A two-step method to incorporate task features for large output spaces

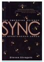
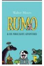
Michiel Stock[1], Tapio Pahikkala[2], Antti Airola[2], Bernard De Baets[1] & Willem Waegeman[1]

[1]KERMIT
Department of Mathematical Modelling, Statistics and Bioinformatics
Ghent University

[2]Department of Computer Science
University of Turku

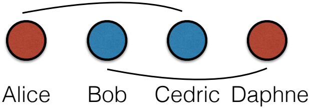
NIPS: extreme classification workshop
December 12, 2015

What will we read next?

	Alice	Bob	Cedric	Daphne
	5		4	
	1			4
		4		
	2		1	
	4		3	

What will we read next?

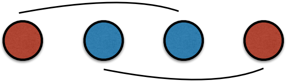
Social graph



Genre	Alice	Bob	Cedric	Daphne
1 1 0 1	5		4	
0 0 1 0	1			4
1 1 0 0		4		
0 0 1 0	2		1	
0 1 0 1	4		3	

What will we read next?

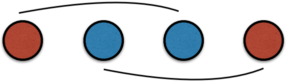
Social graph

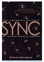
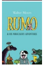


Genre	Alice	Bob	Cedric	Daphne
1 1 0 1	5	2.3	4	3.1
0 0 1 0	1	4.5	1.3	4
1 1 0 0	3.9	4	3.8	0.8
0 0 1 0	2	5.2	1	4.5
0 1 0 1	4	2.5	3	3.6

What will we read next?

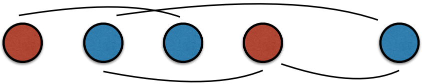
Social graph

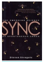
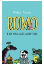


Genre		Alice	Bob	Cedric	Daphne
1 1 0 1		5	2.3	4	3.1
0 0 1 0		1	4.5	1.3	4
1 1 0 0		3.9	4	3.8	0.8
0 0 1 0		2	5.2	1	4.5
0 1 0 1		4	2.5	3	3.6
1 1 0 1		4.8	1.1	3.7	2.3

What will we read next?

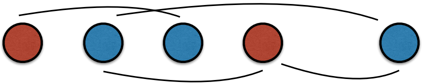
Social graph

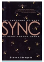
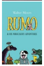


Genre		Alice	Bob	Cedric	Daphne	Eric
1 1 0 1		5	2.3	4	3.1	2.3
0 0 1 0		1	4.5	1.3	4	4.0
1 1 0 0		3.9	4	3.8	0.8	1.7
0 0 1 0		2	5.2	1	4.5	4.8
0 1 0 1		4	2.5	3	3.6	2.9
1 1 0 1		4.8	1.1	3.7	2.3	

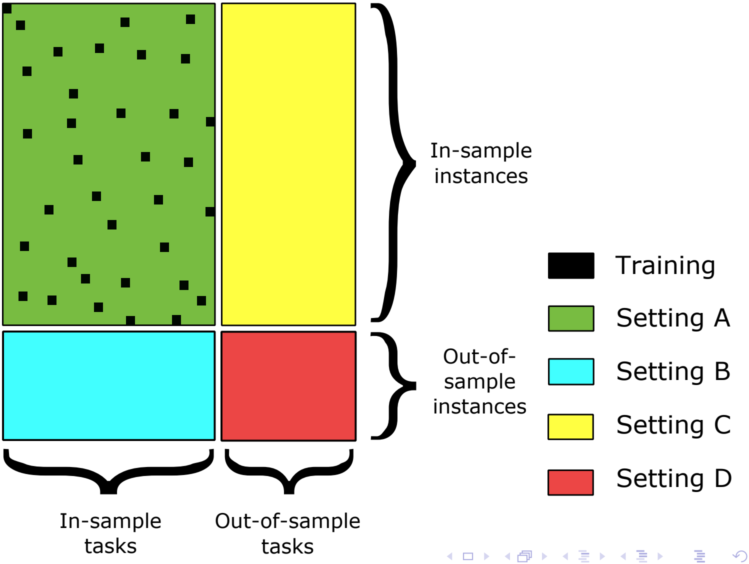
What will we read next?

