Meta-Learning as a Collaborative Filtering Problem

Mustafa Misir, Michèle Sebag

MetaSel@ECAI 2014, August 2014

Control layer in algorithmic platforms

Goal

deliver peak performance on any/most problem instance

A general issue

- In constraint programming — Rice 76
- In stochastic optimization — Grefenstette 87
- In machine learning (meta-learning) — Bradzil 93

Scopes

- Offline control
  Portfolio algorithm selection, optimal hyper-parameter setting
- Online control
  adjusting hyper-parameters during the run

Control layer in algorithmic platforms

Goal

deliver peak performance on any/most problem instance

A general issue

- In constraint programming — Rice 76
- In stochastic optimization — Grefenstette 87
- In machine learning (meta-learning) — Bradzil 93

Scopes

- Offline control
  Portfolio algorithm selection, optimal hyper-parameter setting
- Online control
  adjusting hyper-parameters during the run

Control: A meta-learning problem

Procedure

- Gather problem instances (benchmark suite)
- Design descriptive features for pb instances
- Run algorithms on pb instances
- Build meta-training set:

$$\mathcal{E} = \{(\text{desc. of } i\text{-th pb instance, perf. of } j\text{-th algo})\}$$

- Learn $\hat{h}$ from $\mathcal{E}$
- Decision making (predict, optimize)

Bottleneck: design good cheap descriptive features

Some advances in CP and SAT

- CPHydra — O'Mahony et al. 08
  case-based reasoning; kNN
- Satzilla — Xu et al. 08
  learn $\widehat{\text{runtime}}$(inst,alg); select argmin $\widehat{\text{runtime}}$
- ParamILS — Hutter et al. 09
  learn $\widehat{\text{perf}}$(hyper-param); optimize $\widehat{\text{perf}}$
- Programming by optimization — Holger Hoos, 12
  http://www.prog-by-opt.net/

100 Features — Hutter et al. 06, 07

Static features

Problem definition: density, tightness

Variable size and degree (min, max, average, variance

Constraint degree and cost category (exp, cubic, quadratic, lin. cheap, lin. expensive)

Dynamic features

Heuristic criteria(variable): wdeg, domdeg, impact: min, max, average

Constraint weight (wdeg): min, max, average

Constraint filtering: min, max, average of number of times called by propagation

Some advances in ML

- Matchbox — Stern et al. 10
  Collaborative filtering + Bayesian learning
- SCOT — Bardenet et al. 13
  $\widehat{\text{perf}}$(hyper-param); optimize $\widehat{\text{perf}}$
  where $\widehat{\text{perf}}$ is learned using learning-to-rank.

Overview

Motivations

Algorithm Recommender System

Empirical evaluation

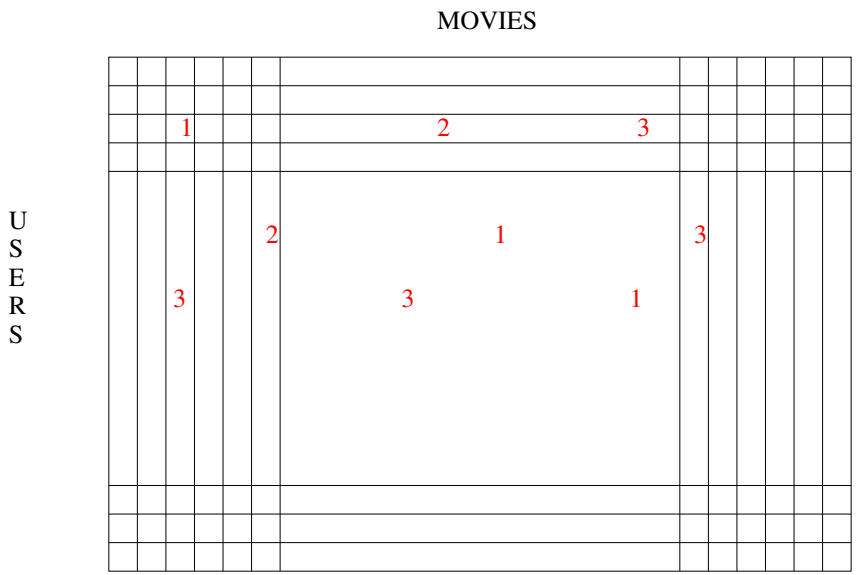
Main idea

Stern et al. 10

Netflix challenge

Recommender systems

- Set of users, set of products
- Users like/dislike a few products
- A sparse matrix



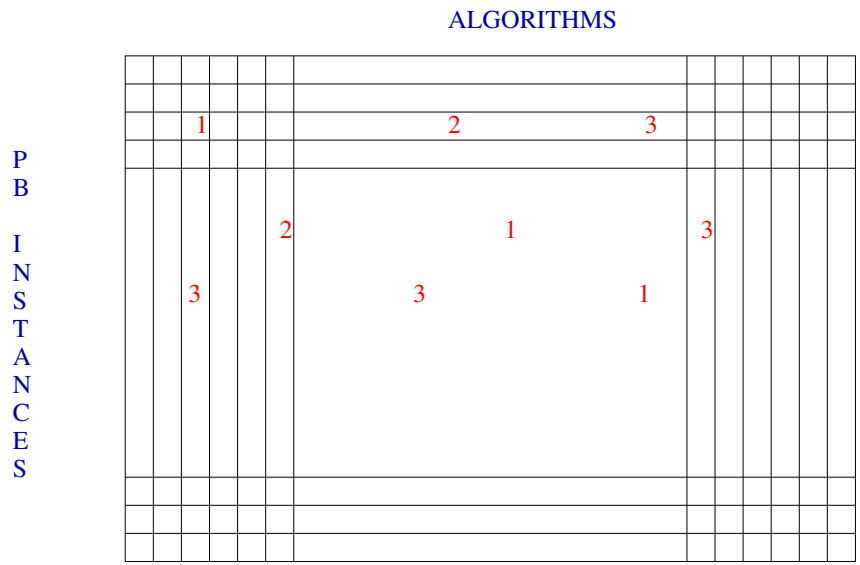
Main idea

Stern et al. 10

Recommender systems

Netflix challenge

- Set of users, set of products
- Users like/dislike a few products
- A sparse matrix



Algorithm selection

- Set of problem instances, set of algorithms
- Pb instance likes "better"

  algorithms that behave "better" on instance

Differences

- Meta-Learning is not (yet) a Big Data problem
  (500.000 users, 180.000 movies in Netflix)
- The main issue is: dealing with a brand new problem instance:
  cold start

