A New Technique for Variable Cluster Analysis in Propositional Formulas

Allen Van Gelder
Computer Science Dept.
Univ. of California
Santa Cruz, CA, USA

http://www.cse.ucsc.edu/~avg/

These slides are eigsat-pos16-trans.pdf

What is Tie Strength?

Social Media Context [Gupte and Eliassi-Rad, WebSci 2012]:

- How strongly are two "people" (generic term) connected?
- Estimate by common connections to "events" (generic term)
- Many ways to measure importance of an "event".
- Duality of "people" and "events": how strongly are two events connected?

SAT Context [Ansótegui, Giráldez-Cru, Levy, SAT 2012]:

- How strongly are two variables "connected"?
- Measure by clauses containing both variables.
- Polarity of literals in common clause is ignored.

What are clusters?

Origins in Physics and Biology
[Clauset, Newman, and Moore, Physical Reviews E (2004)]:

- Many kinds of large networks occur naturally.
- Partition the universe of objects into disjoint sets called "clusters".
- Edges (possibly weighted) between objects in different clusters are bad.
- Edges (possibly weighted) between objects in same cluster are good.
- Heuristic definition of partition quality $Q \leq 1$.

Computation of a partition with highest Q is believed to be exponential.

Several greedy algorithms in the literature:

- [Clauset, Newman, and Moore, Physical Reviews E (2004)]
- Etc.

Published greedy algorithms do not scale to large SAT instances

- [Ansótegui, Giráldez-Cru, Levy, SAT 2012]
- Newsham, Ganesh, Fischmeister, Audemard, Simon SAT 2014
- Etc.

Natural Tie Strength Between Clause Pairs

If the CNF has n variables, there are 2^n total assignments.

One clause of width k eliminates 2^k total assignments.

- Fraction is 2^{-k}.
- Natural measure of the strength of its constraint.

What fraction is eliminated by two clauses, $|C_1| = j$ and $|C_2| = k$?

- In general, there is redundancy.

Key Observation:

If C_1 and C_2 have some clashing variable(s), there is no redundancy.

- Eliminated fraction = $2^{-j} + 2^{-k}$.
- Multiple or unique clashing variables gives same fraction.
- Natural measure of tie strength between C_1 and C_2.

Clashing Neighbor Relation

Weighted tie-strength of "neighbor" clause pairs (C_1, C_2)

- Clause pairs with clashing variables have edge weight $W_{cl}(C_1, C_2) = 2^{-j} + 2^{-k}$;
- Relation between clause pair without clashing variables is considered weak and unimportant;
  i.e., $W_{cl}(C_1, C_2) = 0$ in this case;
  Important for sparcity of the matrix.
  Indirect relation through a common C_3 is still possible.

Computation of Clashing Neighbor Graph (Matrix) by Linear Algebra

CNF $\mathcal{F}$ has m clauses and n variables.

Clause-Variable Incidence Graph [Ansótegui, Giráldez-Cru, Levy, SAT 2012]:

- Edge between clause C and variable v if v is in C with either polarity.
- Edges are undirected.
- Directed version considered by [Katsirelos and Simon, CP 2012]

Clause-Literal Incidence Graph (this talk):

- Edge between clause C and ***literal*** q if q is in C.
- Edges are undirected.
- Edge weight is $2^{-|C|}$.

Matrices for unweighted Clause-Literal Incidence Graph and clause weights:

$$F = \begin{array}{c} \\ \text{clauses} \end{array} \begin{array}{c} \begin{array}{cc} \text{pos. lits} & \text{neg. lits} \end{array} \\ \begin{array}{|c|c|} \hline P & N \\ \hline \end{array} \end{array} \qquad \begin{array}{c} \\ \text{clauses} \end{array} \begin{array}{c} \begin{array}{c} \text{clauses} \end{array} \\ \begin{array}{|c|} \hline S \\ \text{(diagonal)} \\ \hline \end{array} \end{array}$$

P and N are 0-1 matrices.

$P(i, j) = 1$ iff $x_j \in C_i$;

$N(i, j) = 1$ iff $\overline{x_j} \in C_i$;

$S(i, i) = 2^{-|C_i|}$.

Unweighted and Weighted Clashing Neighbor Matrix

$$Clash(F) = \text{logical}\left[P\right]\left[N^T\right] + \left[N\right]\left[P^T\right]$$

I.e., squash all positive entries to 1 to make $Clash(F)$ a 0-1 matrix.

$$Wcl(F) = S\,Clash(F) + Clash(F)\,S$$

Note that pre-multiplying by S scales rows; post-multiplying scales columns.

- For edge (C_i, C_j) pre-multiply gets weight of C_i and post-multiply gets weight of C_j.

Interpreting Eigenvector of $Wcl(F)$

Note that $Wcl(F)$ is symmetric with zeros on the diagonal.

"Importance" of a clause C_i is a weighted sum of the "importance" of its clashing neighbors with weights ***proportional to*** tie strength.

$\xi_i = \sum_j Wcl(i,j)\,\xi_j / \lambda$

Here, ξ is "importance" and λ is a constant to preserve Euclidean length of ξ.

In matrix notation, $\lambda\xi = Wcl\,\xi$.

Perron-Frobenius theorem: For maximum eigenvalue λ, eigenvector $\xi \geq 0$.

Because the trace of $Wcl(F)$ is zero, the sum of the eigenvalues is zero,
so some are positive, some are negative, maybe some are 0.

Design of the Experiment

150 instances that were new to the SAT competition in 2011.

- `minisat` not tuned to these instances !

Preprocessed through `simp`, AKA `satElite`.

- Two instances solved; 148 remain.

Strategy: Assuming 30 CPU-minute limit, run eigen-analysis only on instances that `minisat` does not solve in 2 minutes.

- Principal cluster = large eigenvector components, down to sharp drop.
- **Rerun**, telling `minisat` to bump "activity" in clauses of principal cluster.

Baseline: `minisat` without preprocessing for 30 CPU minutes.

Tried to find principal eigenvector with matlab (31GB limit).

- Out of memory on 41 instances; succeeded on 107.
- Averaged 35 seconds on out-of-memory instances.

instances	avg. vars	avg. clauses	avg. literals	eigenvector secs. (avg. over 107 instances)
148	97003	1096661	3010753	1083

Minisat Baseline

solved instances	secs	solved sat	secs	solved unsat	secs
81	239	41	228	40	250
solved in instances	>2 mins. secs	solved in sat	>2 mins. secs	solved in unsat	>2 mins. secs
30	610	14	621	16	600

Summary: `minisat` solved 51 instances within 2 minutes,
so eigen-analysis would be performed on the other 97.

`minisat` would consume about 38.6 hours on these 97, solving 30.

Eigen-analysis, as implemented, might consume 29 hours on these 97.
It will have to be an order of magnitude faster to have a chance of being useful.

The Good News: Sharp Drop-offs in Eigenvector

Drop-off: The fractional drop between successive components of the sorted eigenvector (0.0011 to 0.0010 is 9 percent).

Average over 95 instances is that the max drop-off averages 30 percent.
8 instances are greater than 90 percent; 24 instances are less than 3 percent.
Even 3 percent is orders of magnitude above the average.

The average number of clauses in the principal cluster is about 36 thousand, compared with about 1.1 million in the formula.

Thus the technique identifies about 3 percent of the clauses that it judges to be the most important.
Whether they really are important is conjectural at this point.