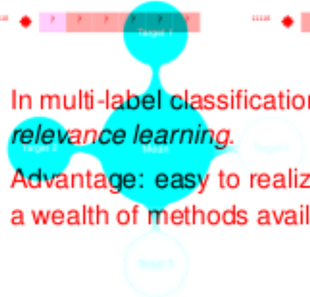
Advanced Machine Learning

Multi-Target Prediction: Methods Part 1



Learning goals

- Independent models for targets
- Mean regularization
- Stacking
- Weight sharing in DL

INDEPENDENT MODELS

- The most naive way to make multi-target predictions: learning a model for each target independently.

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

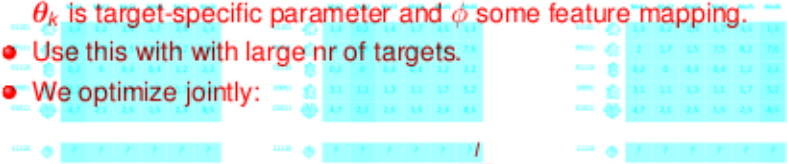
INDEPENDENT MODELS

- Assume a linear basis function model for the m-th target:

$$f_k(\mathbf{x}) = \boldsymbol{\theta}_k^\top \phi(\mathbf{x}),$$

$\boldsymbol{\theta}_k$ is target-specific parameter and ϕ some feature mapping.

- Use this with with large nr of targets.
- We optimize jointly:



$$\min_{\Theta} \|Y - \Phi\Theta\|_F^2 + \sum_{m=1}^{l} \lambda_m \|\boldsymbol{\theta}_m\|^2,$$

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

$$\Phi = \begin{bmatrix} \phi(\mathbf{x}^{(1)})^\top \\ \vdots \\ \phi(\mathbf{x}^{(n)})^\top \end{bmatrix} \qquad \Theta = [\boldsymbol{\theta}_1 \quad \cdots \quad \boldsymbol{\theta}_l].$$

Frobenius norm = sum of SSE-s of all targets

- The most naive way to make multi-target predictions: learning a model for each target independently.
- In multi-label classification this approach is also known as *binary relevance learning*.
- Advantage: easy to realize, as for single-target prediction we have a wealth of methods available.

INDEPENDENT MODELS

The experimental results section of a typical MTP paper:

- Assume a linear basis function model for the k-th target:

$$f_k(\mathbf{x}) = \boldsymbol{\theta}_k^\top \phi(\mathbf{x}),$$

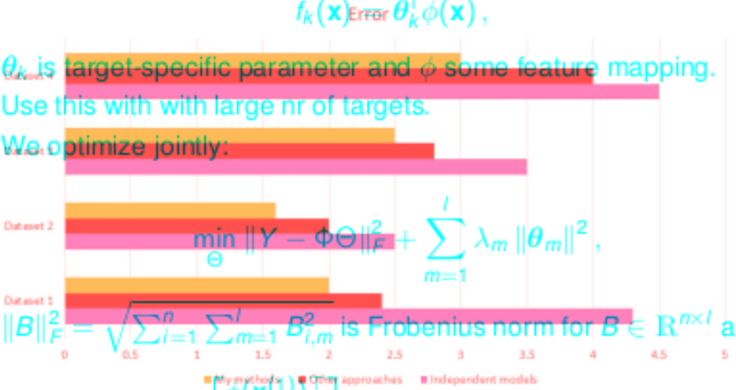
$\boldsymbol{\theta}_k$ is target-specific parameter and ϕ some feature mapping.