Milestones

Acquire data — Sparse matrix

- Run a few alg. on problem instances

Collaborative filtering — Fill the matrix

- Content-based
- Model-based

Cold start

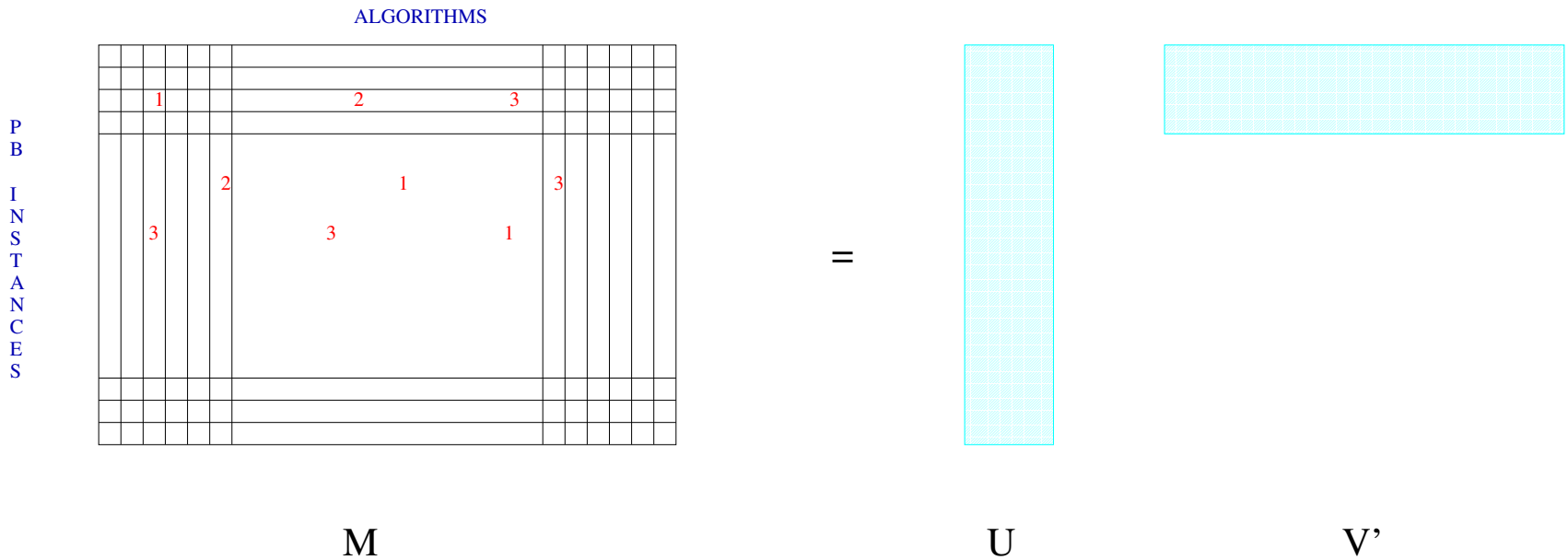
- Handle a brand new pb instance

Collaborative filtering

Matrix decomposition

- U: $P \times k$, k = nb latent factors
- V: $A \times k$,
- s.t. $\mathcal{M} \approx UV'$

$$m_{i,j} \approx \langle u_i, v_j \rangle$$



Loss

- RMSE root mean square error
- MAE mean absolute error
- Rank loss Cofirank, Weimer et al. 07

CofiRank

Criterion NDCG

$$DCG(\pi, k) = \sum_{i=1}^{k} \frac{2^{\pi(i)} - 1}{log(i+2)}$$

$$NDCG(\pi, k) = \frac{DCG(\pi, k)}{DCG(\pi^*, k)}$$

Non convex !

- Use a linear convex upper bound
- Alternate minimization (opt. U with fixed V; then opt. V with fixed U)

Algorithms

Matchbox: Bayesian learning — Sterner et al. 10

- Define priors on U and V
- Finite number of perf. levels (1, 2, 3)
- Learn thresholds from $\langle u, v \rangle$ to perf. level
- Latent features = linear combinations of initial features

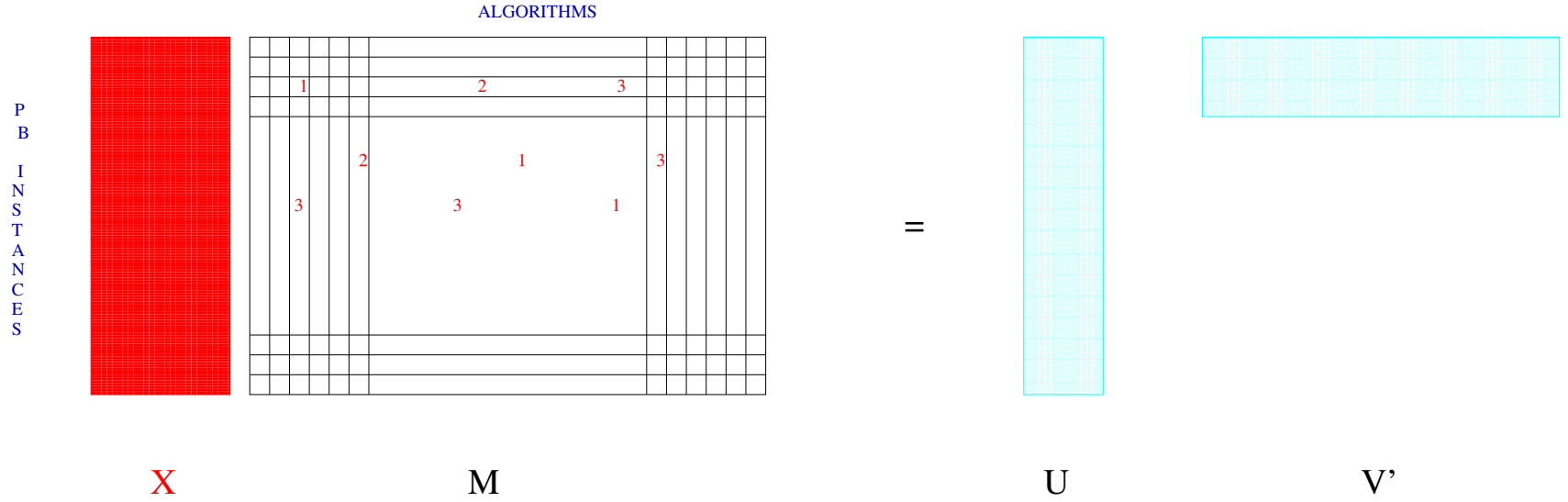
ARS: Algorithm Recommender System — Misir & S. 13

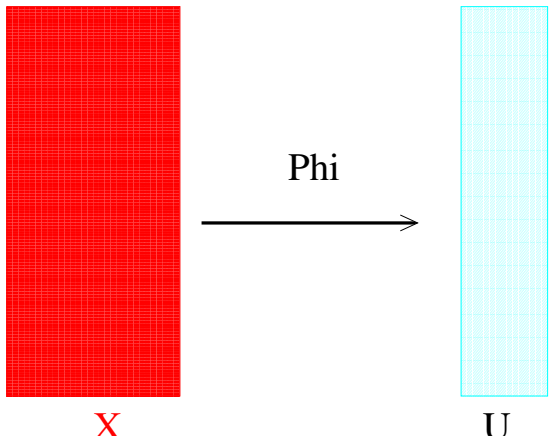
- Content-based: cosine, SVD
- Model-based: RMSE, CofiRank

Cold start in ARS

The touchstone for meta-learning

- Given initial descriptive features X
- Use matrix decomposition to build latent features U
- Learn $U \approx \phi(X)$





Overview

Motivations

Algorithm Recommender System

Empirical evaluation

Experimental setting

Goals of experiments

- Performance wrt matrix sparsity
- Performance of cold-start
- Inspecting latent features

Domains

- Satisfiability benchmark — SAT 2011
- Constraint programming challenge — CP 2008
- Black-box optimization benchmark — BBOB 2012
- Machine learning (from Irvine rep.) — Gama et al. 2000

Experimental setting

- Sparsity in 10% - 90% (at least 1 rank on each line)
- Cold start: 10-fold CV

SAT 2011

The problems

	Dataset	# Solvers	# Instances	# Solved Instances	Best Single Solver
Phase 1	APP	67	300	228 (76%)	(174) Glucose 2.0 [23]
	CRF	52	300	199 (66%)	(138) ppfolio-seq [24]
	RND	47	600	462 (77%)	(399) 3S [25]
Phase 2	APP	26	300	253 (84%)	(215) Glucose 2.0
	CRF	24	300	229 (76%)	(163) 3S
	RND	14	600	492 (82%)	(408) 3S

Descriptive features

Problem Size Features:

1–2. **Number of variables and clauses in original formula**: denoted v and c, respectively
3–4. **Number of variables and clauses after simplification with** `SATElite`: denoted v' and c', respectively
5–6. **Reduction of variables and clauses by simplification**: (v-v')/v' and (c-c')/c'
7. **Ratio of variables to clauses**: v'/c'

Variable-Clause Graph Features:

8–12. **Variable node degree statistics:** mean, variation coefficient, min, max, and entropy
13–17. **Clause node degree statistics:** mean, variation coefficient, min, max, and entropy

Variable Graph Features:

18–21. **Node degree statistics:** mean, variation coefficient, min, and max
22–26. **Diameter:** mean, variation coefficient, min, max, and entropy

Clause Graph Features:

27–31. **Node degree statistics:** mean, variation coefficient, min, max, and entropy
32–36. **Clustering Coefficient:** mean, variation coefficient, min, max, and entropy

