Counting Solution Clusters Using Belief Propagation

Lukas Kroc, Ashish Sabharwal, Bart Selman
Cornell University

Physics of Algorithms
Santa Fe, September 3, 2009
Constraint Satisfaction Problem (CSP)

- Constraint Satisfaction Problem $P$:
  - Input: a set $V$ of variables
  - a set of corresponding domains of variable values [discrete, finite]
  - a set of constraints on $V$ [constraint $\equiv$ set of allowed value tuples]
  - Output: a solution, valuation of variables that satisfies all constraints

Well Known CSPs:

- **k-SAT**: Boolean satisfiability
  - Domains: $\{0,1\}$ or $\{$true, false$\}$
  - Constraints: disjunctions of variables or their negations ("clauses") with exactly $k$ variables each
  - $F = (x \lor y) \land (\neg x \lor z)$

- **k-COL**: Graph coloring
  - Variables: nodes of a given graph
  - Domains: colors $1\ldots k$
  - Constraints: no two adjacent nodes get the same color.
Encoding CSPs

- One can visualize the connections between variables and constraints in so called **factor graph**:
  - A bipartite undirected graph with two types of nodes:
    - **Variables**: one node per variable
    - **Factors**: one node per constraint

- Each factor node $\alpha$ has an associated **factor function** $f_\alpha(x_\alpha)$, weighting the variable setting. For CSP, $f_\alpha(x_\alpha)=1$ iff constraint is satisfied, else $=0$
  - Weight of the full configuration $x$: $F(x) = \prod_\alpha f_\alpha(x_\alpha)$
  - Summing weights of all configurations defines **partition function**:
    $$Z = \sum_x \prod_\alpha f_\alpha(x_\alpha)$$

- For CSPs the partition function computes the number of solutions

Can we count “clusters” of solutions similarly?
Talking about Clusters

1. High density regions
   - BP for BP
     - The original SP derivation from stat. mechanics
       - [Mezard et al. '02]
       - [Mezard et al. '09]

2. Enclosing hypercubes
   - BP for “covers”
     - First rigorous derivation of SP for SAT
       - [Braunstein et al. '04]
       - [Maneva et al. '05]

3. Filling hypercubes
   - BP for $Z_{(-1)}$
     - More direct approach to clusters.
       - [Kroc, Sabharwal, Selman '08 '09]
Clusters as Combinatorial Objects

- **Definition:** A solution graph is an undirected graph where nodes correspond to solutions and are neighbors if they differ in value of only one variable.

- **Definition:** A solution cluster is a connected component of a solution graph.

- **Note:** this is not the only possible definition of a cluster
Thinking about Clusters

- Clusters are subsets of solutions, possibly exponential in size
  - not practical to work with

- To compactly represent clusters, we trade off expressive power for shorter representation
  - lose some details, but gain representability

- **Approximate by hypercubes “from outside” & “from inside”**
  - **Hypercube:** Cartesian product of non-empty subsets of variable domains
    - E.g. with $\ast = \{0,1\}$,
      - $y = (1\ast\ast)$ is a 2-dimensional hypercube in 3-dim space

    - **From outside:** The (unique) minimal hypercube enclosing the whole cluster.
    - **From inside:** A (non-unique) maximal hypercube fitting inside the cluster.
Talking about Clusters

1. High density regions
   - BP for BP
   - The original SP derivation from stat. mechanics
     - [Mezard et al. '02]
     - [Mezard et al. '09]

2. Enclosing hypercubes
   - BP for “covers”
   - First rigorous derivation of SP for SAT
     - [Braunstein et al. '04]
     - [Maneva et al. '05]

3. Filling hypercubes
   - BP for $Z_{(-1)}$
   - More direct approach to clusters.
     - [Kroc, Sabharwal, Selman '08 '09]
Factor Graph for Clusters

- To reason about clusters, we seek a factor graph representation
  - Because we can do approximate inference on factor graphs
  - Need to count clusters with an expression similar to $Z$ for solutions:
    $$Z = \sum_{x \in \{0,1\}^n} \prod_{\alpha} f_\alpha(x_\alpha)$$
    $$= F(x) = 1 \text{ iff } x \text{ is a solution}$$

- Indeed, we derive the following for approximating number of clusters:
  $$Z(-1) = \sum_{y \in \{0,1,*\}^n} (-1)^\#(y) \prod_{\alpha} f'_\alpha(y_\alpha)$$

  - Syntactically very similar to standard $Z$, which computes exactly number of solutions
  - Exactly counts clusters under certain conditions, as discussed later
  - Analogous expression can be derived for any discrete variable domain

$$f'_\alpha(y_\alpha) = \prod_{x_\alpha \in y_\alpha} f_\alpha(x_\alpha)$$
Checks whether all points in $y_\alpha$ are good
Counting Solution Clusters

Divide-and-Conquer Recursively:

- Arbitrarily pick a variable, say $x$, of formula $F$
- **Count** how many clusters contain solutions with $x=0$
  (ok if the cluster has solutions with both $x=0$ and $x=1$)
- **Add** number of clusters that contain solutions with $x=1$
- **Subtract** number of clusters that contain both solutions with $x=0$ and solutions with $x=1$

$$\#\text{clusters} = \#\text{clusters}(F)_{x=0} + \#\text{clusters}(F)_{x=1} - \#\text{clusters}(F)_{x=0 \& x=1}$$

(Inclusion - exclusion formula)

**Key issues:**

- how can we compute $\#\text{clusters}(F)_{x=0}$?
  ($\#\text{clusters}_{x=1}$ would be similar)
- how do we compute $\#\text{clusters}(F)_{x=0 \& x=1}$? (not a problem for SAT)
Computing $\#\text{clusters}(F)|_{x=0}$: **Fragmentation**

- