Multi-Target Prediction: A Unifying View on Problems and Methods

Willem Waegeman

Research Unit Knowledge-based Systems (KERMIT)
Department of Mathematical Modelling, Statistics and Bioinformatics
Ghent University, Belgium

October 13 2017

Oranje verslaat Zweden, maar gaat niet naar WK

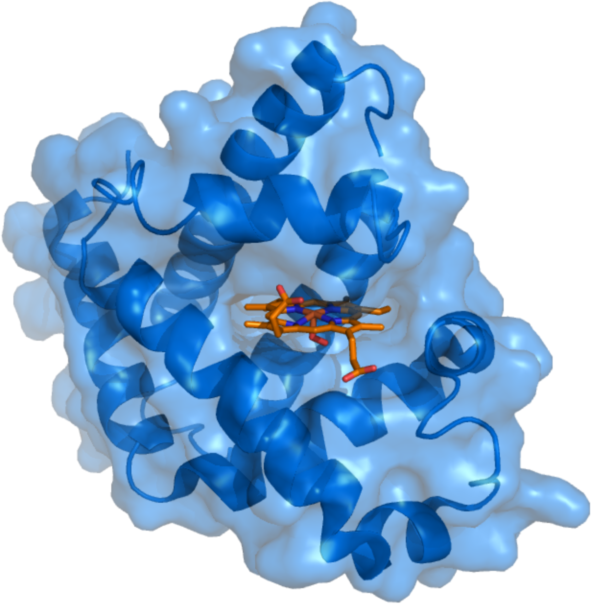
DINSDAG, 22:38 VOETBAL

Het Nederlands elftal heeft zich zoals verwacht niet geplaatst voor de play-offs om de laatste vier Europese WK-tickets. Oranje won in Amsterdam met 2-0 van Zweden, maar het verschil had zeven of meer doelpunten moeten zijn.

Multi-label classification: the example of document categorization

		Tennis	Football	Biking	Movies	TV	Belgium
01101	**Text1**	0	1	0	0	1	1
00111	**Text2**	1	0	0	0	0	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**	?	?	?	?	?	?



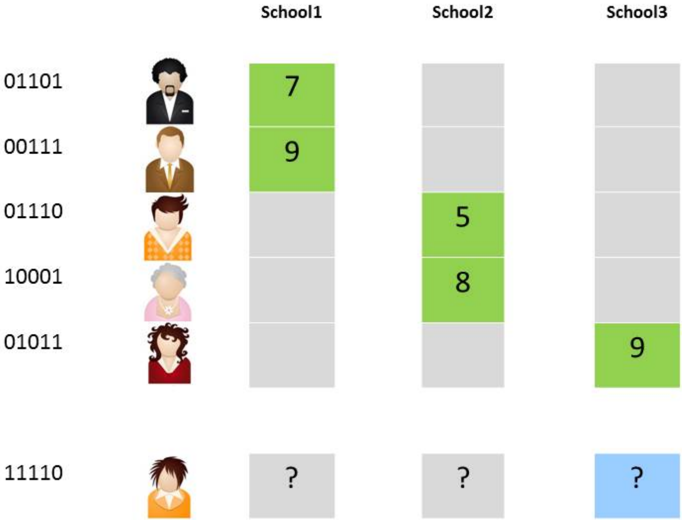
Multivariate regression: the example of protein-ligand interaction prediction

	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	?	?	?	?	?	?


SMITHFIELD S.H.S.
STATE HIGH SCHOOL
SUNNYBANK
MANY KINDS OF EXCELLENCE
FERNY GROVE STATE HIGH SCHOOL
ALWAYS AIM HIGH
BRISBANE STATE HIGH SCHOOL
SCIENTIA est POTESTAS
ELANORA STATE HIGH SCHOOL
URANGAN S.H.S.
WORKING TOGETHER
CENTENARY HEIGHTS
CLEVELAND
HIGH
SEMPER DIGNE
ALWAYS AIMING HIGHER
MIAMI
SEMPER FLOREAT COSTA
KEDRON
HIGH SCHOOL
NITI EST NITERE
TRINITY BAY
ENDEAVOUR
YERONGA
SCIENTIA AC HUMANITATE
SUNSHINE BEACH HIGH
QUALITY IN ALL WE DO
MERRIMAC STATE HIGH SCHOOL
PRIDE IN EXCELLENCE
ENDEAVOUR • DISCOVER • SUCCEED
The Cathedral School
TOWNSVILLE
ROCHEDALE
SHS
INTEGRITY AND ACHIEVEMENT
RECTE CRESCAM
S.H.S
CORINDA
TRUTH
FAITH
COMPASSION
MAROOCHYDORE
HIGH SCHOOL
NITI ET SERVIRE
STATE HIGH SCHOOL
CAIRNS

Multi-task learning: the example of predicting student marks

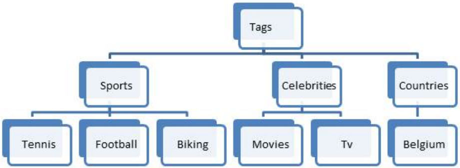


There are a lot of multi-target prediction problems around...

Overview of this talk

1. A unifying view on MTP problems
2. A unifying view on MTP methods
3. Loss functions in multi-target prediction
4. Conclusions

Let's assume a document hierarchy: How would you call this machine learning problem?



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**	?	?	?	?	?	?

Let's assume a target representation:
How would you call this machine learning problem?

	0011 School1	1100 School2	0110 School3
01101	7		
00111	9		
01110		5	
10001		8	
01011			9
11110	?	?	?

Let's assume a target representation:
How would you call this machine learning problem?

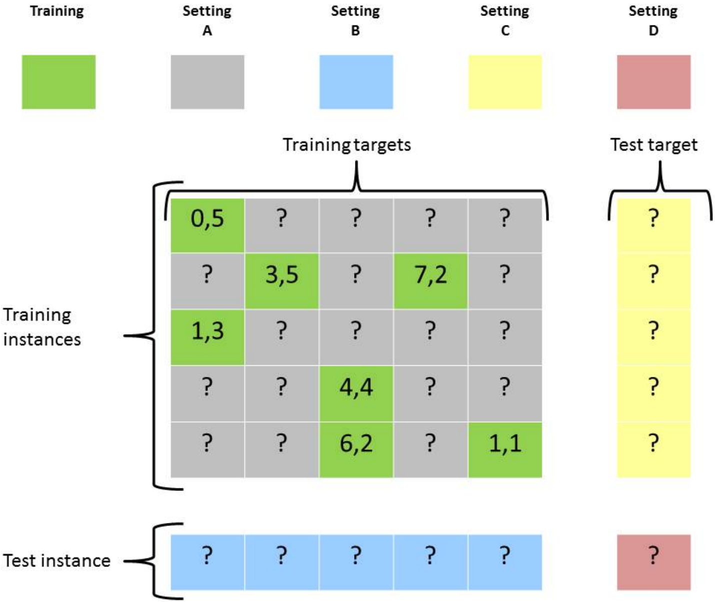
	Mol1	Mol2	Mol3	Mol4	Mol5	Mol6
01101	1,3	0,2	1,4	1,7	3,5	1,3
00111	2	1,7	1,5	7,5	8,2	7,6
01110	0,2	0	0,3	0,4	1,2	2,2
10001	3,1	1,1	1,3	1,1	1,7	5,2
01011	4,7	2,1	2,5	1,5	2,3	8,5
11110	?	?	?	?	?	?

Generalizing to new targets

$g(.,.)$: target similarity

	Mol1	Mol2	Mol3	Mol4	Mol5	Mol6	Mol7
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	?	?	?	?	?	?	?