Balance Features:

37–41. **Ratio of positive to negative literals in each clause:** mean, variation coefficient, min, max, and entropy
42–46. **Ratio of positive to negative occurrences of each variable:** mean, variation coefficient, min, max, and entropy
47–49. **Fraction of unary, binary, and ternary clauses**

Proximity to Horn Formula:

50. **Fraction of Horn clauses**
51–55. **Number of occurrences in a Horn clause for each variable:** mean, variation coefficient, min, max, and entropy

DPLL Probing Features:

56–60. **Number of unit propagations:** computed at depths 1, 4, 16, 64 and 256
61–62. **Search space size estimate:** mean depth to contradiction, estimate of the log of number of nodes

LP-Based Features:

63–66. **Integer slack vector:** mean, variation coefficient, min, and max
67. **Ratio of integer vars in LP solution**
68. **Objective value of LP solution**

Local Search Probing Features, based on 2 seconds of running each of SAPS and GSAT

69–78. **Number of steps to the best local minimum in a run:** mean, median, variation coefficient, 10th and 90th percentiles
79–82. **Average improvement to best in a run** mean and coefficient of variation of improvement per step to best solution
83–86. **Fraction of improvement due to first local minimum:** mean and variation coefficient
87–90. **Coefficient of variation of the number of unsatisfied clauses in each local minimum:** mean and variation coefficient

Clause Learning Features (based on 2 seconds of running Zchaff_rand):

91–99. **Number of learned clauses:** mean, variation coefficient, min, max, 10%, 25%, 50%, 75%, and 90% quantiles
100–108. **Length of learned clauses:** mean, variation coefficient, min, max, 10%, 25%, 50%, 75%, and 90% quantiles

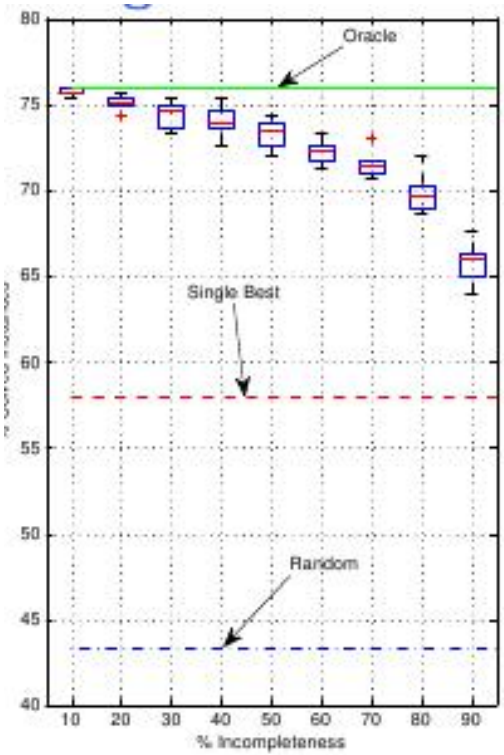
Survey Propagation Features

109–117. **Confidence of survey propagation:** For each variable, compute the higher of $P(true)/P(false)$ or $P(false)/P(true)$. Then compute statistics across variables: mean, variation coefficient, min, max, 10%, 25%, 50%, 75%, and 90% quantiles
118–126. **Unconstrained variables:** For each variable, compute $P(unconstrained)$. Then compute statistics across variables: mean, variation coefficient, min, max, 10%, 25%, 50%, 75%, and 90% quantiles

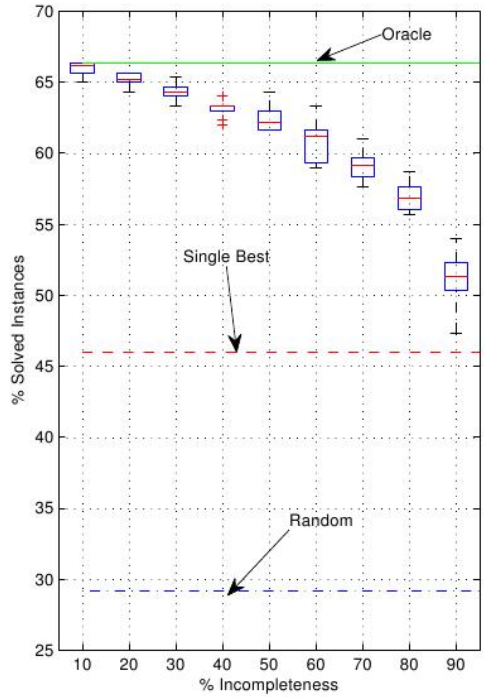
Timing Features

127–138. **CPU time required for feature computation:** one feature for each of 12 computational subtasks

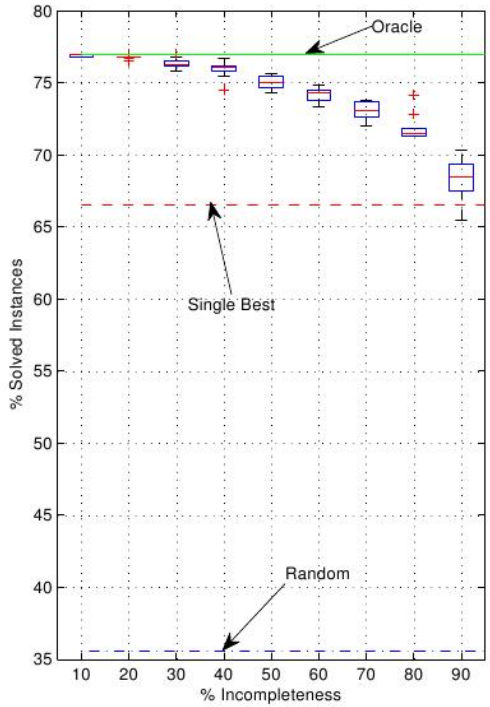
SAT 2011, filling the matrix



APP



CRF



RND

Improves on Single Best for sparsity up to 90%.

(number of problems solved: the higher the better)

SAT 2011, cold start

Number of solved instances 10-CV

	Method	Phase 1			Phase 2		
		APP	CRF	RND	APP	CRF	RND
	Oracle	22.8 ± 2.5	19.9 ± 2.1	46.2 ± 3.7	25.3 ± 2.1	22.9 ± 2.5	49.2 ± 2.4
	Random	13.0 ± 2.2	8.8 ± 1.5	21.4 ± 1.8	19.3 ± 2.9	12.0 ± 1.2	33.5 ± 2.3
	SingleBest	17.4 ± 3.0	13.8 ± 1.9	39.9 ± 3.2	$\mathbf{21.5 \pm 3.6}$	16.3 ± 2.5	40.8 ± 2.4
SVM	CofiRank	17.3 ± 3.2	13.9 ± 3.3	42.8 ± 3.8	21.1 ± 3.1	16.7 ± 2.7	44.0 ± 4.5
SVM	10NN-Sol	$\mathbf{17.5 \pm 2.6}$	14.0 ± 3.0	42.5 ± 4.1	21.2 ± 3.1	17.5 ± 1.8	$\mathbf{46.7 \pm 3.1}$
SVM	FullNN-Sol	$\mathbf{17.5 \pm 2.6}$	14.1 ± 3.1	42.4 ± 4.1	21.3 ± 3.1	17.8 ± 1.6	$\mathbf{46.8 \pm 3.1}$
SVM	10NN-Inst	$\mathbf{17.9 \pm 2.5}$	14.5 ± 3.3	43.1 ± 4.2	$\mathbf{21.6 \pm 3.1}$	17.9 ± 1.7	46.0 ± 3.2
SVM	FullNN-Inst	$\mathbf{17.9 \pm 2.6}$	14.6 ± 3.3	43.2 ± 4.1	21.4 ± 3.1	18.0 ± 1.6	46.0 ± 3.6
N. Network	CofiRank	15.8 ± 3.5	13.2 ± 3.0	41.0 ± 5.0	20.4 ± 3.2	15.9 ± 2.9	43.9 ± 3.9
N. Network	10NN-Sol	$\mathbf{17.6 \pm 2.6}$	13.8 ± 2.8	42.4 ± 3.7	21.3 ± 3.2	17.2 ± 1.8	45.7 ± 3.3
N. Network	FullNN-Sol	17.3 ± 2.6	$\mathbf{15.3 \pm 2.7}$	42.2 ± 3.7	21.3 ± 3.2	$\mathbf{19.0 \pm 2.3}$	46.2 ± 3.2
N. Network	10NN-Inst	$\mathbf{17.9 \pm 2.9}$	$\mathbf{15.6 \pm 2.9}$	$\mathbf{43.6 \pm 3.3}$	$\mathbf{21.8 \pm 3.0}$	$\mathbf{19.3 \pm 2.1}$	45.9 ± 2.9
N. Network	FullNN-Inst	$\mathbf{17.7 \pm 2.6}$	14.0 ± 2.8	$\mathbf{43.5 \pm 3.6}$	$\mathbf{21.8 \pm 3.0}$	17.4 ± 1.8	45.7 ± 3.2

