CSC5240 Paper Presentation

Understanding Random SAT: Beyond the Clauses-to-Variables Ratio

XIAO Zigang
zgxiao@cse.cuhk.edu.hk

Department of Computer Science and Engineering
The Chinese University of Hong Kong

November 14th, 2008

Outline

Outline

Contribution of their work

1. Fact:
   - Empirical hardness of k-SAT is **correlated** with ratio of **clauses to variables**(c/v)
2. Goal:
   - Use **inexpensive computable feature** to predict run-time
   - Use *hardness model* to **choose** algorithm per instance
3. Approach:
   - Identify features using machine learning
   - Build models using previous result
   - Construct an algorithm portfolio
   - Predict algorithm runtime and choose best

Outline

SAT,3-SAT and SAT solver

- SATisfiability: Given a formula of the propositional calculus, decide if there is an **assignment** to its variables that **makes the formula true**
  - e.g. $(x_1 \wedge x_2) \vee ((x1 \wedge \neg x_3) \wedge (x_3 \vee \neg x_4))$
- Important in Computer Science, especially AI
  - Simple, fundamental
  - Prototypical *NP-hard problem* [1]
  - Can be **reduced** to many other NPC problem
- 3-SAT: *conjunctive normal form* with 3 variables per clause
  - parameter: n variables, c clauses and v variables per clause
  - e.g. $n = 5, c = 2, v = 3$
  - $(x_1 \vee \neg x_2 \vee x_5) \wedge (x_2 \vee x_3 \vee \neg x_4)$

[1] Cook, Stephen ,The complexity of theorem proving procedures, Proc. of 3rd ACM Symposium on Theory of Computing, 151-158,1971

Outline

Really hard problems [3]

- "Phase Transition" [2] : **hardness** of random instances of various NPC problems
- Conjecture:
  - All NPC problems have at least one **order parameter(op)**
  - Instances become hard when **op** is around a critical value
- The critical value separate regions:
  - *over-constrained*
  - *under-constrained*
- **Preserved** when reducing different hard problems
  - e.g. hard to color K-col graphs → hard to solve K-sat

[2] R Monasson, R Zecchina, S Kirkpatrick, B Selman, Lidor Troyansky, Determining computational complexity from characteristic 'phase transitions', Nature,1999

[3] Cheeseman,Kanefsky,Taylor,Where the really hard problems are, In Proc. IJCAI-1991,331-337,1991

Intuitive understanding of boundary value

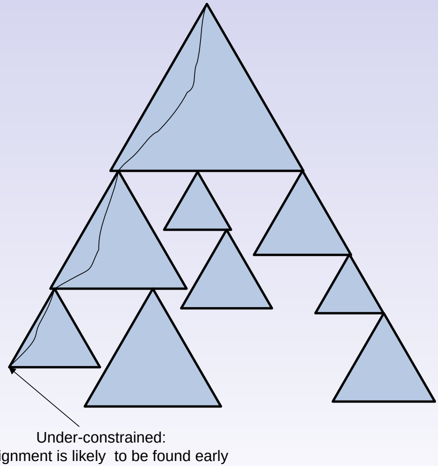


Figure: Illustration of *under-constrained*

Intuitive understanding of boundary value(cont.)

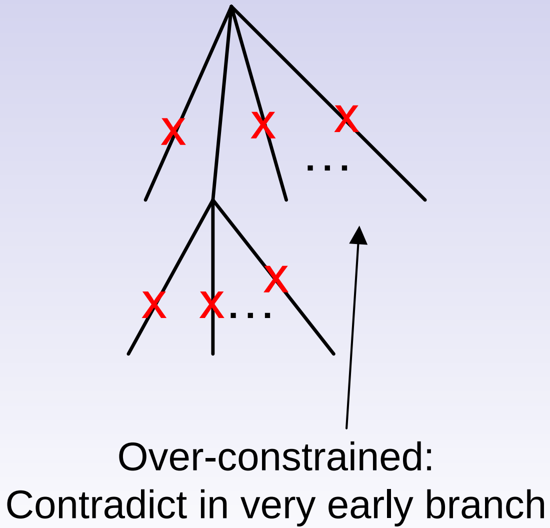


Figure: Illustration of *over-constrained*

Intuitive understanding of boundary value(cont.)