Important subdivision of different learning settings



General framework

Definition

A multi-target prediction setting is characterized by instances $\boldsymbol{x} \in \mathcal{X}$ and targets $\boldsymbol{t} \in \mathcal{T}$ with the following properties:

1. A training dataset consists of triplets $(\boldsymbol{x}_i, \boldsymbol{t}_j, y_{ij})$, where $y_{ij} \in \mathcal{Y}$.
2. In total n instances and m targets are observed during training, with n and m finite numbers.
3. As such, the scores y_{ij} of the training data can be arranged in an $n \times m$ matrix Y.
4. The score set $\mathcal{Y}$ is one-dimensional. It consists of nominal, ordinal or real values.
5. The goal consists of making predictions for any instance-target couple $(\boldsymbol{x}, \boldsymbol{t}) \in \mathcal{X} \times \mathcal{T}$.

Overview of this talk

1. A unifying view on MTP problems
2. A unifying view on MTP methods
3. Loss functions in multi-target prediction
4. Conclusions

A unifying view on MTP methods

Group of methods	Applicable setting
Independent models	B and C
Similarity-enforcing methods	B and C
Relation-exploiting methods	B, C and D
Relation-constructing methods	B and C
Representation-exploiting methods	B, C and D
Representation-constructing methods	A, B and C

A baseline method:
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	?	?	?	?	?	?

A baseline method: 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	?	?	?	?	?	?

A baseline: Independent Models

	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	?	?	?	?	?	?

A baseline: Independent Models

Linear basis function model for i-th target:

$$f_i(\boldsymbol{x}) = \boldsymbol{a}_i^\intercal \phi(\boldsymbol{x}) \,,$$

Solving as a joint optimization problem:

$$\min_A ||Y - XA||_F^2 + \sum_{i=1}^{m} \lambda_i \, ||\boldsymbol{a}_i||^2 \,,$$

With the following notations:

$$X = \begin{bmatrix} \phi(\boldsymbol{x}_1)^T \\ \vdots \\ \phi(\boldsymbol{x}_n)^T \end{bmatrix} \qquad A = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_m \end{bmatrix} .$$

Hamming loss as alternative for binary labels:

$$L_{\mathsf{Ham}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{j=1}^{m} I(y_j = \hat{y}_j)$$

Learning a model for each target independently is still state-of-the-art in extreme multi-label classification[1]:

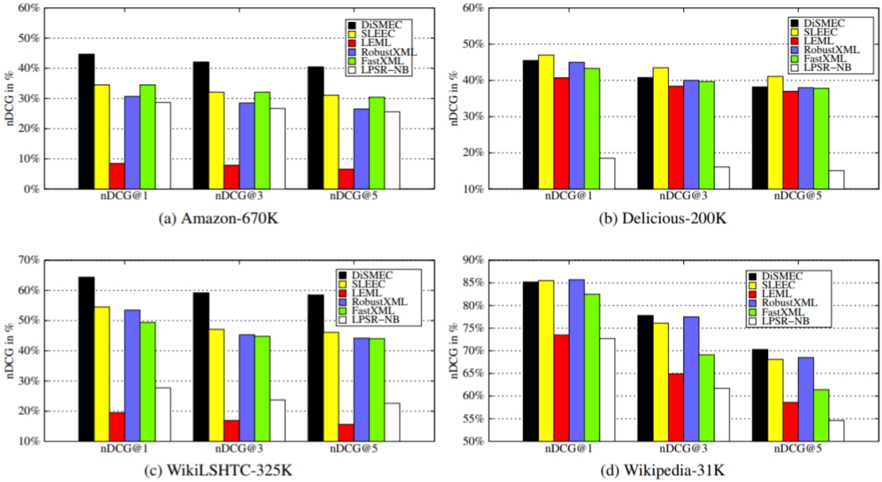


Figure 3: nDCG@k for k=1, 3 and 5

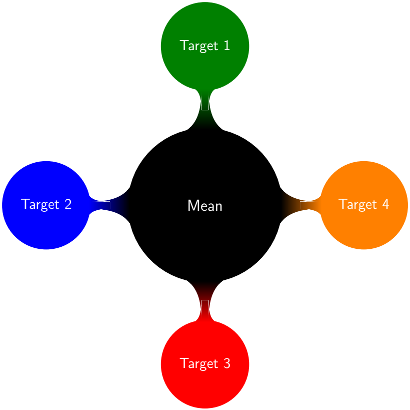
[1] Babbar and Schölkopf, DISMEC: Distributed Sparse Machines for Extreme Multi-label classification, WSDM 2017

A unifying view on MTP methods

Group of methods	Applicable setting
Independent models	B and C
Similarity-enforcing methods	B and C
Relation-exploiting methods	B, C and D
Relation-constructing methods	B and C
Representation-exploiting methods	B, C and D
Representation-constructing methods	A, B and C

Mean-regularized multi-task learning[2]

- **Simple assumption**: models for different targets are related to each other.
- **Simple solution**: the parameters of these models should have similar values.
- **Approach**: bias the parameter vectors towards their mean vector.



$$\min_A ||Y - XA||_F^2 + \lambda \sum_{i=1}^{m} ||\boldsymbol{a}_i - \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{a}_j||^2 \,,$$

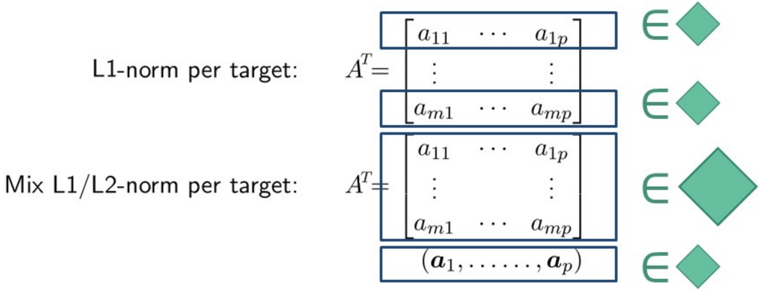
[2] Evgeniou and Pontil, Regularized multi–task learning, KDD 2004.

Joint feature selection

- Enforce that the same features are selected for different targets[3]:

$$\min_{A} ||Y - XA||_F^2 + \lambda \sum_{j=1}^{p} ||\boldsymbol{a}_j||^2$$

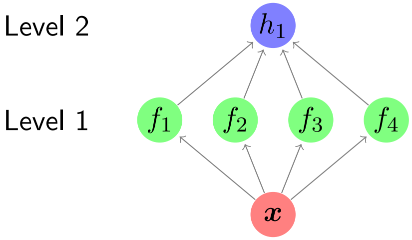
- The vectors $\boldsymbol{a}_j$ now represent the rows of matrix A^T:



[3] Obozinski et al. Joint covariate selection and joint subspace selection for multiple classification problems. Statistics and Computing 2010

Stacking (Stacked generalization)

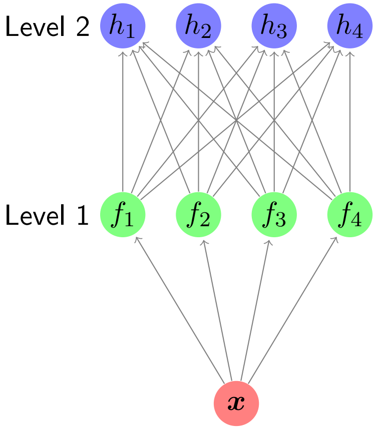
- Originally introduced as a general ensemble learning or blending technique.[4]
- Level 1 classifiers: apply a series of ML methods on the same dataset (or, one ML method on bootstrap samples of the dataset)
- Level 2 classifier: apply an ML method to a new dataset consisting of the predictions obtaining at Level 1



[4] Wolpert, Stacked generalization. Neural Networks 1992

Stacking applied to multi-target prediction[5]

