



Machine Learning

Clause DB Managment

Mate Soos, Raghav Kulkarni, Kuldeep
Meel



Setup + Goals

- “Glue”, “activity”, “last_used” etc. are features.
- Humans are writing the predictors (cutoffs etc), based on runtimes, and intuition
- Idea 1: Let’s dump a ton of data about clauses, solver state, CNF, etc. + modify DRAT to give us clause use
- Idea 2: Replace current generation of clause database maintenance with supervised learning-based predictors

ML in SAT/SMT

- “Learning to Solve SMT Formulas” – M. Balunovi, P. Bielik, M. Vechev (NIPS’18)
- **“Learning Rate Based Branching Heuristic for SAT Solvers”** by J. H. Liang, V. Ganesh, P. Poupart, and K. Czarnecki (SAT’16)
- “SpySMAC: Automated Configuration and Performance Analysis of SAT Solvers” – S. Falkner, M. Lindauer, F. Hutter (SAT’15)
- SATzilla-07: The Design and Analysis of an Algorithm Portfolio for SAT – L. Xu, F. Hutter, H. H. Hoos, K. Leyton-Brown (CP’07)



More ML

- Lingeling uses kNN for deciding configuration
- CryptoMiniSat uses SatZilla-based features + decision tree to decide on configuration
- There are others I guess?
- [Blank space for everyone else's stuff]

CryptoMinSat modifications

- SQLite data dumping
 - SatZilla features computed every N confls
 - Restart features for every restart
 - Clause features for every learned clause
 - Clause DB features at ever 10k confls
- Clauses have unique Clause ID
- Clause ID is dumped into all the above + DRAT
- Conflict no. when clause is generated dumped to DRAT



DRAT

- Each clause is marked with: ID, conflict no. it was generated in
- Tracks all marked clauses for which conflicts it was used exactly
- Dumps full exact data to a binary format
- Dumps aggregate data (when used 1st, when used last, how many times used) as well

What is that data you mentioned?

- 2012 – http://msoos.org/sat_visualization/
- 218 features in total for each clause at every 10k confl – decision to keep/throw
- ~50 restart stats
- ~15 clause DB stats
- ~50 clause stats
- ~60 CNF stats (some from SATZilla)
- ~30 computed stats (e.g. glue/avg. glue)
- 8 DRAT-based data

Gathering data

- Run on a wide variety of problems. But what is variety? MD5-1, MD5-2, are they different? No, so take only one! Which one??
- Difficult+easy problems!?! Only difficult?
- Data is tainted with already existing predictor. So, keep all clauses. But that runs out of time+mem.
- UNSAT: use DRAT on its own, easy!
- SAT. Oops.

Price model for keeping a clause

- So now we know how often a clause is used + what is the last time it was useful + how dispersed its use is
- (Any other data you'd like to have? Would it help?)
- **Should we keep it?**
- There is clearly a trade-off, as a form of the clause can be re-learned later
- **We still need to write a predictor!** But it's not about raw features of the clause, but meta-features...

What is wrong?

- DRAT is actually **not** what the SAT solver did
 - its ONE possible proof
- Inprocessing deletes some learnt clauses, perturbing data
- Re-running the SAT solver again with “fixed” Clause DB is not possible: dynamic restarts, VSIDS not the same



Demo time