CP 2008

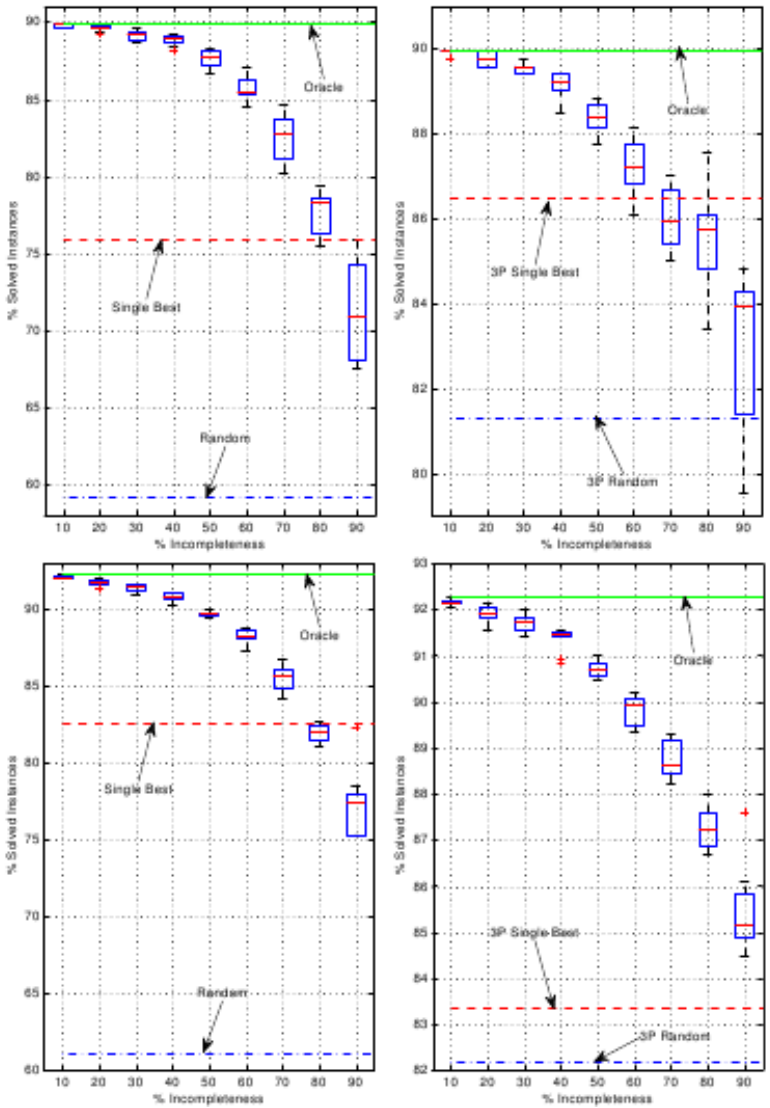
The problems

36 features

Dataset	# Solvers	# Instances	# Solved Instances	Best Single Solver
GLOBAL	17	548	493 (90%)	(416) Sugar-v1.13+picosat [TTB08]
k-ARY-INT ($k \geq 2$)	22	1411	1302 (92%)	(1165) cpHydra-k_40 [OHH+08]
2-ARY-EXT	23	633	620 (98%)	(572) cpHydra-k_10
N-ARY-EXT ($N > 2$)	24	546	449 (82%)	(431) cpHydra-k_40