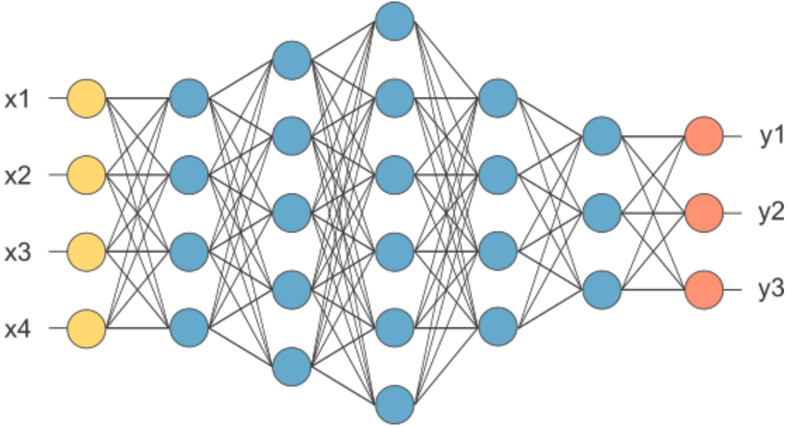
- Level 1 classifiers: learn a model for every target independently
- Level 2 classifier: learn again a model for every target independently, using the predictions of the first step as features



[5] Cheng and Hüllermeier, Combining Instance-based learning and Logistic Regreession for Multi-Label classification, Machine Learning, 2009

MTP in (Deep) Neural Networks

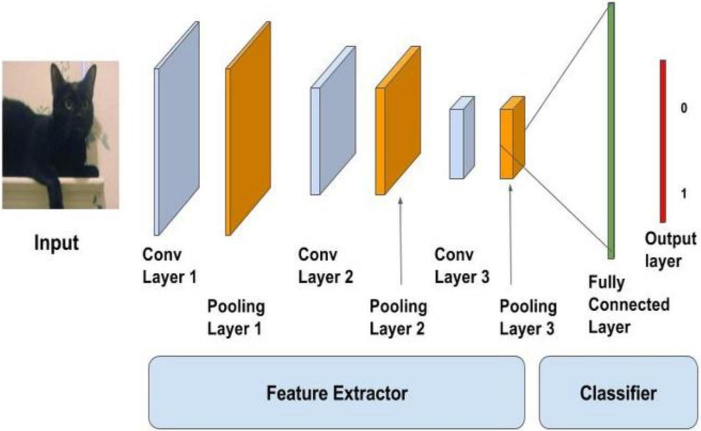
Commonly-used architecture: weight sharing among targets[6]



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

Re-using Pretrained Models in (Deep) Neural Networks

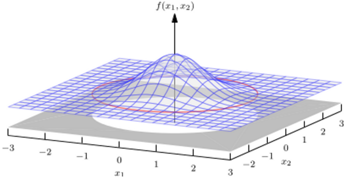
Commonly-used training method: first train on targets that have a lot of observations, only train some parameters for targets that have few observations [7]



[7] Keras Tutorial: Transfer Learning using pre-trained models

An intuitive explanation: James-Stein estimation

- Consider a multivariate normal distribution $\boldsymbol{y} \sim N(\boldsymbol{\theta}, \sigma^2 \mathbf{I})$.

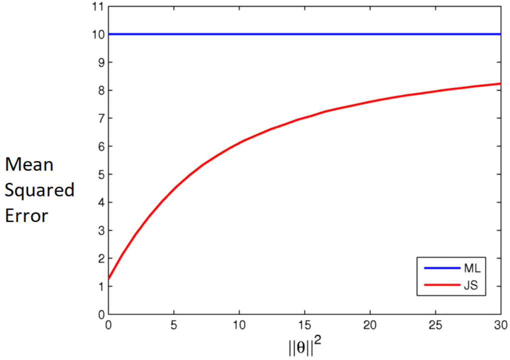


- What is the best estimator of the mean vector $\boldsymbol{\theta}$?
- Evaluation w.r.t. MSE: $\mathbb{E}[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^2]$
- Single-observation maximum likelihood estimator: $\hat{\boldsymbol{\theta}}^{\mathrm{ML}} = \boldsymbol{y}$
- James-Stein estimator[8]:

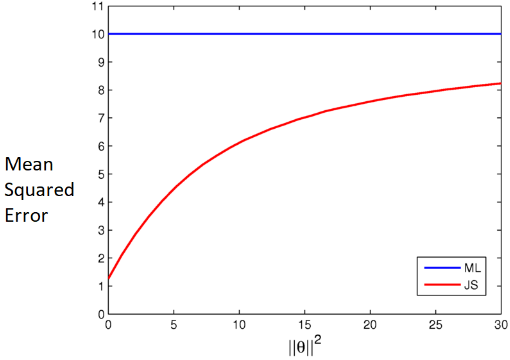
$$\hat{\theta}^{\mathrm{JS}} = \left(1 - \frac{(m-2)\sigma^2}{\|\boldsymbol{y}\|^2}\right) \boldsymbol{y}$$

[8] W. James and C. Stein. Estimation with quadratic loss. In Proc. Fourth Berkeley Symp. Math. Statist. Prob. 1, pages 361-379, 1961

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



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



- Regularization towards other directions is also possible:

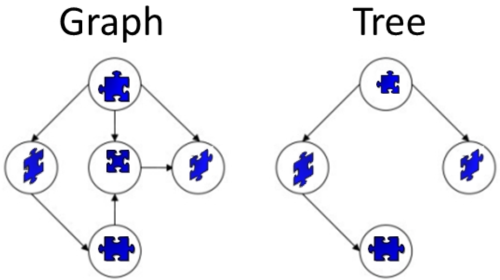
$$\hat{\theta}^{\mathrm{JS+}} = \left(1 - \frac{(m-2)\sigma^2}{\|\boldsymbol{y} - \boldsymbol{v}\|^2}\right)(\boldsymbol{y} - \boldsymbol{v}) + \boldsymbol{v}$$

- Only outperforms the maximum likelihood estimator w.r.t. the sum of squared errors over all components.

A unifying view on MTP methods

Group of methods	Applicable setting
Independent models	B and C
Similarity-enforcing methods	B and C
Relation-exploiting methods	B, C and D
Relation-constructing methods	B and C
Representation-exploiting methods	B, C and D
Representation-constructing methods	A, B and C

Exploiting relations in regularization terms



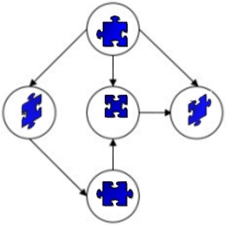
Similarity

	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

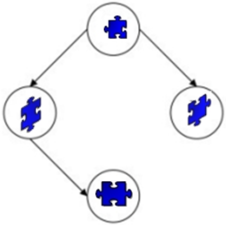
[9] Gopal and Yang, Recursive regularization for large-scale classification with hierarchical and graphical dependencies, KDD 2013

Exploiting relations in regularization terms

Graph



Tree



Similarity

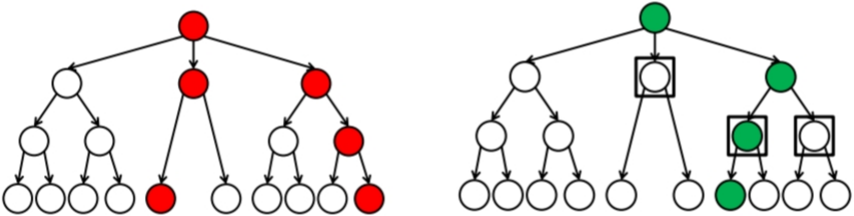
	1	0.26	0.26	0.04
	0.26	1	0.7	0.57
	0.26	0.7	1	0.44
	0.04	0.57	0.44	1

Graph-based regularization is an approach that can be applied to the three types of relations[9]:

$$\min_{A} ||Y - XA||_F^2 + \lambda \sum_{i=1}^{m} \sum_{j \in \mathcal{N}(i)} ||\boldsymbol{a}_i - \boldsymbol{a}_j||^2$$

