# Non-Model-Based Algorithm Portfolios for SAT

Yuri Malitsky[2], Ashish Sabharwal[1], Horst Samulowitz[1], Meinolf Sellmann[1]

IBM · BROWN

## Algorithm Portfolios for SAT

**Motivation**

- SAT community has produced dozens of excellent solvers!
  - complementary strengths: no single solver 'wins' on all benchmarks ☹
  - algorithm portfolios: *given F, can we predict which solver will work best on F?*
- Dominant technique: runtime prediction, e.g., highly successful SATzilla variants
  - limitation: must fit a rather simplistic runtime model to complex solver behavior
- Observation: all we need for portfolios is name of best solver, not actual runtime!

**Main Findings**

- ***A simple k-NN classifier can outperform state-of-the-art portfolio solvers for SAT***
- E.g., improves upon SATzilla_R, gold medal winner, random category, Competition 2009
- Further improvements: distance-weighting, clustering, and solver scheduling **[CP-2011]**

## k-NN Classification for Algorithm Selection:

**(enhanced version participating in SAT Competition 2011)**

**Training Phase (offline):**

$T_{train}$ (training set) → compute features** & runtimes of all $F \in T_{train}$ → repeat for $k \in \{1, 2, ..., 200, ...\}$ [ repeat for 100 random 70-30 base-validation splits of $T_{train}$ [ for all $F \in T_{validation}$: identify $k$ nearest* nbrs $T_{nbrs} \subseteq T_{base}$; $S$ = solver with best PAR10 on $T_{nbrs}$; performance = PAR10(S, F); output overall performance on $T_{validation}$ ] ] → "best" k (for $T_{train}$)

** features: 48 core SATzilla features
* distance: Euclidean, $L^2$

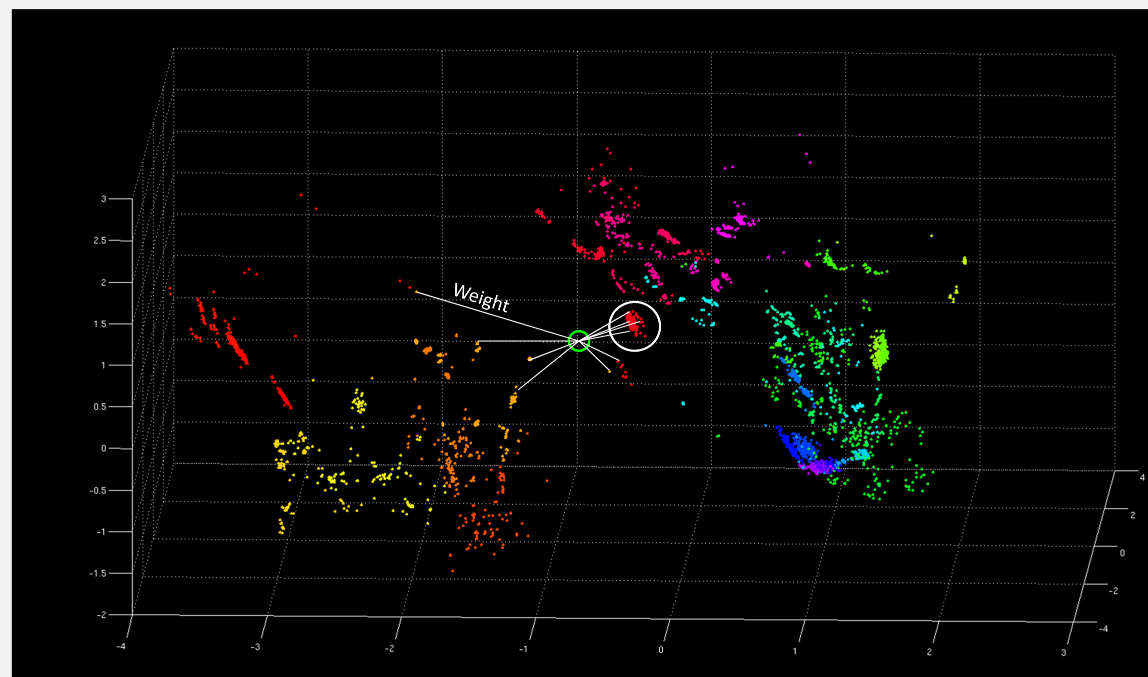
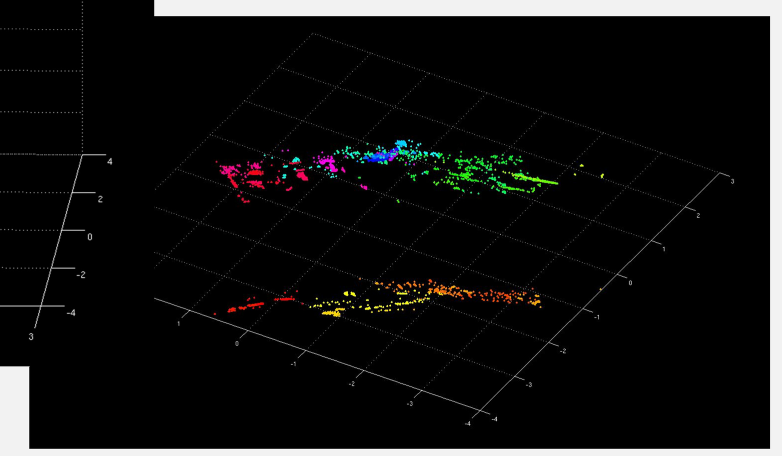
**Solver Selection:**