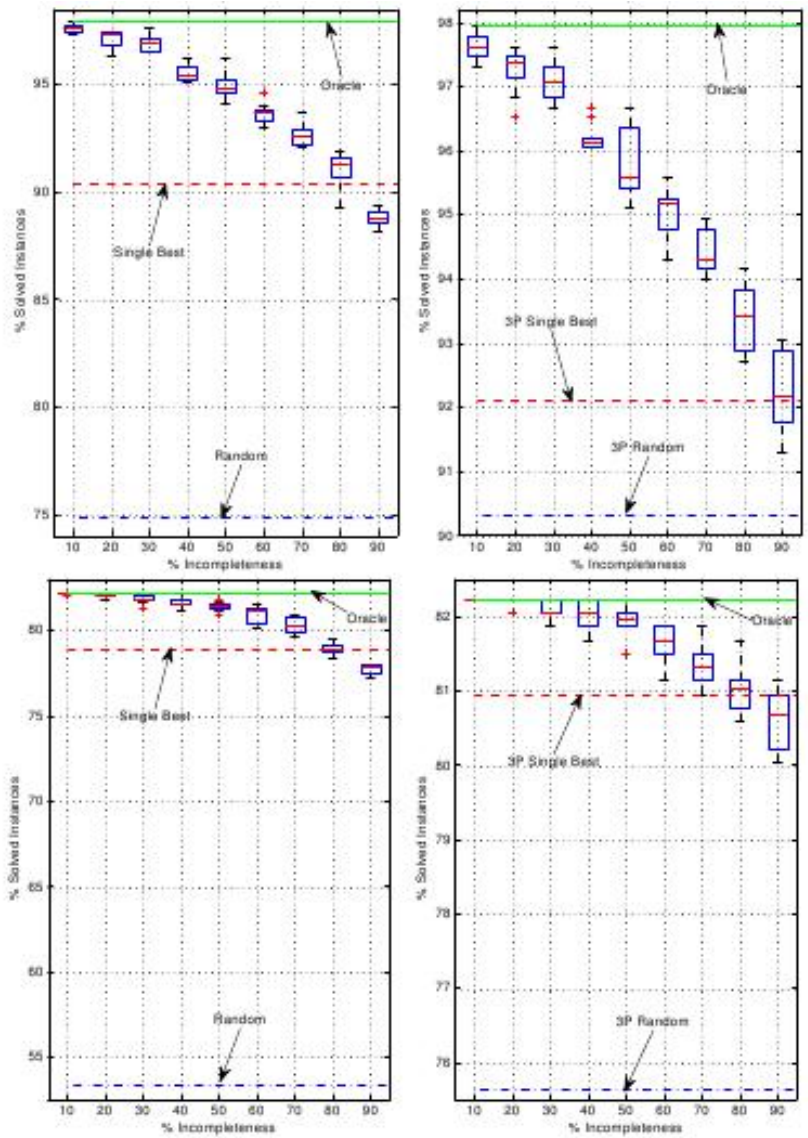
CP 2008, filling the matrix 1/2

CSP Global and CSP-k-AryInt



CP 2008 filling the matrix 2/2

CSP-2-AryExt and CSP-k-AryExt



CP 2008, cold start

Number of solved instances 10-CV

	Method	GLOBAL	k-ARY-INT	2-ARY-EXT	N-ARY-EXT
	Oracle	49.3 ± 3.2	130.2 ± 3.2	62.0 ± 1.3	44.9 ± 2.3
	Random	32.4 ± 3.2	86.2 ± 4.9	47.4 ± 3.2	29.2 ± 2.4
	SingleBest	41.6 ± 5.2	$\mathbf{116.5 \pm 5.8}$	$\mathbf{57.2 \pm 2.3}$	43.1 ± 2.8
	3P-Random	44.6 ± 3.2	115.9 ± 4.4	57.2 ± 2.3	41.3 ± 2.0
	3P-SingleBest	$\mathbf{47.4 \pm 3.8}$	117.6 ± 5.6	$\mathbf{58.3 \pm 2.5}$	$\mathbf{44.2 \pm 2.3}$
SVM	CofiRank	39.5 ± 5.1	111.5 ± 7.4	56.2 ± 2.9	42.2 ± 2.0
SVM	FullNN-Inst	43.6 ± 4.8	$\mathbf{115.3 \pm 6.9}$	$\mathbf{57.1 \pm 2.9}$	$\mathbf{43.4 \pm 2.4}$
SVM	3P-CofiRank	44.0 ± 4.0	119.6 ± 5.8	57.4 ± 2.4	43.8 ± 2.2
SVM	3P-FullNN-Inst	$\mathbf{47.0 \pm 3.6}$	$\mathbf{122.2 \pm 5.8}$	$\mathbf{58.3 \pm 2.2}$	$\mathbf{44.2 \pm 2.2}$
NNet	CofiRank	39.4 ± 5.1	110.8 ± 6.9	56.1 ± 2.9	42.1 ± 2.1
NNet	FullNN-Inst	$\mathbf{44.1 \pm 4.4}$	115.0 ± 6.4	$\mathbf{57.4 \pm 2.9}$	$\mathbf{43.4 \pm 2.6}$
NNet	3P-CofiRank	43.9 ± 4.0	119.1 ± 5.7	57.3 ± 2.5	43.8 ± 2.2
NNet	3P-FullNN-Inst	$\mathbf{46.9 \pm 3.5}$	$\mathbf{121.9 \pm 5.4}$	$\mathbf{58.6 \pm 2.3}$	$\mathbf{44.2 \pm 2.2}$

BBOB 2012

The problems

Function	multim.	gl.-struc.	separ.	scaling	homog.	basins	gl.-loc.	plat.
1: Sphere	none	none	high	none	high	none	none	none
2: Ellipsoidal separable	none	none	high	high	high	none	none	none
3: Rastrigin separable	high	strong	none	low	high	low	low	none
4: Büche-Rastrigin	high	strong	high	low	high	med.	low	none
5: Linear Slope	none	none	high	none	high	none	none	none
6: Attractive Sector	none	none	high	low	med.	none	none	none
7: Step Ellipsoidal	none	none	high	low	high	none	none	small
8: Rosenbrock	low	none	none	none	med.	low	low	none
9: Rosenbrock rotated	low	none	none	none	med.	low	low	none
10: Ellipsoidal high cond.	none	none	none	high	high	none	none	none
11: Discus	none	none	none	high	high	none	none	none
12: Bent Cigar	none	none	none	high	high	none	none	none
13: Sharp Ridge	none	none	none	low	med.	none	none	none
14: Different Powers	none	none	none	low	med.	none	none	none
15: Rastrigin multimodal	high	strong	none	low	high	low	low	none
16: Weierstrass	high	med.	none	med.	high	med.	low	none
17: Schaffer F7	high	med.	none	low	med.	med.	high	none
18: Sch. F7 mod. ill-cond.	high	med.	none	high	med.	med.	high	none
19: Griewank-Rosenbrock	high	strong	none	none	high	low	low	none
20: Schwefel	med.	deceptive	none	none	high	low	low	none
21: Gallagher 101 Peaks	med.	none	none	med.	high	med.	low	none
22: Gallagher 21 Peaks	low	none	none	med.	high	med.	med.	none
23: Katsuura	high	none	none	none	high	low	low	none
24: Lunacek bi-Rastrigin	high	weak	none	low	high	low	low	none

BBOB 2012, results

Cold start results

	Method	BBOB
	Oracle	3.5 ± 0.0
	Random	13.0 ± 0.0
	SingleBest	8.11 ± 0.7
SVM	CofiRank	5.7 ± 1.6
SVM	10NN-Sol	$\mathbf{4.9 \pm 2.3}$
SVM	FullNN-Sol	$\mathbf{4.8 \pm 3.8}$
SVM	10NN-Inst	6.3 ± 2.2
SVM	FullNN-Inst	6.0 ± 3.4
N. Network	CofiRank	6.3 ± 0.7
N. Network	10NN-Sol	$\mathbf{4.9 \pm 3.8}$
N. Network	FullNN-Sol	$\mathbf{4.9 \pm 3.8}$
N. Network	10NN-Inst	6.3 ± 2.1
N. Network	FullNN-Inst	6.1 ± 3.4

Irvine

Classifiers

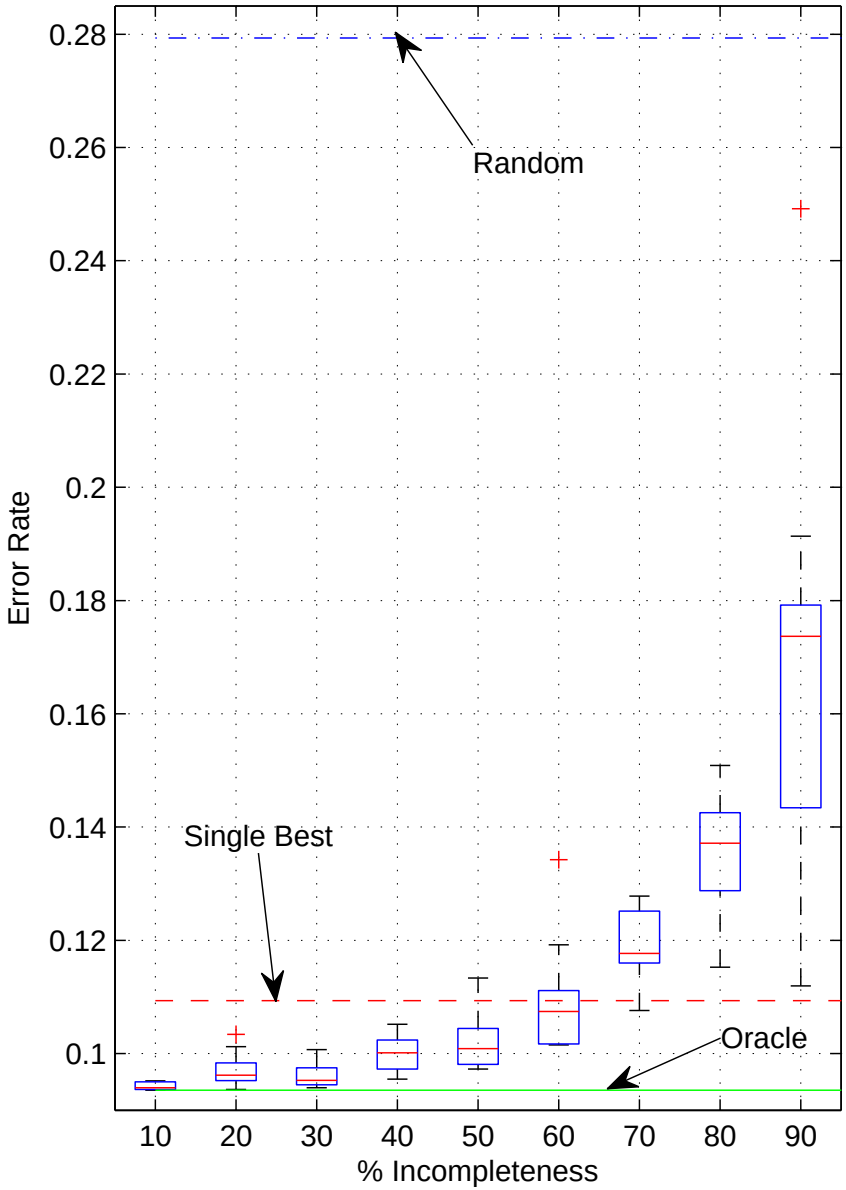
Classifier	#Parameters
Support Vector Machines (SVM)	2
Stochastic Gradient Descent (SGD)	5
Gradient Boosting Classifier (GBC)	5
Random Forests (RF)	5
Decision Trees (DT)	4
Extremely Randomized Trees (ERT)	5