- Use this with with large nr of targets.
- We optimize jointly:

$$\min_{\Theta} \|Y - \Phi\Theta\|_F^2 + \sum_{m=1}^{l} \lambda_m \|\boldsymbol{\theta}_m\|^2,$$

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

$$\Phi = \begin{bmatrix} \phi(\mathbf{x}^{(1)})^\top \\ \vdots \\ \phi(\mathbf{x}^{(n)})^\top \end{bmatrix}, \quad \Theta = [\boldsymbol{\theta}_1 \; \cdots \; \boldsymbol{\theta}_l].$$



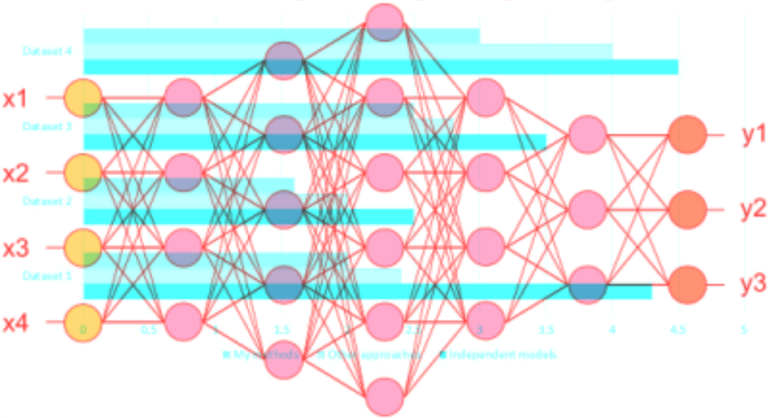
$\rightsquigarrow$ Independent models don't exploit target deps, compared to more sophisticated methods, seems to be key for better performance.

Frobenius norm = sum of SSE-s of all targets

ENFORCING SIMILARITY IN DEEP LEARNING

The experimental results section of a typical MTP paper:

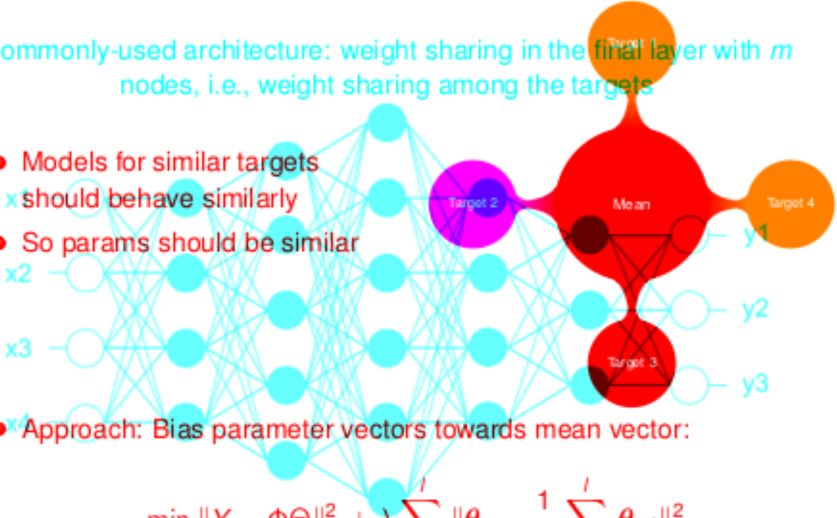
Commonly-used architecture: weight sharing in the final layer with m nodes, i.e., weight sharing among the targets



⇝ Independent models don't exploit target deps, compared to more sophisticated methods, seems to be key for better performance.

MEAN-REGULARIZED MULTI-TASK LEARNING

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



- Models for similar targets should behave similarly
- So params should be similar
- Approach: Bias parameter vectors towards mean vector:

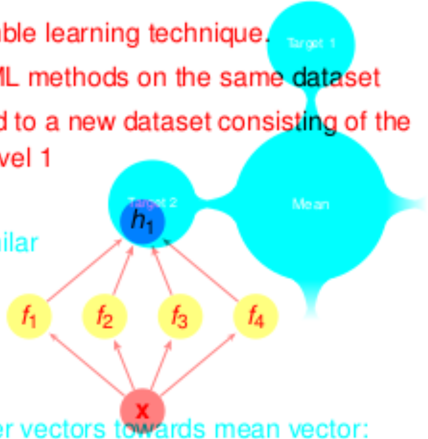
$$\min_{\Theta} \|Y - \Phi\Theta\|_F^2 + \lambda \sum_{m=1}^{l} \|\theta_m - \frac{1}{l}\sum_{m'=1}^{l} \theta_{m'}\|^2$$