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

Hierarchical multi-label classfication



(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 H-loss[10]:

$$L_H(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{i: y_i \neq \hat{y}_i} c_i \, I(\mathit{anc}(y_i) = \mathit{anc}(\hat{y}_i))$$

c_i depends on the depth of node i

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

Exploiting similarity measures among targets

	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

Can be done within the framework of vector-valued kernel functions[11]:

$$f(\boldsymbol{x}, \boldsymbol{t}) = \boldsymbol{w}^T \Psi(\boldsymbol{x}, \boldsymbol{t}) = \sum_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}}) \in \mathcal{D}} \alpha_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}})} \Gamma((\boldsymbol{x}, \boldsymbol{t}), (\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}}))$$

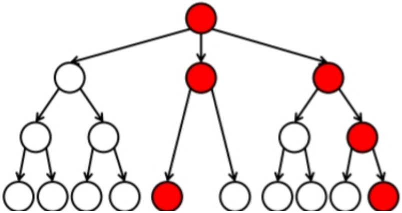
Model the joint kernel as a product of an instance kernel $k(\cdot, \cdot)$ and a target kernel $g(\cdot, \cdot)$:

$$\Gamma((\boldsymbol{x}, \boldsymbol{t}), (\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}})) = k(\boldsymbol{x}, \bar{\boldsymbol{x}}) \cdot g(\boldsymbol{t}, \bar{\boldsymbol{t}})$$

[11] Alvarez et al., Kernels for vector-valued functions: a review, Foundation and Trends in Machine Learning, 2012

Converting graphs to similarities or target representations

- **Similarities:** use graph structure to express target similarities
  e.g. the shortest-path kernel between two nodes
- **Representations:** often characteristics of a specific vertex or edge
  e.g. the number of positive labels that are siblings of a vertex[12]

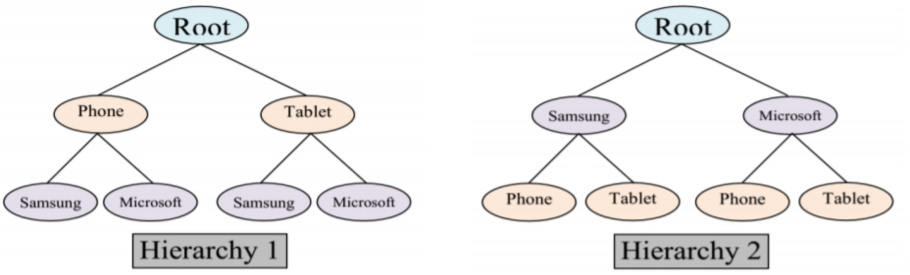


[12] Rousu et al., Kernel-based learning of hierarchical multilabel classification models, JMLR 2006

A unifying view on MTP methods

Group of methods	Applicable setting
Independent models	B and C
Similarity-enforcing methods	B and C
Relation-exploiting methods	B, C and D
Relation-constructing methods	B and C
Representation-exploiting methods	B, C and D
Representation-constructing methods	A, B and C

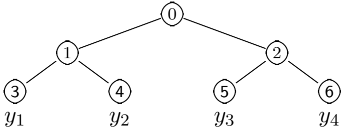
Constructing target hierarchies



- It might be difficult for a human expert to define a hierarchy[13]
- Perhaps one can try to learn the hierarchy from data?
- Algorithms: level flattening, node removal, hierarchy modification, hierarchy generation, etc.

[13] Rangwala and Naik, Tutorial on Large-Scale Hierarchical Classification, KDD 2017.

Label trees ($\neq$ decision trees)



- Organize classifiers in a tree structure (one leaf $\Leftrightarrow$ one label)
- Mainly used in multi-class and multi-label classification
- Goal is fast prediction: almost logarithmic in the number of labels
- Algorithms: Label embedding trees[14], Nested dichotomies[15], Conditional probability trees[16], Hierarchical softmax[17], FastText[18], Probabilistic classifier chains[19]

[14] Bengio et al., Label embedding trees for large multi-class tasks, NIPS 2010

[15] Frank and Kramer, Ensembles of nested dichotomies for multi-class problems, ICML 2004

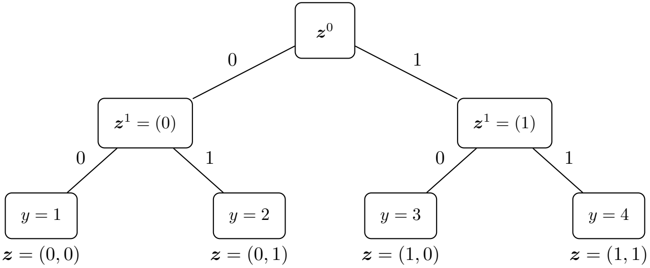
[16] Beygelzimer et al., Conditional probability tree estimation analysis and algorithms. UAI 2009

[17] Morin and Bengio, Hierarchical probabilistic neural network language model, AISTATS 2005

[18] Joulin et al., Bag of tricks for efficient text classifcation. CoRR, abs/1607.01759, 2016

[19] Dembczynski et al., Bayes optimal multilabel classification via probabilistic classifier chains, ICML 2010

Hierarchical softmax / Probabilistic classifier trees



- Encode the targets by a **prefix code** ($\Rightarrow$ tree structure)[20]
- Multi-class classification: each label y **coded** by $\boldsymbol{z} = (z_1, \ldots, z_l) \in \mathcal{C}$
- Multi-label classification: a label vector $\boldsymbol{y} = (y_1, \ldots, y_m)$ is a prefix code.

[20] Dembczynski et al., Consistency of probabilistic classifier trees. ECMLPKDD 2016

Probabilistic classifier chains

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

$$\ell_{0/1}(\boldsymbol{y}, \hat{y}) = [\![\boldsymbol{y} \neq \hat{y}]\!]$$

- Repeatedly apply the **product rule of probability**:

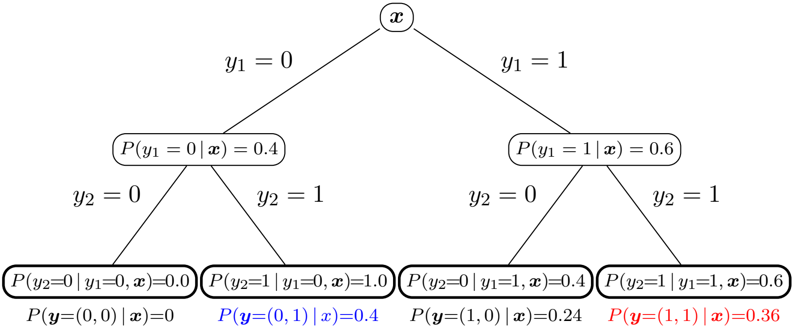
$$P(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{x}) = \prod_{i=1}^{m} P(Y_i = y_i \mid \boldsymbol{x}, y_1, \ldots, y_{i-1}) \, .$$

- Learning relies on constructing probabilistic classifiers for estimating

$$P(Y_i = y_i | \boldsymbol{x}, y_1, \ldots, y_{i-1}) \, ,$$

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

- Inference relies on exploiting a probability tree:



- For subset 0/1 loss one needs to find $\boldsymbol{h}(\boldsymbol{x}) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} P(\boldsymbol{y} \mid \boldsymbol{x})$.
- Greedy and approximate search techniques with guarantees exist.[21]
- Other losses: compute the prediction on a sample from $P(\boldsymbol{Y} \mid \boldsymbol{x})$.[22]