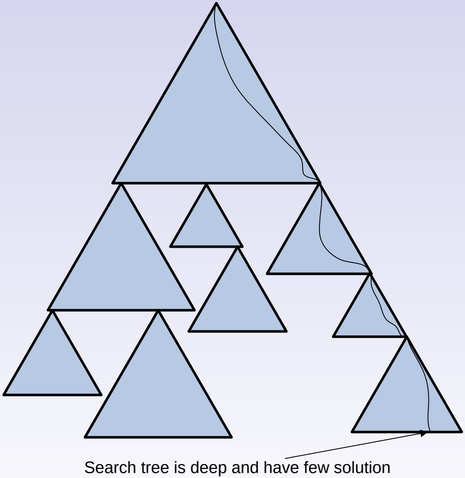


Figure: Illustration of *in between* formulas

Generating hard problems

- Selman et al. distinguish two instance distribution of SAT [4]
  - Fixed clause-length (3-SAT)
    - hard when reaching boundary value
  - Constant-density model($P(x_i)$=p)
    - easy anyway
- "50% satisfiable" point
  - occur at **fixed ratio** of c/v : 4.26
- Implication: larger formula is **not necessarily** harder
- Algorithm: *Worst case analysis* vs. *empirical behavior*

[4] Selman,Mitchell,Levesque,Generating hard satisfiability problems, *Artificial Intelligence* 81(1-2):17-29.1996

Ratio of clauses-to-variables [5]

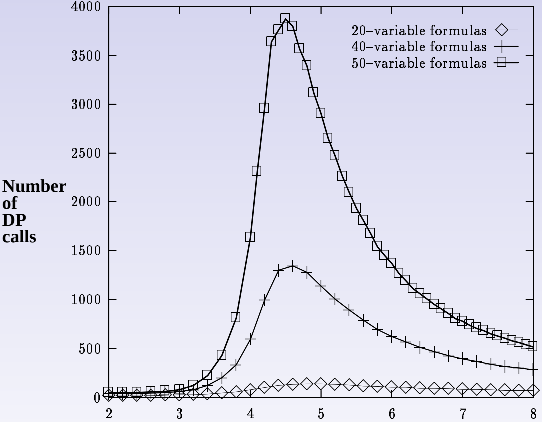


Figure: Median DP calls for 50-variable Random 3-SAT as a function of the ratio of clauses-to-variables

[5] Selman,Mitchell,Levesque,Generating hard satisfiability problems, *Artificial Intelligence* 81(1-2):17-29.1996

Ratio of clauses-to-variables [6]

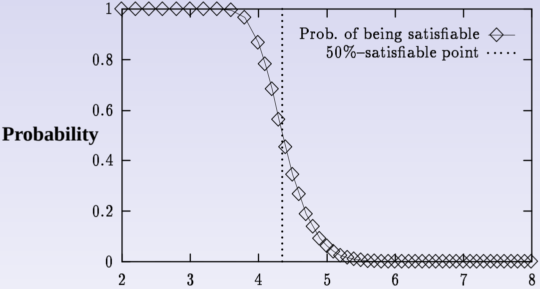
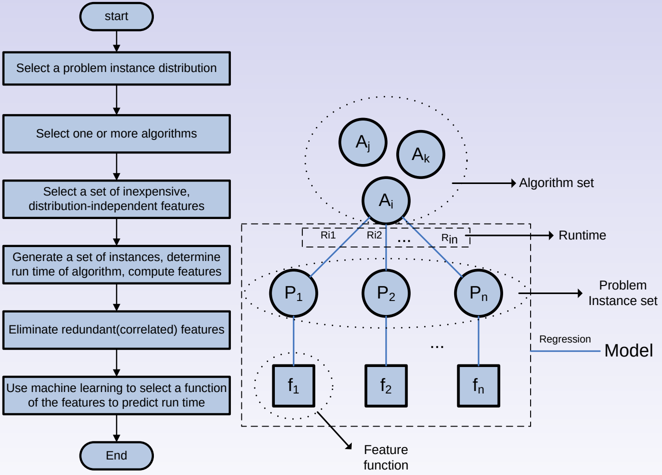