Social graph

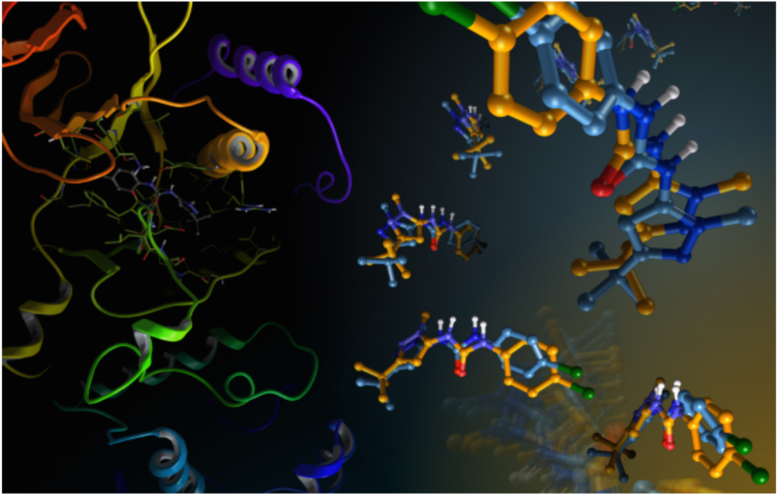


Genre		Alice	Bob	Cedric	Daphne	Eric
1 1 0 1		5	2.3	4	3.1	2.3
0 0 1 0		1	4.5	1.3	4	4.0
1 1 0 0		3.9	4	3.8	0.8	1.7
0 0 1 0		2	5.2	1	4.5	4.8
0 1 0 1		4	2.5	3	3.6	2.9
1 1 0 1		4.8	1.1	3.7	2.3	2.4

Learning relations

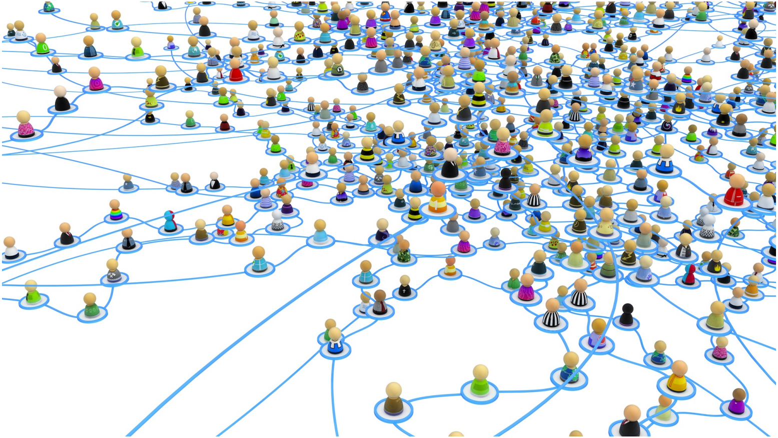


Other cool applications: drug design



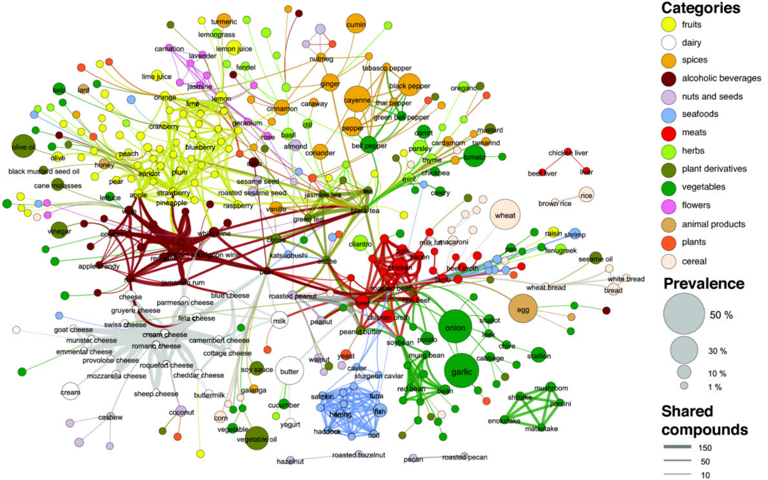
Predicting interaction between proteins and small compounds

Other cool applications: social network analysis



Predicting links between people

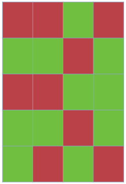
Other cool applications: food pairing



Finding ingredients that pair well

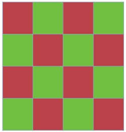
Learning with pairwise feature representations

Features books



Φ

Features readers

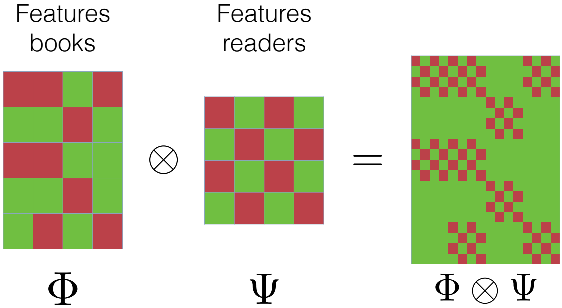


Ψ

- d : instance (e.g. book)
- $\phi(d)$: instance features (e.g. genre)