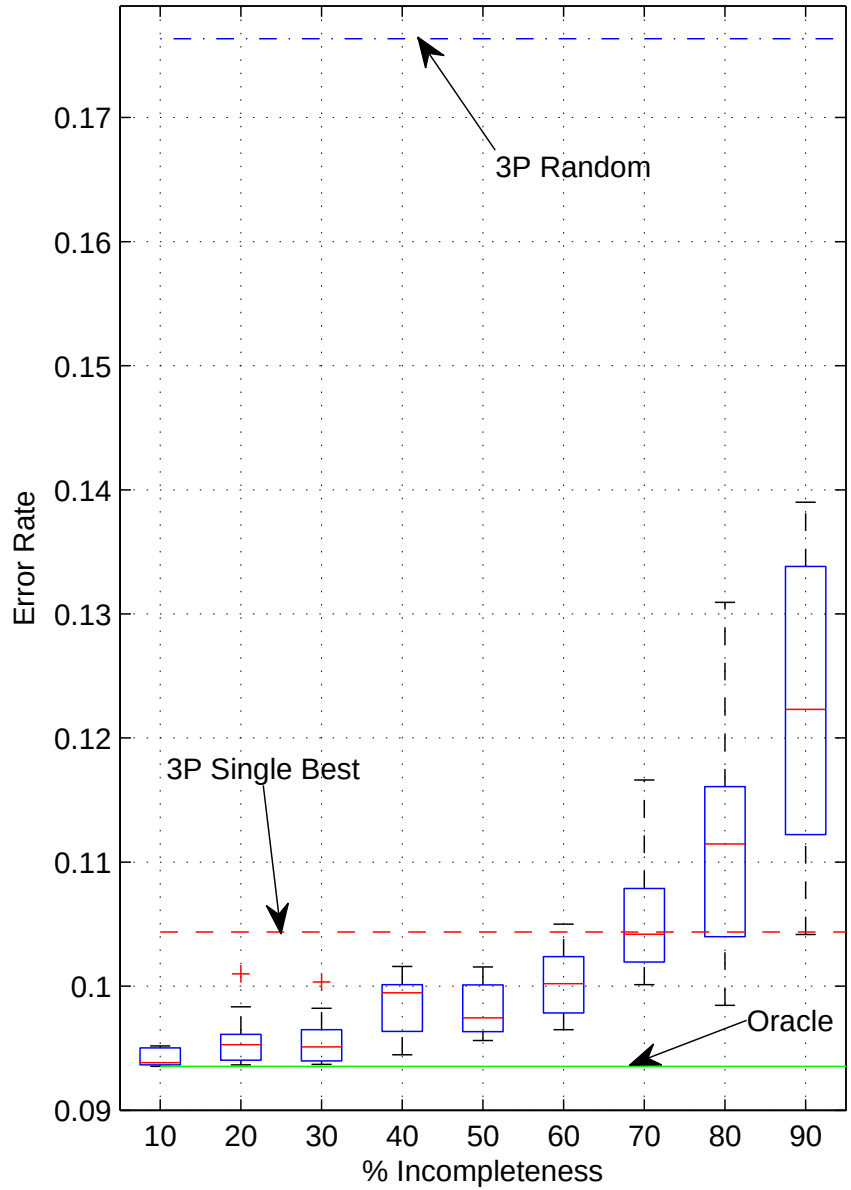
Irvine

Datasets

Dataset	# Classes	# Attributes (C|D)	# Examples
Adult	2	14 (6|8)	48842
Australian	2	14 (6|8)	690
Balance	3	4 (0|4)	625
Breast (W)	2	9 (0|9)	699
Cleveland	2	13 (0|13)	303
Credit	2	15 (6|9)	690
Diabetes	2	8 (2|6)	768
German	2	24 (0|24)	1000
Glass	7	10 (9|0)	214
Heart	2	13 (4|9)	270
Hepatitis	2	19 (1|18)	155
Ionosphere	2	34 (34|0)	351
Iris	3	4 (4|0)	150
Letter	26	16 (0|16)	20000
Monks-1	2	6 (0|6)	432
Monks-2	2	6 (0|6)	432
Monks-3	2	6 (0|6)	432
Mushroom	2	22 (0|22)	8124
Satimage	6	36 (0|36)	6435
Segment	7	19 (16|3)	2310
Sonar	2	60 (60|0)	208
Vehicle	4	18 (0|18)	846

Filling the matrix



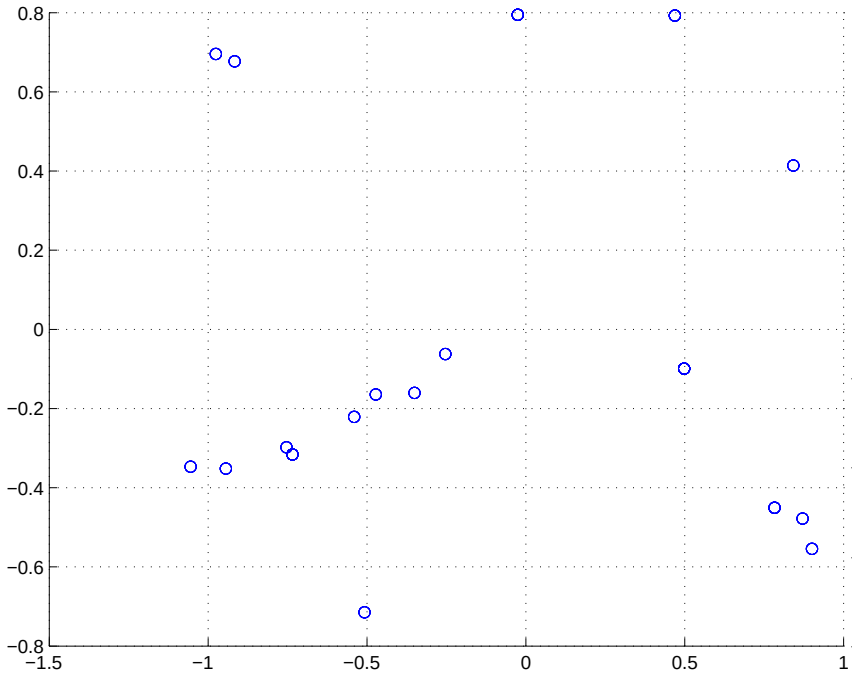


Cold Start = Failure

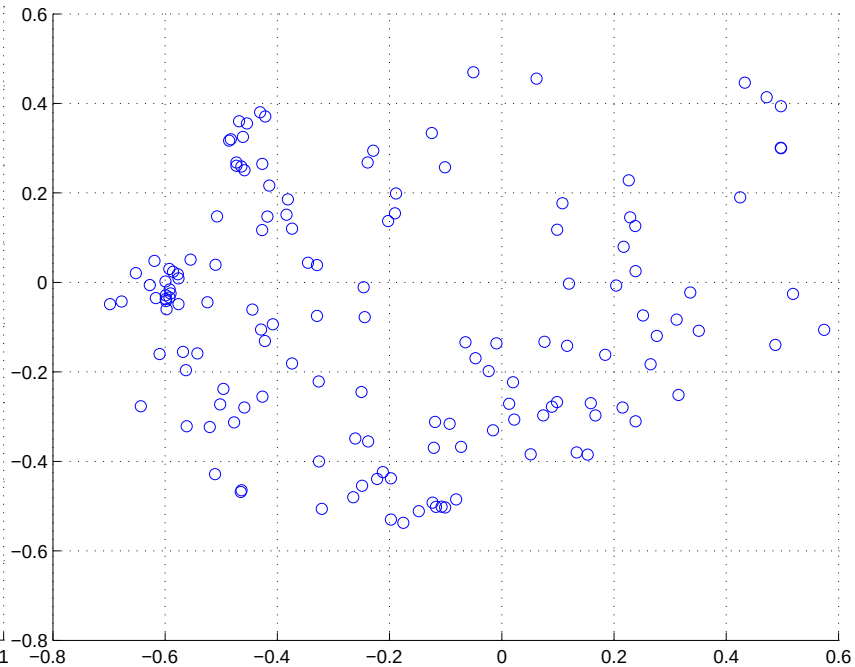
Where we learn something about the problems

Each pb instance: a vector in $\mathbb{R}^d$; mapped onto $\mathbb{R}^2$ using Multi-dimensional scaling.

