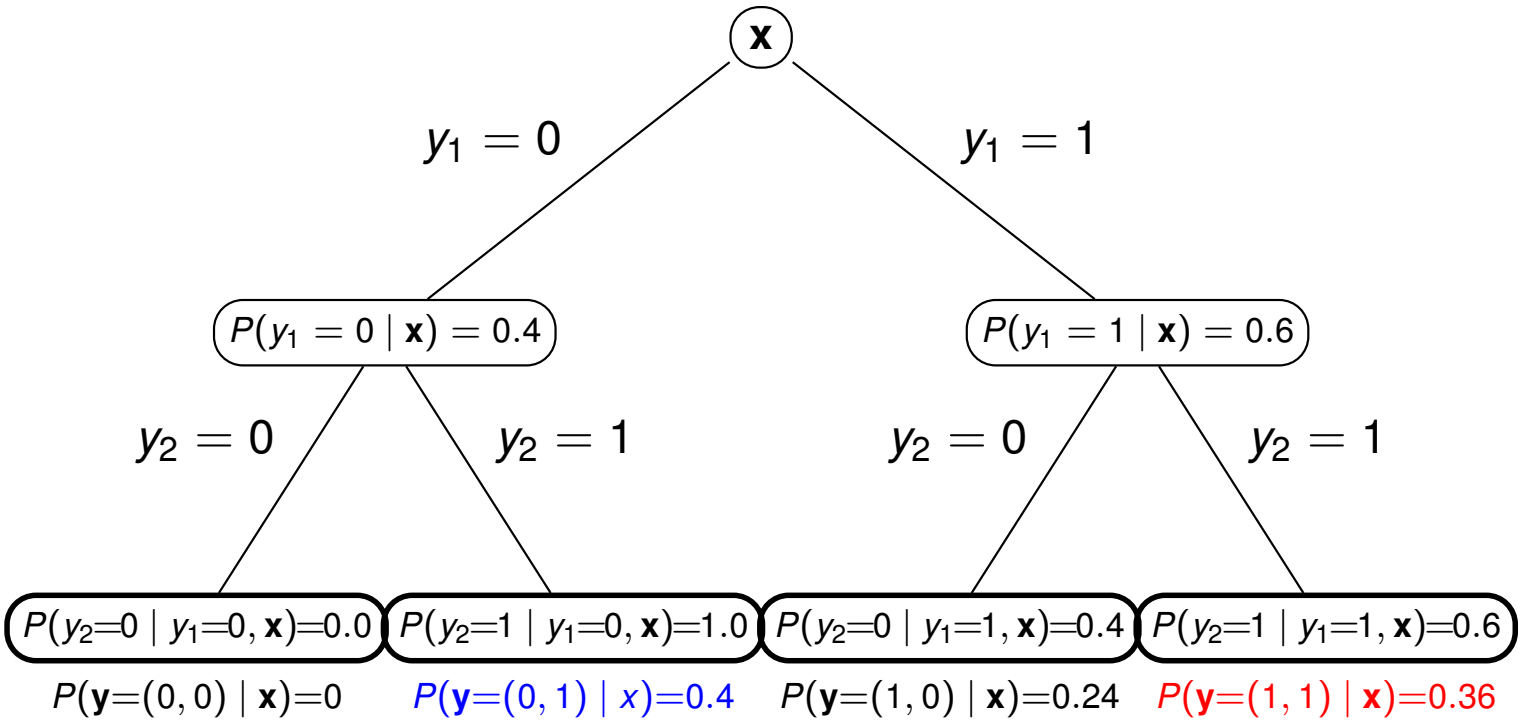
- Learning relies on constructing probabilistic classifiers for estimating