Figure: Probability of satisfiability of 50-variable formulas, as a function of the ratio of clauses-to-variables.

[6] Selman, Mitchell, Levesque, Generating hard satisfiability problems, *Artificial Intelligence* 81(1-2):17-29.1996

Outline

Building Empirical Hardness Model



(a) (b)

Figure: Construction of Empirical Hardness Model

Basic idea

- Use supervised learning
  - Choose a function from a given hypothesis space
    - $f : feature^n \rightarrow run\_time$
    - Minimize a given error metric
- Their approach:
  - Regression technique: *linear regression(LR)*
  - Error metric: *root mean squared error(RMSE)*
- LR is appealing due to the small computational cost
  - Choosing a good hypothesis space
  - Choosing an appropriate error metric

Extending linear regression

LR seems quite limited,but can be extended:

1. By including all pairwise products of features, e.g. C_n^2
   - Resulting *quadratic manifold* in the original feature space
   - Only k most important features' pairwise products to avoid becoming *intractable*
2. By transforming *non-linear*(e.g.sigmoid) to *linear*
   - Suppose hypothesis spaces of the form $h(y)$
   - Replace response variable y by inverse function h^{-1}
   - Besides linear model,exponential and logistic models are used
     - $h(y) = 10^y; h^{-1}(y) = log_{10}(y)$
     - $h(y) = 1/(1 + e^{-y}); h^{-1}(y) = ln(y)ln(1 - y)$

Outline

Features

- Totally 91 features used,divided into 9 groups

Problem Size Features:

1. **Number of clauses:** denoted c
2. **Number of variables:** denoted v

3-5. **Ratio:** c/v, $(c/v)^2$, $(c/v)^3$

6-8. **Ratio reciprocal:** (v/c), $(v/c)^2$, $(v/c)^3$

9-11. **Linearized ratio:** $|4.26 - c/v|$, $|4.26 - c/v|^2$, $|4.26 - c/v|^3$

Variable-Clause Graph Features:

12-16. **Variable nodes degree statistics**: mean, variation coefficient, min, max and entropy.

17-21. **Clause nodes degree statistics**: mean, variation coefficient, min, max and entropy.

Variable Graph Features:

22-25. **Nodes degree statistics**: mean, variation coefficient, min, and max.

Clause Graph Features:

26-32. **Nodes degree statistics**: mean, variation coefficient, min, max, and entropy.

33-35. **Weighted clustering coefficient statistics**: mean, variation coefficient, min, max, and entropy.

Balance Features:

36-40. **Ratio of positive and negative literals in each clause**: mean, variation coefficient, min, max, and entropy.

41-45. **Ratio of positive and negative occurrences of each variable**: mean, variation coefficient, min, max, and entropy.

46-48. **Fraction of unary, binary, and ternary clauses**

Proximity to Horn Formula

49. **Fraction of Horn clauses**

50-54. **Number of occurrences in a Horn clause for each variable** : mean, variation coefficient, min, max, and entropy.

LP-Based Features:

55. **Objective value of linear programming relaxation**
56. **Fraction of variables set to 0 or 1**

57-60. **Variable integer slack statistics:** mean, variation coefficient, min, max.