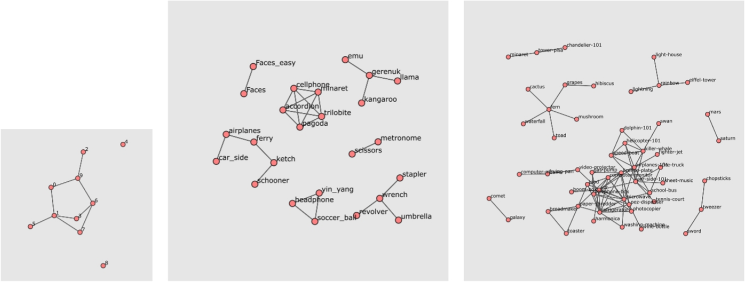
[21] Kumar et al., Beam search algorithms for multilabel learning, Machine Learning 2013

[22] 23 Dembczynski et al., An analysis of chaining in multi-label classification, ECAI 2012

Constructing target similarities by output kernel learning

- Consider models $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^{\mathbf{m}}$
- Training dataset $\{\mathbf{x_i}, \mathbf{y_i}\}_{\mathbf{i=1}}^{\mathbf{n}}$
- Learnable correlation matrix $\mathbf{\Gamma}$ between targets
- Learn output kernel and model parameters jointly[23]:

$$\min_{\mathbf{\Gamma} \in \mathbb{R}^{\mathbf{m} \times \mathbf{m}}} \left[\min_{\mathbf{f} \in \mathcal{F}} \sum_{i=1}^{n} \frac{||\mathbf{f}(\mathbf{x_i}) - \mathbf{y_i}||_{\mathbf{2}}^{\mathbf{2}}}{2\lambda} + \frac{||\mathbf{f}||_{\mathcal{F}}^{\mathbf{2}}}{2} + \frac{||\mathbf{\Gamma}||_{\mathbf{F}}^{\mathbf{2}}}{2} \right]$$

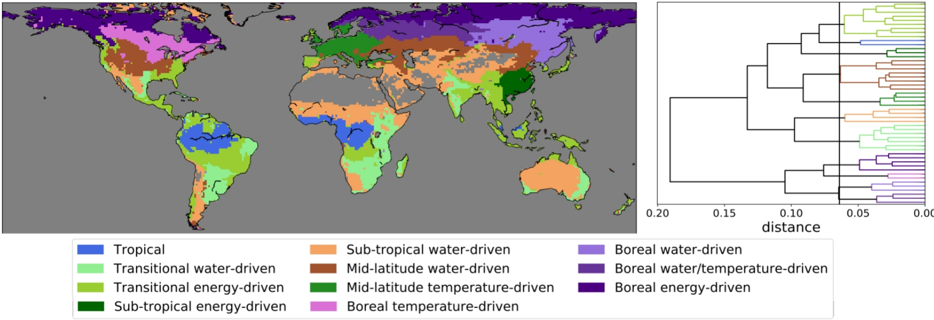


(a) USPS digits (b) Caltech 101 (c) Caltech 256

[23] Dinuzzo et al., Learning Output Kernels with Block Coordinate Descent, ICML 2011

Construcing hierarchies to obtain additional insight

- Application in climate science
- Result of learning 20000 tasks simultaneously with a multi-task learning method
- Followed by hierarchical clustering of the learned weight vectors[24]:

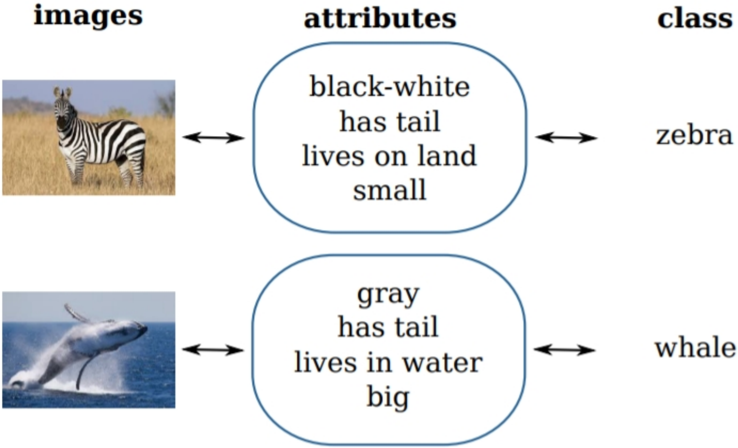


[24] Papagiannopoulou et al. Globral hydro-climatic biomes identified with multi-task learning, Geoscientific Model Development Discussions 2018

A unifying view on MTP methods

Group of methods	Applicable setting
Independent models	B and C
Similarity-enforcing methods	B and C
Relation-exploiting methods	B, C and D
Relation-constructing methods	B and C
Representation-exploiting methods	B, C and D
Representation-constructing methods	A, B and C

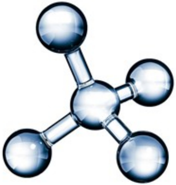
A target representation in computer vision

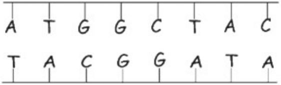


Target representations are the key element of zero-shot learning methods[25]

[25] Examples taken from the CVPR 2016 Tutorial on Zero-shot learning for Computer Vision

Target representations can take many forms

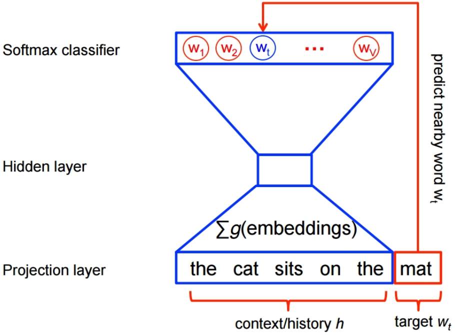


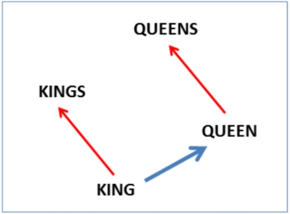


Learning target embeddings from text: Word2Vec

Predict the probability of the next word w_t given the previous words h[26]:

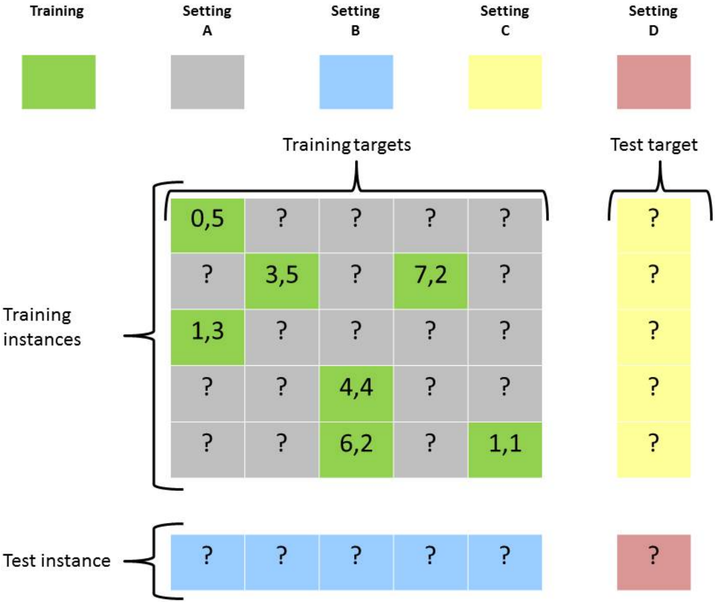
$$P(w_t \mid h) = \frac{\exp(f(w_t, h))}{\sum_{\text{allwords}} \exp(f(w_t, h))}$$





[26] Mikolov et al., Efficient Estimation of Word Representations in Vector Space, Arxiv 2013

Different learning settings revisited



Kronecker kernel ridge regression

Pairwise model representation in the primal:

$$f(\boldsymbol{x}, \boldsymbol{t}) = \boldsymbol{w}^T \left(\phi(\boldsymbol{x}) \otimes \psi(\boldsymbol{t})\right)$$

Kronecker product pairwise kernel in the dual[27]:

$$f(\boldsymbol{x}, \boldsymbol{t}) = \sum_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}}) \in \mathcal{D}} \alpha_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}})} k(\boldsymbol{x}, \bar{\boldsymbol{x}}) \cdot g(\boldsymbol{t}, \bar{\boldsymbol{t}}) = \sum_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}}) \in \mathcal{D}} \alpha_{(\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}})} \Gamma((\boldsymbol{x}, \boldsymbol{t}), (\bar{\boldsymbol{x}}, \bar{\boldsymbol{t}}))$$

Least-squares minimization with $\mathbf{z} = \mathrm{vec}(Y)$:

$$\min_{\boldsymbol{\alpha}} ||\boldsymbol{\Gamma}\boldsymbol{\alpha} - \mathbf{z}||_2^2 + \lambda \boldsymbol{\alpha}^\intercal \boldsymbol{\Gamma}\boldsymbol{\alpha}$$

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

Two-step zero-shot learning[28] [29]

	Mol1	Mol2	Mol3	Mol4	Mol5	Mol6	Mol7
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	?	?	?	?	?	?	?

[28] Pahikkala et al. A two-step approach for solving full and almost full cold-start problems in dyadic prediction, ECML/PKDD 2014.

[29] Romero-Paredes and Torr, An embarrassingly simple approach to zero-shot learning, ICML 2015.

Two-step zero-shot learning[30] [31]

	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	1,2	2,1	1,7	4,3	2,4	2,5

[30] Pahikkala et al. A two-step approach for solving full and almost full cold-start problems in dyadic prediction, ECML/PKDD 2014.

[31] Romero-Paredes and Torr, An embarrassingly simple approach to zero-shot learning, ICML 2015.

Two-step zero-shot learning[32] [33]

Mol1	Mol2	Mol3	Mol4	Mol5	Mol6	Mol7
1,3	0,2	1,4	1,7	3,5	1,3	1,2
2	1,7	1,5	7,5	8,2	7,6	1,4
0,2	0	0,3	0,4	1,2	2,2	3,8
3,1	1,1	1,3	1,1	1,7	5,2	1,1
4,7	2,1	2,5	1,5	2,3	8,5	1,5
1,2	2,1	1,7	4,3	2,4	2,5	4,3

[32] Pahikkala et al. A two-step approach for solving full and almost full cold-start problems in dyadic prediction, ECML/PKDD 2014.

[33] Romero-Paredes and Torr, An embarrassingly simple approach to zero-shot learning, ICML 2015.

Two-step kernel ridge regression

- Kernel evaluations for new test instance:

$$\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n))^\intercal$$
$$\mathbf{g}(\mathbf{t}) = (g(\mathbf{t}, \mathbf{t}_1), \ldots, g(\mathbf{t}, \mathbf{t}_q))^\intercal$$

Two-step kernel ridge regression

- Kernel evaluations for new test instance:

$$\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n))^\intercal$$

$$\mathbf{g}(\mathbf{t}) = (g(\mathbf{t}, \mathbf{t}_1), \ldots, g(\mathbf{t}, \mathbf{t}_q))^\intercal$$

- Step 1: prediction for $\mathbf{x}$ on all the training targets

$$\mathbf{f}_T(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\intercal A^{IT} = \mathbf{k}(\mathbf{x})^\intercal \left(\mathbf{K} + \lambda_d \mathbf{I}\right)^{-1} Y$$

Two-step kernel ridge regression

- Kernel evaluations for new test instance:

$$\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_n))^\intercal$$

$$\mathbf{g}(\mathbf{t}) = (g(\mathbf{t}, \mathbf{t}_1), \dots, g(\mathbf{t}, \mathbf{t}_q))^\intercal$$

- Step 1: prediction for $\mathbf{x}$ on all the training targets

$$\mathbf{f}_T(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\intercal A^{IT} = \mathbf{k}(\mathbf{x})^\intercal \left(\mathbf{K} + \lambda_d \mathbf{I}\right)^{-1} Y$$

- Step 2: generalizing to new targets

$$f^{\mathsf{TS}}(\mathbf{x}, \mathbf{t}) = \mathbf{g}(\mathbf{t})^\intercal \left(\mathbf{G} + \lambda_t \mathbf{I}\right)^{-1} \mathbf{f}_T(\mathbf{x})^\intercal$$

Two-step kernel ridge regression

- Kernel evaluations for new test instance:

$$\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n))^\intercal$$
$$\mathbf{g}(\mathbf{t}) = (g(\mathbf{t}, \mathbf{t}_1), \ldots, g(\mathbf{t}, \mathbf{t}_q))^\intercal$$

- Step 1: prediction for $\mathbf{x}$ on all the training targets

$$\mathbf{f}_T(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\intercal A^{IT} = \mathbf{k}(\mathbf{x})^\intercal \left(\mathbf{K} + \lambda_d \mathbf{I}\right)^{-1} Y$$

- Step 2: generalizing to new targets

$$\begin{aligned} f^{\mathsf{TS}}(\mathbf{x}, \mathbf{t}) &= \mathbf{g}(\mathbf{t})^\intercal \left(\mathbf{G} + \lambda_t \mathbf{I}\right)^{-1} \mathbf{f}_T(\mathbf{x})^\intercal \\ &= \mathbf{k}(\mathbf{x})^\intercal \left(\mathbf{K} + \lambda_d \mathbf{I}\right)^{-1} Y \left(\mathbf{G} + \lambda_t \mathbf{I}\right)^{-1} \mathbf{g}(\mathbf{t}) \end{aligned}$$

Two-step kernel ridge regression

- Kernel evaluations for new test instance:

$$\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n))^\intercal$$

$$\mathbf{g}(\mathbf{t}) = (g(\mathbf{t}, \mathbf{t}_1), \ldots, g(\mathbf{t}, \mathbf{t}_q))^\intercal$$

- Step 1: prediction for $\mathbf{x}$ on all the training targets

$$\mathbf{f}_T(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\intercal A^{IT} = \mathbf{k}(\mathbf{x})^\intercal \left(\mathbf{K} + \lambda_d \mathbf{I}\right)^{-1} Y$$

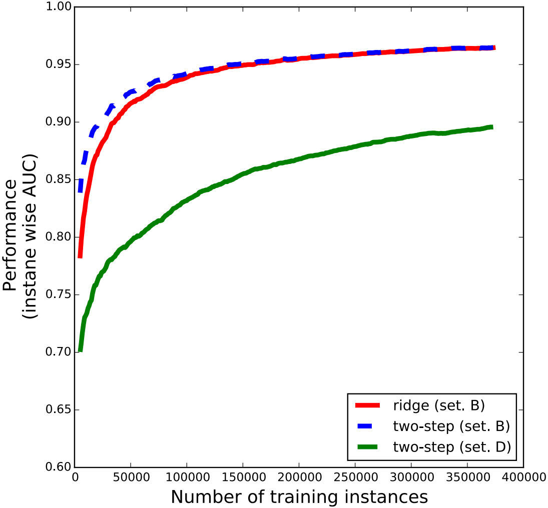
- Step 2: generalizing to new targets

$$\begin{aligned} f^{\mathsf{TS}}(\mathbf{x}, \mathbf{t}) &= \mathbf{g}(\mathbf{t})^\intercal \left(\mathbf{G} + \lambda_t \mathbf{I}\right)^{-1} \mathbf{f}_T(\mathbf{x})^\intercal \\ &= \mathbf{k}(\mathbf{x})^\intercal \left(\mathbf{K} + \lambda_d \mathbf{I}\right)^{-1} Y \left(\mathbf{G} + \lambda_t \mathbf{I}\right)^{-1} \mathbf{g}(\mathbf{t}) \\ &= \mathbf{k}(\mathbf{x})^\intercal A^{\mathsf{TS}} \mathbf{g}(\mathbf{t}) \\ &= \boldsymbol{w}^T \left(\phi(\boldsymbol{x}) \otimes \psi(\boldsymbol{t})\right) \end{aligned}$$

In which situations does zero-shot learning work?

[34] M. Stock, Exact and efficient algorithms for pairwise learning, PhD thesis, 2017

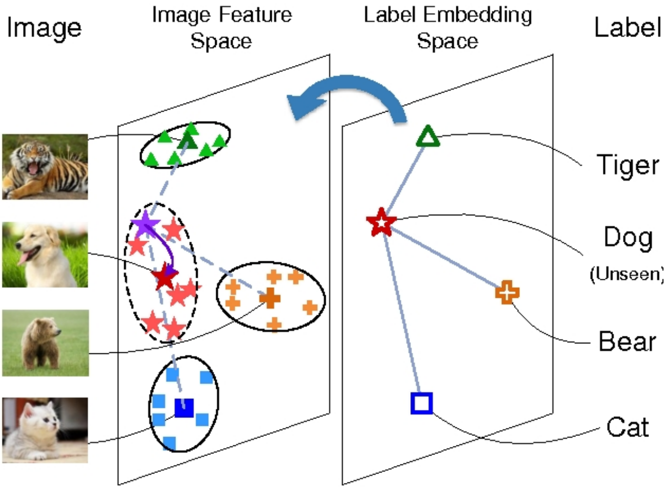
In which situations does zero-shot learning work?



$12,000$ labels: from $5,000$ to $350,000$ instances[34]

[34] M. Stock, Exact and efficient algorithms for pairwise learning, PhD thesis, 2017

Zero-shot learning in computer vision



Zero-shot learning in computer vision

Pairwise model representation as before:

$$f(\boldsymbol{x}, \boldsymbol{t}) = \boldsymbol{w}^T \left(\phi(\boldsymbol{x}) \otimes \psi(\boldsymbol{t})\right)$$

Inference in a structured prediction fashion:

$$\hat{c}(\boldsymbol{x}) = \arg\max_{\boldsymbol{t} \in \mathcal{T}} f(\boldsymbol{x}, \boldsymbol{t})$$

Different optimization problems:

- Multi-class objective[35]
- Ranking objective[36]
- Regression objective[37]
- Canonical correlation analysis

Different model formulations:

- Linear embeddings
- Nonlinear embeddings

---

[35] Akata et al., Evaluation of Output Embeddings for Fine-Grained Image Classification, CVPR2015

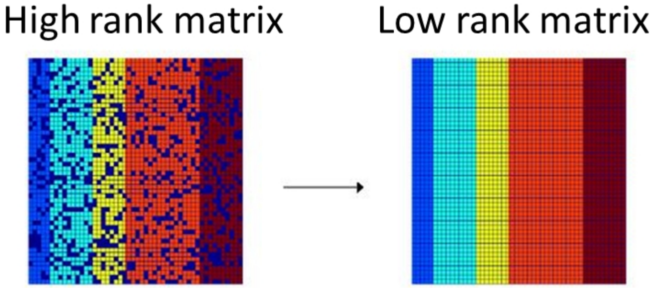
[36] Frome et al., Devise: A deep visual-semantic embedding model, NIPS 2013

[37] Socher et al., g. Zero-shot learning through cross-modal transfer, NIPS 2013

A unifying view on MTP methods

Group of methods	Applicable setting
Independent models	B and C
Similarity-enforcing methods	B and C
Relation-exploiting methods	B, C and D
Relation-constructing methods	B and C
Representation-exploiting methods	B, C and D
Representation-constructing methods	A, B and C

Low-rank approximation in Settings B and C



Typically perform a low-rank approximation of the parameter matrix[38]:

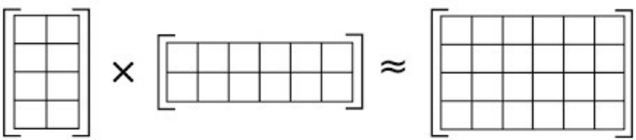
$$\min_{A} ||Y - XA||_F^2 + \lambda \, \mathrm{rank}(A)$$

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

Low-rank approximation in Settings B and C

- 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:

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



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

Low-rank approximation in Settings B and C

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[39], Canonical correlation analysis[40], Partial least squares
  - Singular value decomposition[41], Alternating structure optimization[42]
  - Compressed sensing[43], Output codes[44], Landmark labels[45], Bloom filters[46], Auto-encoders[47]

---

[39] Weston et al., Kernel dependency estimation, NIPS 2002

[40] Multi-label prediction via sparse infinite CCA, NIPS 2009

[41] Tai and Lin, Multilabel classification with principal label space transformation, Neural Computation 2012

[42] Zhou et al., Clustered Multi-Task Learning Via Alternating Structure Optimization, NIPS 2011

[43] Hsu et al., Multi-label prediction via compressed sensing. NIPS 2009

[44] Zhang and Schneider, Multi-label Output Codes using Canonical Correlation Analysis, UAI 2011

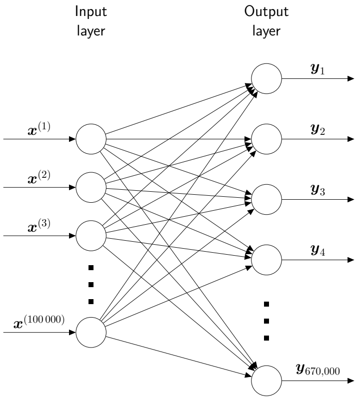
[45] Balasubramanian and Lebanon, The landmark selection method for multiple output prediction, ICML 2012

[46] Cissé et al., Robust bloom filters for large multilabel classification tasks, NIPS 2013

[47] Wicker et al., A nonlinear label compression and transformation method for multi-label classification using autoencoders, PAKDD 2016

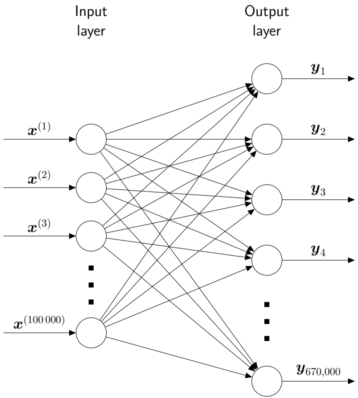
Target embeddings in neural networks

- Shallow Networks - SVM
  - Direct mapping of input to output

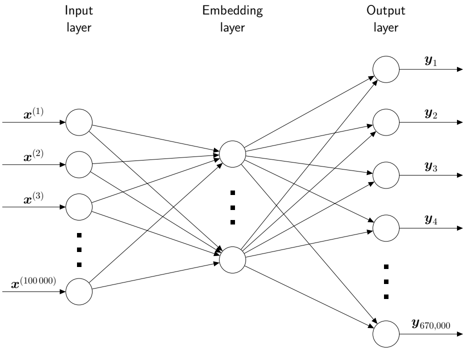


Target embeddings in neural networks

- Shallow Networks - SVM
  - Direct mapping of input to output

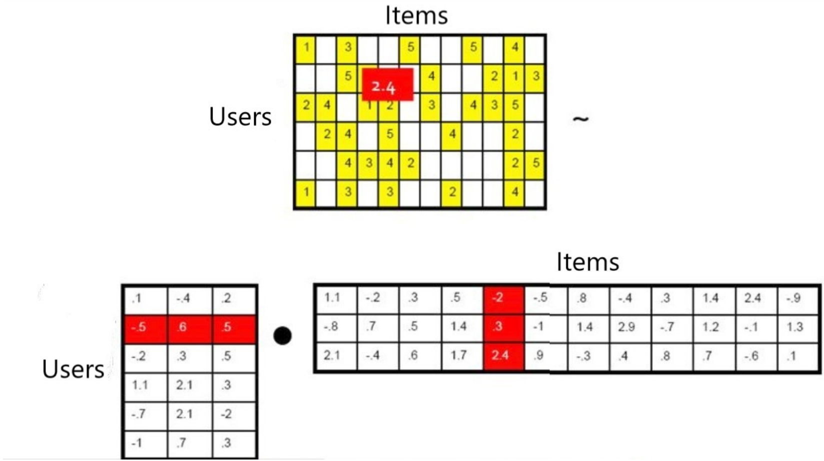


- Label embedding
  - Mapping input to output via embedding layer



Low-rank approximation in Setting A

Factorize the matrix Y instead of the parameter matrix A:



$$Y \quad = \quad U \quad \times \quad V$$

Low-rank approximation in Setting A

Overview of algorithms

- Nuclear norm minimizationan[48]
- Gaussian processes[49]
- Probabilistic methods[50]
- Spectral regularization[51]
- Non-negative matrix factorization[52]
- Alternating least-squares minimization[53]