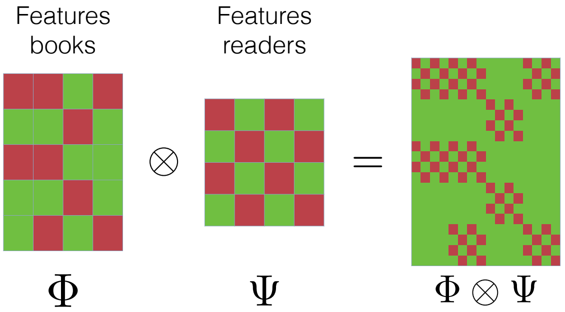
- t : task (e.g. reader)
- $\psi(t)$: task features (e.g. social network)

Learning with pairwise feature representations



- d : instance (e.g. book)
- $\phi(d)$: instance features (e.g. genre)
- t : task (e.g. reader)
- $\psi(t)$: task features (e.g. social network)

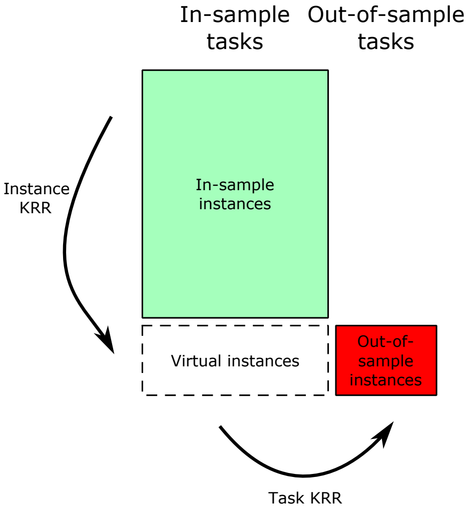
Learning with pairwise feature representations



- d : instance (e.g. book)
- $\phi(d)$: instance features (e.g. genre)
- t : task (e.g. reader)
- $\psi(t)$: task features (e.g. social network)

Pairwise prediction function: $f(d,t) = \mathbf{w}^\intercal(\phi(d) \otimes \psi(t))$

Learning relations in two steps



1. Build a ridge regression model to generalize to new instances
2. Build a ridge regression model to generalize to new tasks

The two-step ridge regression

Prediction function:

$$f(d,t)=\phi(d)^\intercal \mathbf{W}\psi(t)$$

Parameters can be found by solving:

$$\mathbf{\Phi}^\intercal\mathbf{Y}\mathbf{\Psi}=(\mathbf{\Phi}^\intercal\mathbf{\Phi}+\lambda_d\mathbf{I})\mathbf{W}(\mathbf{\Psi}^\intercal\mathbf{\Psi}+\lambda_t\mathbf{I})$$

The two-step ridge regression

Prediction function:

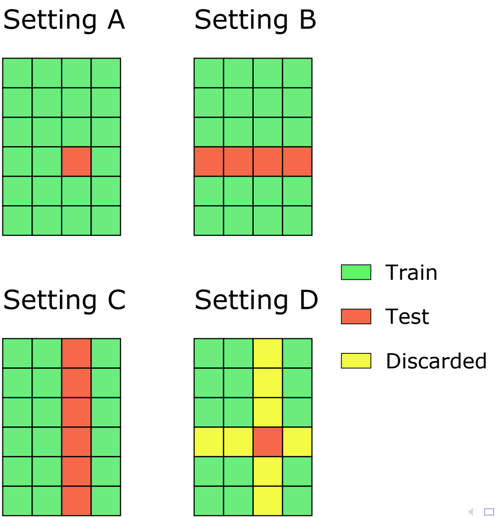
$$f(d,t)=\phi(d)^\intercal\mathbf{W}\psi(t)$$

Parameters can be found by solving:

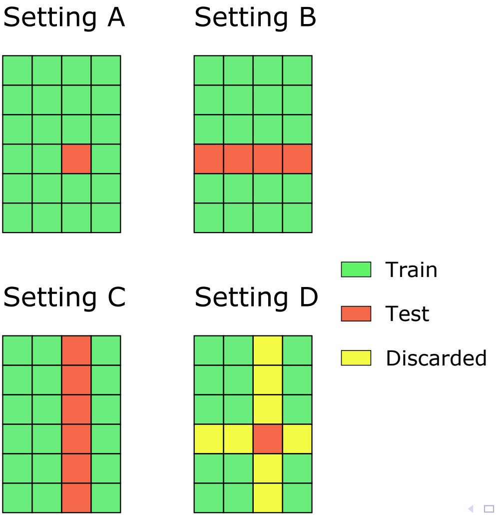
$$\mathbf{\Phi}^\intercal\mathbf{Y}\mathbf{\Psi}=(\mathbf{\Phi}^\intercal\mathbf{\Phi}+\lambda_d\mathbf{I})\mathbf{W}(\mathbf{\Psi}^\intercal\mathbf{\Psi}+\lambda_t\mathbf{I})$$

Two hyperparameters: λ_d and λ_t!