Initial features (10)

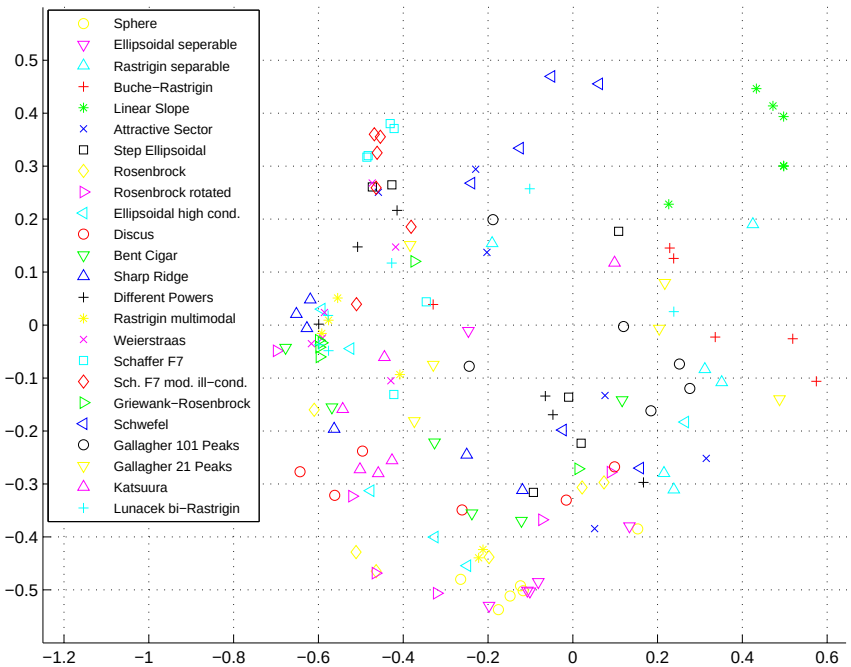


Latent features (10)

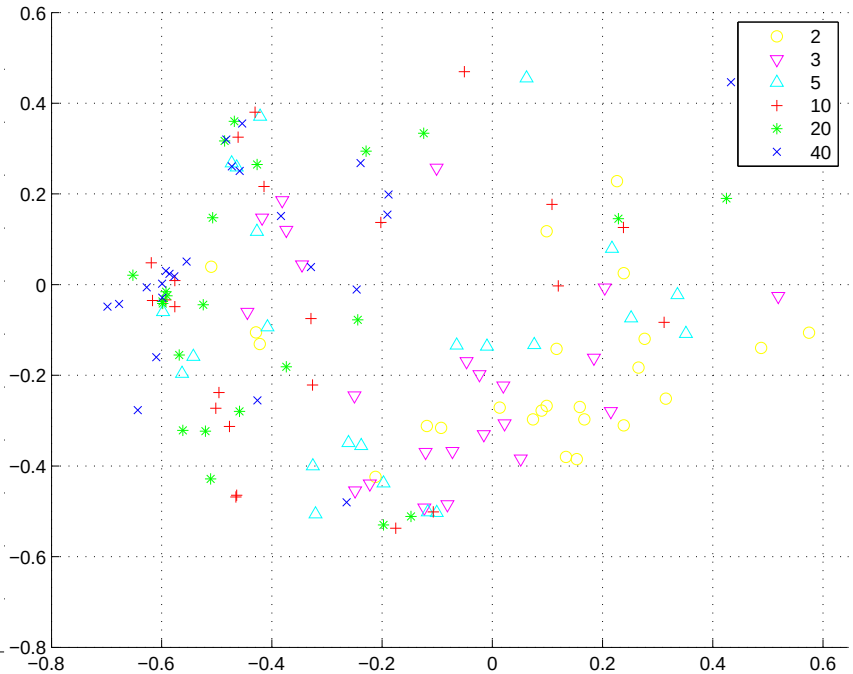


Where we learn something about the problems (BBOB)