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

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

PROBABILISTIC CLASSIFIER CHAINS

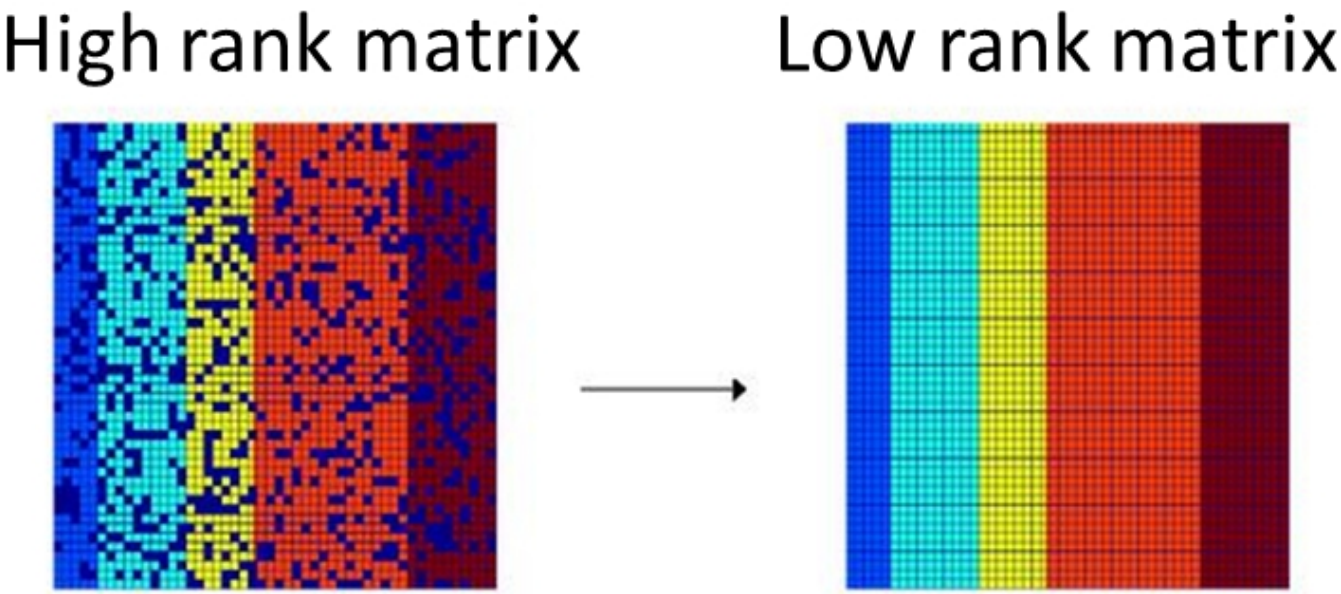
- Inference relies on exploiting a probability tree:



- For subset 0/1 loss one needs to find $f^*(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}^m} \mathbb{P}(\mathbf{y} \mid \mathbf{x})$.
- 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 materializes the idea that some structure is shared across different targets.
- Typically perform a low-rank approximation of the parameter matrix:

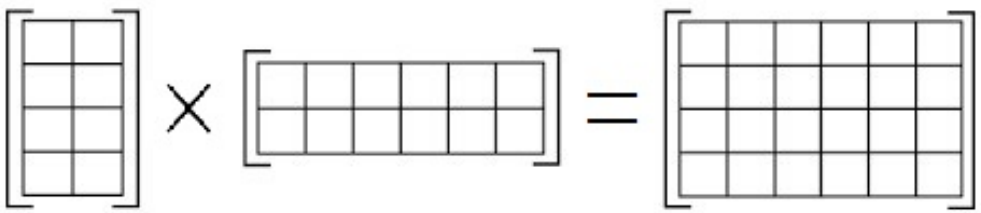
$$\min_A \|Y - \mathbf{X}A\|_F^2 + \lambda \operatorname{rank}(A)$$

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

LOW-RANK APPROXIMATION

- A: parameter matrix of dimensionality $p \times m$
- p: the number of features
- m: the number of targets
- Assume a low-rank structure of A:

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



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

LOW-RANK APPROXIMATION: OVERVIEW OF METHODS

- Popular for multi-output regression, multi-task learning and multi-label classification.
- Linear as well as nonlinear methods.
- Algorithms:
  - Principal component analysis, Canonical correlation analysis, Partial least squares.
  - Singular value decomposition, Alternating structure optimization.
  - Compressed sensing, Output codes, Landmark labels, Bloom filters, Auto-encoders.