DPLL Search Space:

61-65. **Number of unit propagations:** computed at depths 1, 4, 16, 64 and 256

66-67. **Search space size estimate:** mean depth to contradiction, estimate of the log of number of nodes.

Local Search Probes:

68-71. **Minimum fraction of unsat clauses in a run:** mean and variation coefficient for SAPS and GSAT (see [17]).

72-81. **Number of steps to the best local minimum in a run:** mean, median, variation coefficient, 10^{th} and 90^{th} percentiles for SAPS and GSAT.

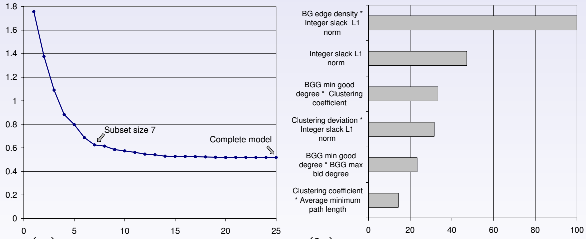
82-85. **Average improvement to best:** For each run, we calculate the mean improvement per step to best solution. We then compute mean and variation coefficient over all runs for SAPS and GSAT.

86-89. **Fraction of improvement due to first local minimum:** mean and variation coefficient for SAPS and GSAT.

90-91. **Coefficient of variation of the number of unsatisfied clauses in each local minimum:** mean over all runs for SAPS and GSAT.

Building smaller models

1. Discard **highly correlated** or **uninformative** features
   - e.g. when c/v is fixed, $(c/v), (c/v)^2$ etc. is not needed
2. Use statistical technique to evaluate **importance** of features
   - Compute *cost of omission* (with & without)
   - Use *cross-validation* (split dataset)
3. Choose appropriate small **subset** [7]



(a) subset size vs. RMSE (b) Feature omission example

[7] K. Leyton-Brown, E. Nudelman, and Y. Shoham. Learning the empirical hardness of optimization problems: The case of combinatorial auctions. In Proc. CP-2002, pages

Outline

Experimental Setup

- Two dataset:
  - 20000 uniform random 3-SAT instances with 400 variables
    - Varied ratio: $C/V \in [3.26, 5.26]$
  - 20000 uniform random 3-SAT instances with 400 variables,1704 clauses
    - Fixed ratio: $C/V = 4.26$
  - Each dataset split into 3 parts
    - training (for tuning)
    - test (for testing)
    - validation (for tuning)
- Three algorithms:
  - `ksnfs`
  - `oksolver`
  - `satz`
- Platform: 2.4GHz Xeon processors, Linux 2.4.20
- Machine learning tools: *R* and *Matlab*

Outline

Building different models

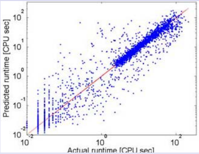


Figure: Actual vs. predicted runtimes for kcnfs in quadratic case

- linear,logistic,exponential models + 91 features
  - Linear the worst
  - Others similar, **logistic** slightly better
- Consider quadratic expansion of features. After expansion, preserve 360 features
  - All three are better
  - **Logistic** the best

Decrease subset size

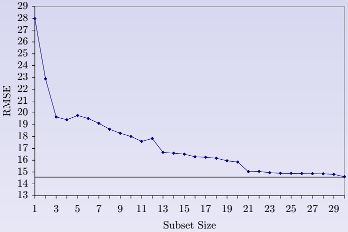


Figure: RMSE as a function of subset size

- We want a small subset containing **core** features
  - **Sufficient** to approximate full model
  - **Computing time** also decreases
- Enumerate subset size and calculate RMSE
  - Choose smallest subset at which **little incremental benefit** can be gained
  - Subset of size 4,RMSE=19.42 is chosen here