Color = fonction

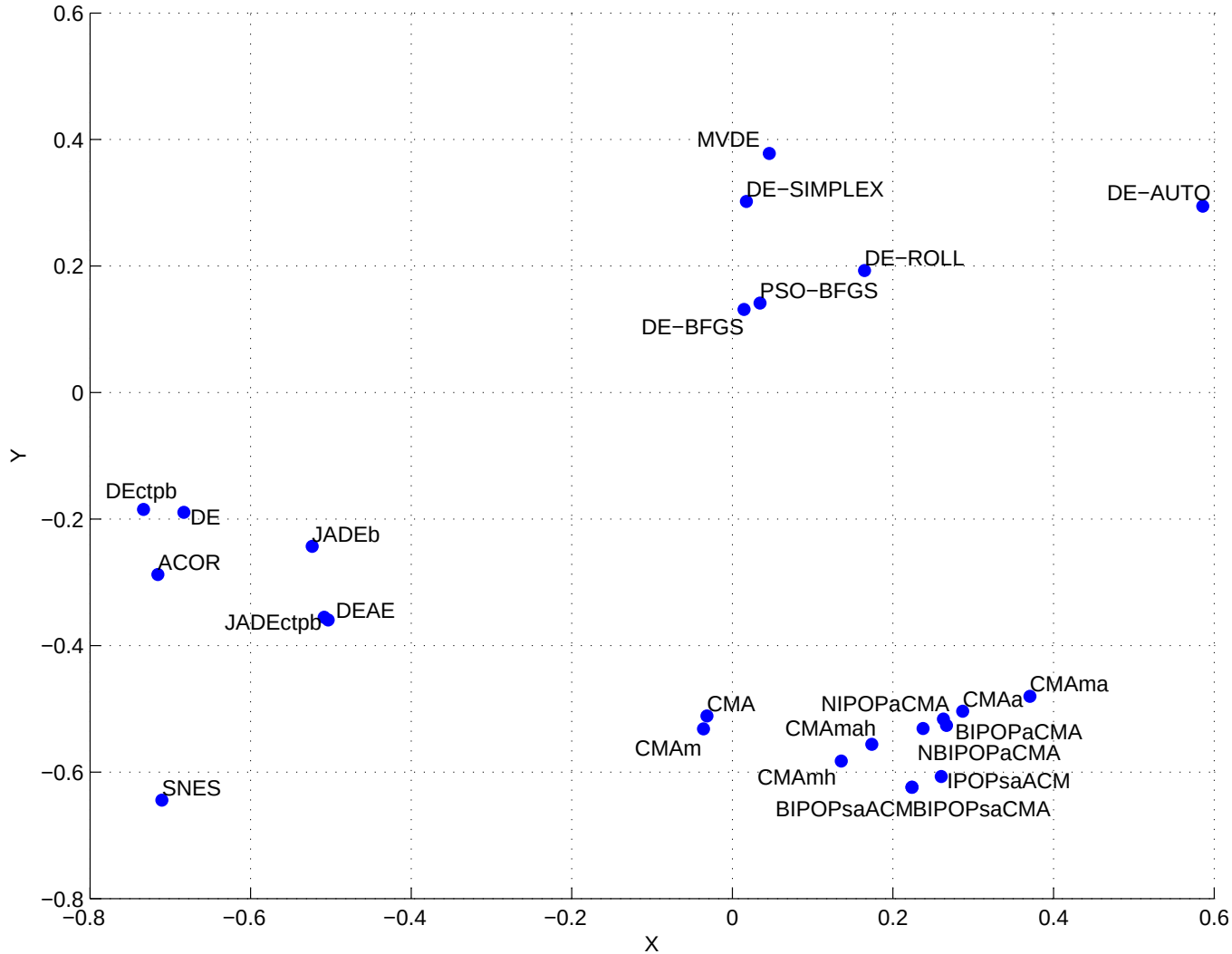


Color = dimension

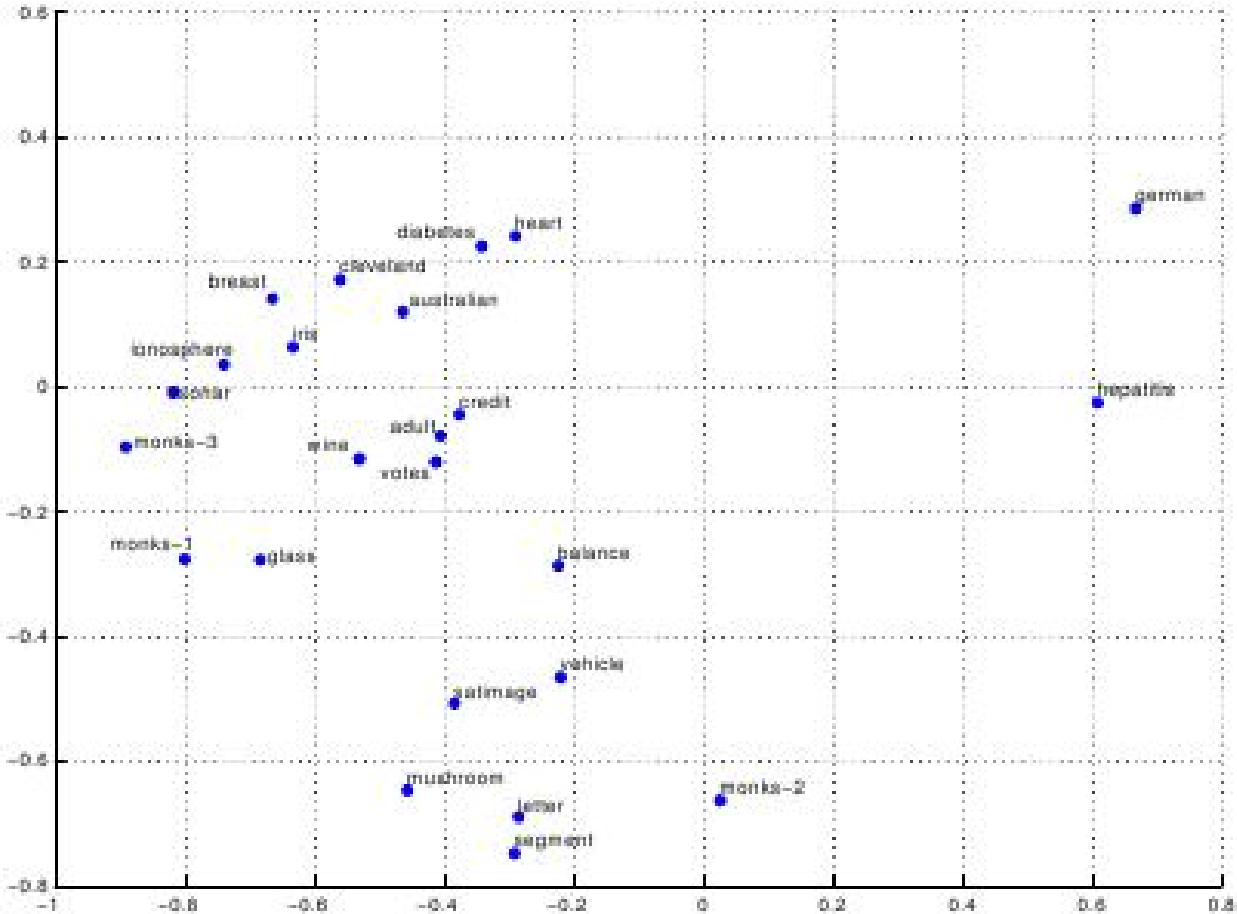


Where we learn something about the algorithms (BBOB)

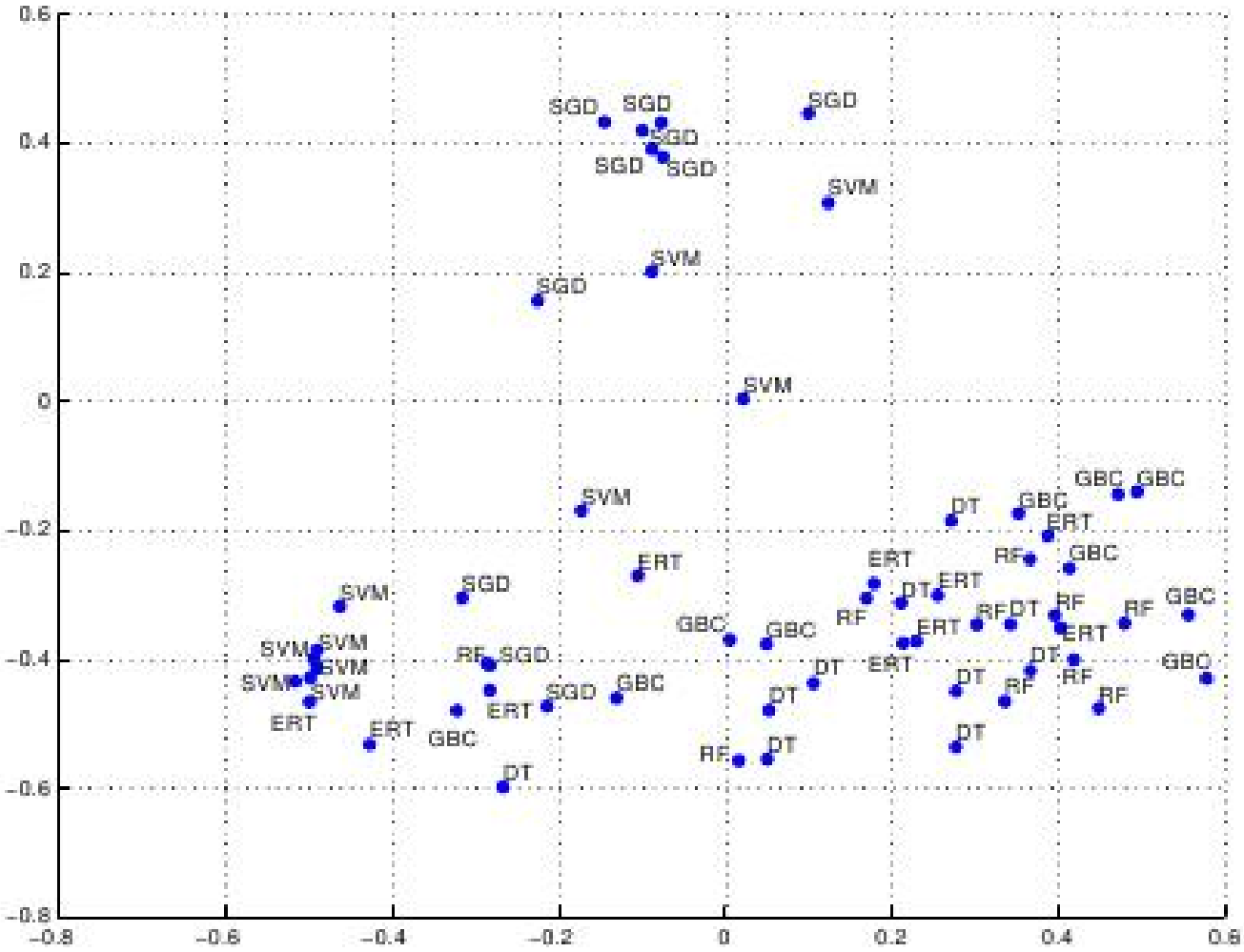
Each alg. a vector in $\mathbb{R}^d$; mapped onto $\mathbb{R}^2$ using MDS.



Where we learn something about the problems (Irvine)



Where we learn something about the algorithms (Irvine)



Conclusion

- Algorithm recommender system works
- Cold start requires initial features
  - These can be poorly informative (BBOB)
  - Current ML features are not informative enough
- Provides educated (latent) features

Perspectives

Use latent features in order

- Assess a benchmark suite (diversity);
- Assess a validation procedure (coverage of the benchmark suite used to validate a new algorithm)
- Assess novelty of an algorithm

Learn descriptive features

- using clusters based on latent features