▸ Caruana, 1997

▸ Evgeniou and Pontil, 2004

STACKING

- Originally, general ensemble learning technique.
- Level 1: apply series of ML methods on the same dataset
- Level 2: apply ML method to a new dataset consisting of the predictions obtained at level 1



Wolpert, 1992

REGULARIZED MULTI-TASK LEARNING

- Models for similar targets should behave similarly
- So params should be similar
- Approach: Bias parameter vectors towards mean vector:

$$\min_{\Theta} \|Y - \Phi\Theta\|_F^2 + \lambda \sum_{m=1}^{l} \|\theta_m - \frac{1}{l}\sum_{m'=1}^{l} \theta_{m'}\|^2$$

Evgeniou and Pontil, 2004

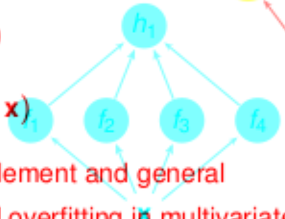
STACKING APPLIED TO MTP

- Level 1: learn all $f_k(\mathbf{x})$ independently
- Level 2: learn model for each target independently, using predictions of level 1 $\rightsquigarrow$
  $f(\mathbf{x}) = g(f_1(\mathbf{x}), \ldots, f_l(\mathbf{x}))$
  Or:
  $f(\mathbf{x}) = g(f_1(\mathbf{x}), \ldots, f_l(\mathbf{x}), \mathbf{x})$

- Advantages: easy to implement and general
- 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, 2009

- Originally, general ensemble learning technique.
- Level 1: apply series of ML methods on the same dataset
- Level 2: apply ML method to a new dataset consisting of the predictions obtained at level 1



▸ Wolpert, 1992

STACKING VS BINARY RELEVANCE: EXAMPLE

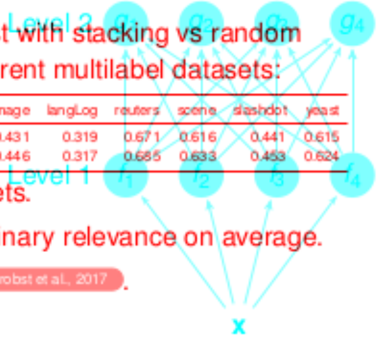
- Compare F1-Score of random forest with stacking vs random forest with binary relevance on different multilabel datasets:

	birds	emotions	enron	genbase	image	langLog	reuters	scene	slashdot	yeast
BR(rf) F1-Score	0.637	0.620	0.578	0.989	0.431	0.319	0.671	0.616	0.441	0.615
STA(rf) F1-Score	0.646	0.634	0.583	0.986	0.446	0.317	0.685	0.633	0.453	0.624

- F1-Score is decomposed over targets.
- NB: Stacking slightly outperforms binary relevance on average.
- For more details, please refer to ▸ Probst et al., 2017.

- Level 1: learn all $f_k(\mathbf{x})$ independently
- Level 2: learn model for each target independently, using predictions of level 1: $f(\mathbf{x}) = g(f_1(\mathbf{x}), \ldots, f_l(\mathbf{x}))$

  Or: $f(\mathbf{x}) = g(f_1(\mathbf{x}), \ldots, f_l(\mathbf{x}), \mathbf{x})$



- Advantages: easy to implement and general
- Has been shown to avoid overfitting in multivariate regression
- If level 2 learner uses regularization ⇝ models are forced to learn similar parameters for different targets.

▸ Cheng and Hüllermeier, 2009

STACKING VS BINARY RELEVANCE: EXAMPLE

- Compare F1-Score of random forest with stacking vs random forest with binary relevance on different multilabel datasets:

	birds	emotions	enron	genbase	image	langLog	reuters	scene	slashdot	yeast
BR(rf) F1-Score	0.637	0.620	0.578	0.989	0.431	0.319	0.671	0.616	0.441	0.615
STA(rf) F1-Score	0.646	0.634	0.583	0.986	0.446	0.317	0.685	0.633	0.453	0.624

- F1-Score is decomposed over targets.
- NB: Stacking slightly outperforms binary relevance on average.
- For more details, please refer to ▸ Probst et al., 2017 .