[48] Candes and Recht, Exact low-rank matrix completion via convex optimization. Foundations of Computational Mathematics 2008

[49] Lawrence and Urtasun, Non-linear matrix factorization with Gaussian processes, ICML 2009

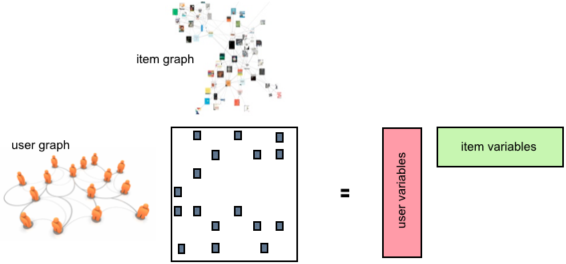
[50] Shan and Banerjee, Generalized probabilistic matrix factorizations for collaborative filtering, ICDM 2010

[51] Mazumder et al., Spectral regularization algorithms for learning large incomplete matrices., JMLR 2010

[52] Gaujoux and Seoighe, A flexible R package for nonnegative matrix factorization. BMC bioinformatics 2010

[53] Jain et al., Low-rank matrix completion using alternating minimization, ACM Symposium on Theory of Computing 2013

Matrix factorization with side information for Setting A



- Construct **implicit** features $(\boldsymbol{x}^I, \boldsymbol{t}^I)$ for users and items with matrix factorization methods
- Exploit **explicit** features $(\boldsymbol{x}^E, \boldsymbol{t}^E)$ (a.k.a. side information)
- Concatenate:

$$\boldsymbol{x}^C = (\boldsymbol{x}^I, \boldsymbol{x}^E), \qquad \boldsymbol{t}^C = (\boldsymbol{t}^I, \boldsymbol{t}^E)$$

- Apply methods that we have seen before[54][55]:

$$f(\boldsymbol{x}^C, \boldsymbol{t}^C) = \boldsymbol{w}^T \left(\phi(\boldsymbol{x}^C) \otimes \psi(\boldsymbol{t}^C)\right)$$

[54] Menon and Elkan, A log-linear model with latent features for dyadic prediction, ICDM 2010

[55] Volkovs and Zemel, Collaborative filtering with 17 parameters, NIPS 2012

Hybrid matrix factorization for Setting A

Basilico and Hofmann, 2004; Abernethy et al, 2008; Adams et al, 2010; Fang and Si, 2011; Zhou et al, 2011a; Menon and Elkan, 2011; Zhou et al, 2012b).

When is it useful to construct target representations?

Does not work well in extreme multi-label classification[56]:

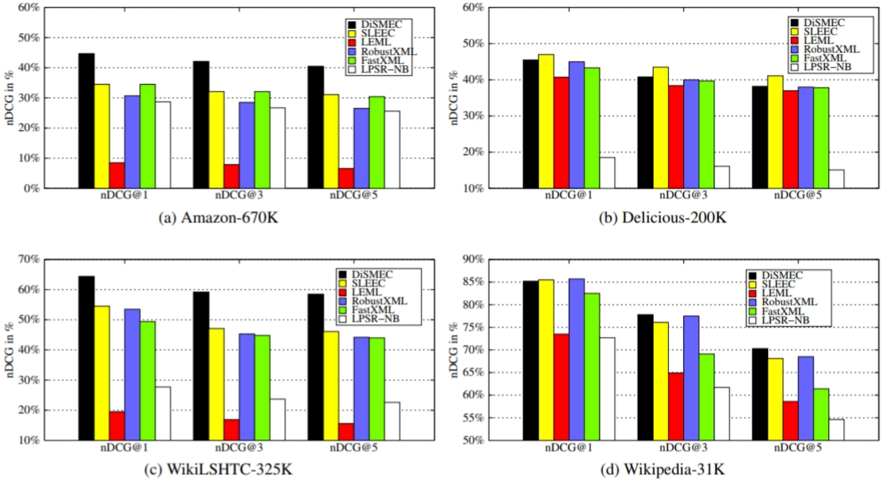
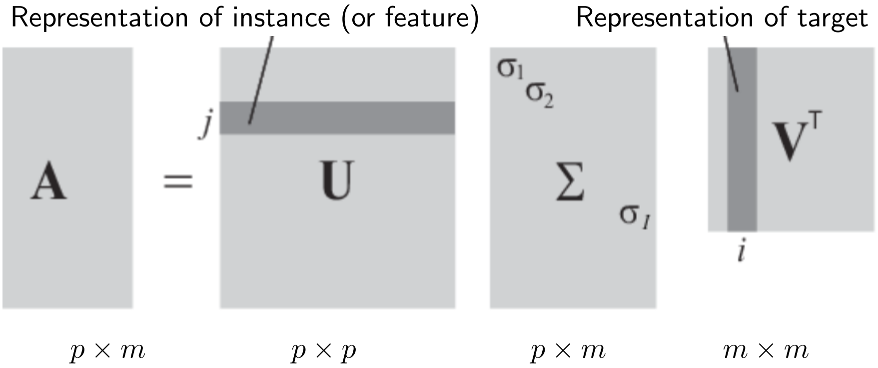


Figure 3: nDCG@k for k=1, 3 and 5

[56] Babbar and Schölkopf, DISMEC: Distributed Sparse Machines for Extreme Multi-label classification, WSDM 2017

When is it useful to construct target representations?

SVD interpretation



- $\sigma_1, \sigma_2, \ldots$: singular values of A
- Rank of A = number of non-zero **singular values**
- **High rank** when a lot of singular values differ from zero
- **Low rank** when a lot of singular values are zero
- Singular values **give insight** in what can be gained

Overview of this tutorial

1. A unifying view on MTP problems
2. A unifying view on MTP methods
3. Loss functions in multi-target prediction
4. Conclusions

Conclusions

- Multi-target prediction is an active field of research that connects different types of machine learning problems
- In the corresponding subfields of machine learning, problems have typically been solved in isolation, without establishing connections between methods
- Two-step zero-shot learning is a simple MTP method with a lot of interesting properties

Upcoming paper:
Waegeman et al.
Multi-Target Prediction:
A Unifying View on Problems and Methods

Multi-target prediction papers at ECML/PKDD 2018

- **Djerrab et al.**, Output Fisher embedding regression, Tuesday, 15h20
- **Pikalos et al.**, Global multi-output decision trees for interaction prediction, Thursday 11h20
- **Masera and Blanzieri**, AWX: An integrated approach to hierarchical multi-label classification, Tuesday 14h40
- **Decubber et al.**, Deep F-measure maximization in multi-label classification: a comparative study, Tuesday 14h20
- **Park and Read**, A blended metric for multi-label optimization and evaluation, Thursday 11h40
- **Rafailidis and Crestani**, Deep collaborative filtering with multifaceted contextual information in location-based social networks, Thursday 14h20
- **Du et al.**, POLAR: Attention-based CNN for one-shot personalized article recommendation, Thursday 14h40
- **Lan et al.**, Personalized thread recommendation of MOOC discussion forums, Thursday 14h20
- **Marecek et al.**, Matrix completion under interval uncertainty, Thursday 16h30