Four ways of cross validation



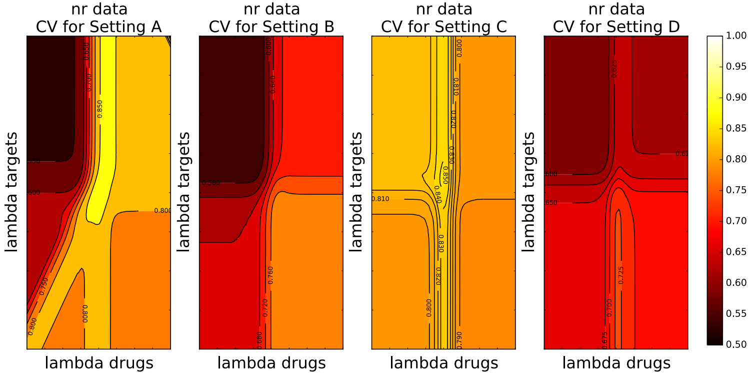
Four ways of cross validation



- Analytic shortcuts can be derived to perform LOOCV for each setting!
- Tuning λ_d and λ_t essentially free!

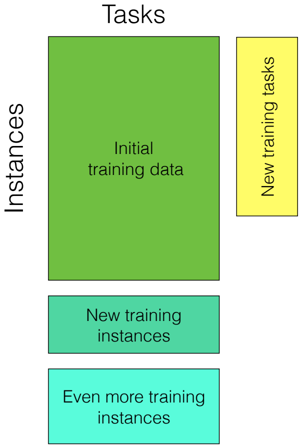
Effect of regularization for the four settings

Data: protein-ligand interactions.
Evaluation by AUC (lighter = better performance)

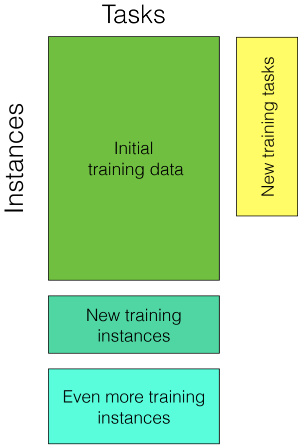


Clear difference between four settings and λ_d and λ_t!

Learning with mini-batches



Learning with mini-batches

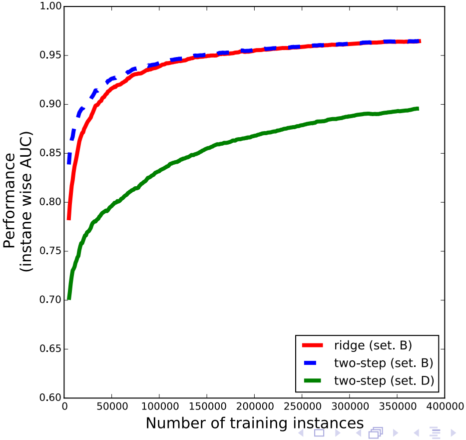


Exact updating of the parameters when new training instances and/or taks become available

- scalable for "Big Data" applications
- updating model in dynamic environment

Exact online learning for hierarchical text classification

Hierarchical text classification ($> 12,000$ labels): from 5,000 to 350,000 instances in steps of 1,000 instances.



Why two-step ridge regression?

- Zero-shot learning, transfer learning, multi-task learning... in one line of code

Why two-step ridge regression?

- Zero-shot learning, transfer learning, multi-task learning... in one line of code
- Theoretically well founded

Why two-step ridge regression?

- Zero-shot learning, transfer learning, multi-task learning... in one line of code
- Theoretically well founded
- Allows for nifty computational tricks
    - 'free' tuning for the hyperparameters
    - 'free' LOOCV for all four settings!
    - closed-form solution for updating with mini-batches

A two-step method to incorporate task features for large output spaces

Michiel Stock[1], Tapio Pahikkala[2], Antti Airola[2], Bernard De Baets[1] & Willem Waegeman[1]

[1]KERMIT
Department of Mathematical Modelling, Statistics and Bioinformatics
Ghent University

[2]Department of Computer Science
University of Turku

NIPS: extreme classification workshop
December 12, 2015