INDEPENDENT MODELS

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

	Mol1	Mol2	Mol3	Mol4	Mol5	Mol6
01101	1,3	0,2	1,4	1,7	3,5	1,3
00111	2	1,7	1,5	7,5	8,2	7,6
01110	0,2	0	0,3	0,4	1,2	2,2
10001	3,1	1,1	1,3	1,1	1,7	5,2
01011	4,7	2,1	2,5	1,5	2,3	8,5
11110	?	?	?	?	?	?

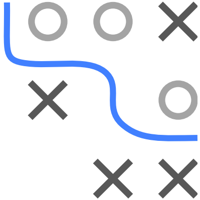
	Mol1	Mol2	Mol3	Mol4	Mol5	Mol6
01101	1,3	0,2	1,4	1,7	3,5	1,3
00111	2	1,7	1,5	7,5	8,2	7,6
01110	0,2	0	0,3	0,4	1,2	2,2
10001	3,1	1,1	1,3	1,1	1,7	5,2
01011	4,7	2,1	2,5	1,5	2,3	8,5
11110	?	?	?	?	?	?

. . .

	Mol1	Mol2	Mol3	Mol4	Mol5	Mol6
01101	1,3	0,2	1,4	1,7	3,5	1,3
00111	2	1,7	1,5	7,5	8,2	7,6
01110	0,2	0	0,3	0,4	1,2	2,2
10001	3,1	1,1	1,3	1,1	1,7	5,2
01011	4,7	2,1	2,5	1,5	2,3	8,5
11110	?	?	?	?	?	?

- 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:

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

  $\boldsymbol{\theta}_k$ is target-specific parameter and ϕ some feature mapping.
- Use this with with large nr of targets.
- We optimize jointly:

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

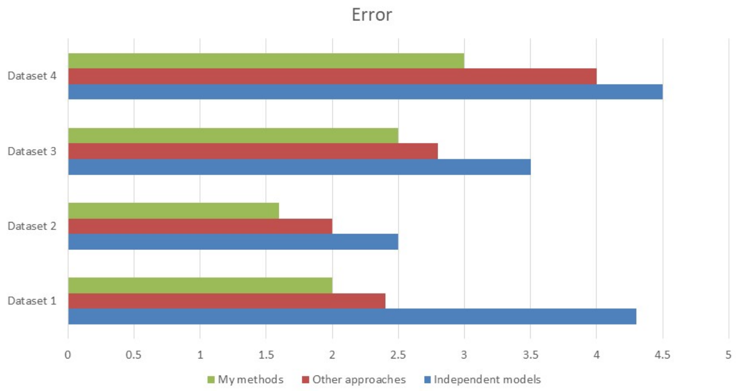
$\|B\|_F^2 = \sqrt{\sum_{i=1}^{n} \sum_{m=1}^{l} B_{i,m}^2}$ is Frobenius norm for $B \in \mathbb{R}^{n \times l}$ and

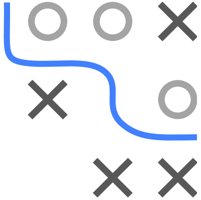
$$\Phi = \begin{bmatrix} \phi(\mathbf{x}^{(1)})^\top \\ \vdots \\ \phi(\mathbf{x}^{(n)})^\top \end{bmatrix} \qquad \Theta = \begin{bmatrix} \boldsymbol{\theta}_1 & \cdots & \boldsymbol{\theta}_l \end{bmatrix} .$$

Frobenius norm = sum of SSE-s of all targets

INDEPENDENT MODELS

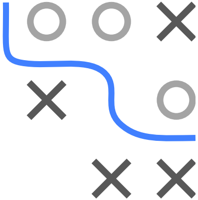
The experimental results section of a typical MTP paper:



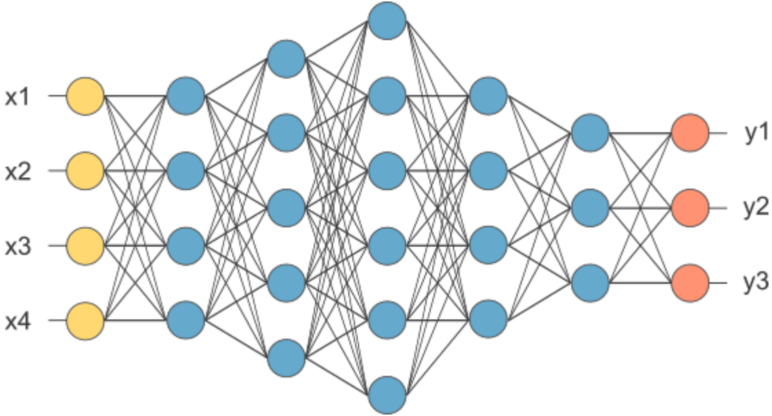


⇝ Independent models don't exploit target deps, compared to more sophisticated methods, seems to be key for better performance.

ENFORCING SIMILARITY IN DEEP LEARNING

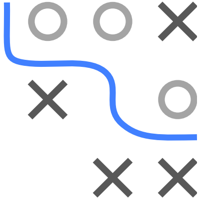


Commonly-used architecture: weight sharing in the final layer with m nodes, i.e., weight sharing among the targets

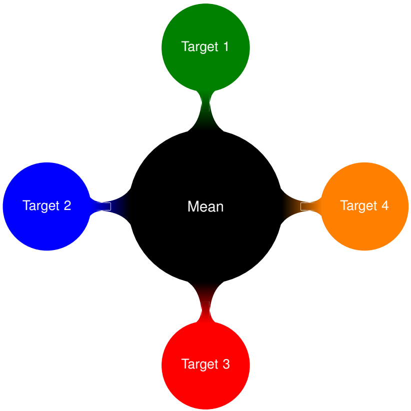


Caruana, 1997

MEAN-REGULARIZED MULTI-TASK LEARNING



- Models for similar targets should behave similarly
- So params should be similar



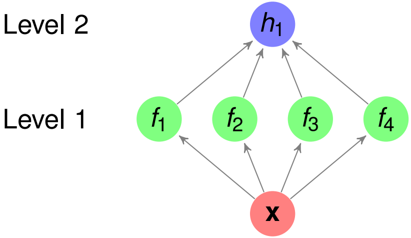
- Approach: Bias parameter vectors towards mean vector:

$$\min_{\Theta} \|Y - \Phi\Theta\|_F^2 + \lambda \sum_{m=1}^{l} \|\boldsymbol{\theta}_m - \frac{1}{l} \sum_{m'=1}^{l} \boldsymbol{\theta}_{m'}\|^2$$

▸ Evgeniou and Pontil, 2004

STACKING

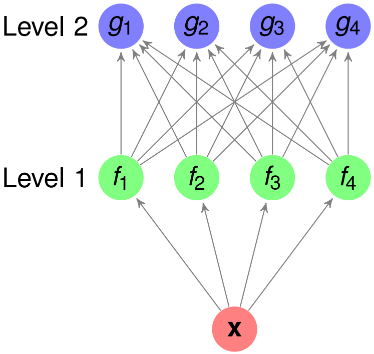
- Originally, general ensemble learning technique.
- Level 1: apply series of ML methods on the same dataset
- Level 2: apply ML method to a new dataset consisting of the predictions obtained at level 1

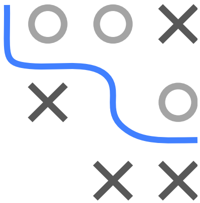


▸ Wolpert, 1992

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





- 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

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.