$T_{train}$: training set (with features and runtimes)
k : "trained" neighborhood size

instance F → compute features of F → identify **k** nearest nbrs $T_{nbrs} \subseteq T_{train}$; output solver with best PAR10 on $T_{nbrs}$ → solver **S**

## SAT Instances in the Feature Space

**Working hypothesis**:
instances close* in this space
are best solved by similar solvers
⇒ **ask neighbors** rather than,
e.g., try to predict runtime

[ "3D" projection of PCA on the 48-dimensional feature space ]

* distance: Euclidean, $L^2$

## Experimental Results (sample)

Base solvers: those used in SATzilla_R (2009 Competition version)
Training instances: random category, SAT Comp. 2002-2007 | Testing: random, SAT Comp. 2009

| | Pure Solvers | | | | | | | Portfolios | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | agw-sat0 | agw-sat+ | gnov-elty+ | **kcnfs** | march | pico-sat | SAT-enstein | SAT-zilla | **12-NN** | VBS |
| PAR10 | 6400 | 6667 | 6362 | **5813** | 6524 | 7384 | 7089 | 4399 | **3940** | 3454 |
| Avg Time | 678 | 698 | 677 | **659** | 688 | 752 | 722 | 534 | **529** | 480 |
| # Solved | 268 | 255 | 270 | **298** | 262 | 220 | 234 | 366 | **390** | 413 |
| % Solved | 47.0 | 44.7 | 47.4 | **52.3** | 46.0 | 38.6 | 41.1 | 64.2 | **68.4** | 72.5 |

68 more instances solved (closes 55% of gap to VBS)

*24 additional solved (closes **80% of gap**)*

## Boosting the Performance of k-NN Portfolios [CP-2011]

### (a) distance-based weighting (b) clustering (c) solver scheduling

*Challenging benchmark*: a mix of 5567 application, crafted, and random instances from SAT Competitions 2002-2009; split 10-ways into 70-30 training-test datasets in a "realistic" / "mean" fashion: complete instance families missing from training!

| | Basic k-NN | Weighting | Clustering | **Weight.+Clust.** |
|---|---|---|---|---|
| # Solved | 1609 | 1611 | 1615 | **1617** (9/10) |
| # Unsolved | 114 | 112 | 108 | **106** (9/10) |
| % Solved | 93.5 | 93.6 | 93.8 | **93.9** (9/10) |
| Avg Runtime | 588 | 584 | 584 | **577** (7/10) |
| PAR10 Score | 3518 | 3459 | 3369 | **3314** (8/10) |

⇒

| Fixed-Split Schedules | | | |
|---|---|---|---|
| Basic k-NN | Weighting | Clustering | **Wtg+Clu** |
| 1637 | 1641 | 1638 | **1642** (9/10) |
| 86 | 82 | 85 | **81** (9/10) |
| 95.0 | 95.3 | 95.1 | **95.3** (9/10) |
| 455 | 446 | 452 | **445** (9/10) |
| 2683 | 2567 | 2652 | **2551** (9/10) |