Identifying important features: *cost of omission*

Variable	Cost of Omission
$\|c/v - 4.26\|$ [9]	100
$\|c/v - 4.26\|^2$ [10]	69
$(v/c)^2 \times$ SAPS_BestCoeffVar_Mean [7 × 90]	53
$\|(c/v) - 4.26\| \times$ SAPS_BestCoeffVar_Mean [9 × 90]	33

Table: Variable importance in size 4 model for variable-ratio instances

- The most important one is c/v ratio, supporting “phase transition”
- Note that the remaining feature are **local search probing feature**
    - Suggests local minima corresponds to large subtrees with no solution
- Also note that we may explore new understanding from remaining feature

Prediction on satisfiable and unsatisfiable

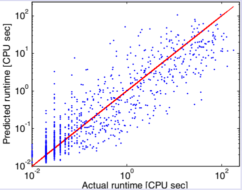


Figure: Actual vs. predicted runtimes for `kcnfs on satisfiable instances`

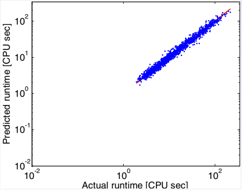
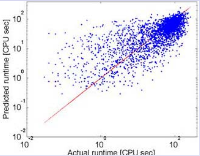


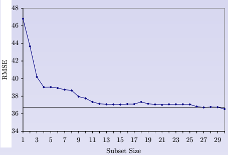
Figure: Actual vs. predicted runtimes for `kcnfs on unsatisfiable instances`

Outline

Fixed-Ratio experiment



(a) fixed-ratio predict runtime



(b) RMSE as a function of subset size

Variable	Cost of Omission
SAPS_BestSolution_Mean^2 [68^2]	100
SAPS_BestSolution_Mean $\times$ Mean_DPLL_Depth [68×66]	74
GSAT_BestSolution_CoeffVar $\times$ Mean_DPLL_Depth [71×66]	21
VCG_CLAUSE_Mean $\times$ GSAT_FirstLMRatio_Mean [17×88]	9

- What if we **fix** clause-to-variable?
- Challenge: identifying other features for hardness
- Again we reached the best using logistic model in quadratic expansion
  - Dominant feature: local search and DPLL probing features
  - Captures the degree to which a given instance has “almost” satisfying assignments

Outline

Application of Empirical Hardness Model

1. Harder instance generator of random 3-SAT
2. Construction of algorithm portfolios – SATzilla
   - Consists of `2clseq`, `eqSatz`, `HeerHugo`, `JeruSat`, ...
   - Win award in SAT competition
   - Newer version in 2007: SATzilla 07

SATzilla

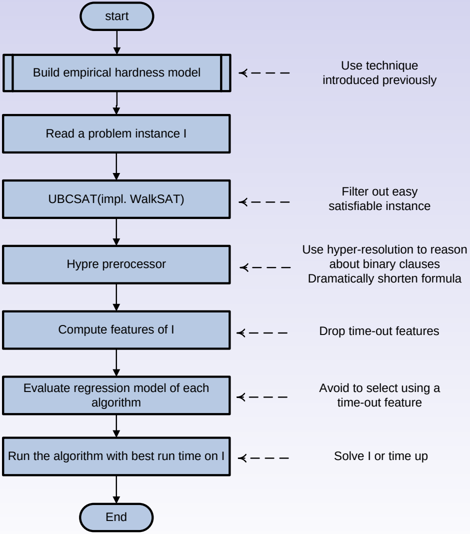


Figure: Work flow of SATzilla

Outline

CSC5240 Paper Presentation

Conclusion and future work

- Empirical hardness model is valuable for the study of empirical behavior of complex algorithm
- Future work:
  1. Apply empirical hardness model to stochastic search algorithm
  2. Build stronger structural/hierachical model
  3. Study how some features cause instance to be hard or easy for certain types of algorithms

Q & A and Acknowledgement

- Thank You -

- Thanks to Nudelman et al. for their excellent work
- Thanks to *HUANG Zheng-hua* in *Wuhan University* for providing this beamer template(adapted)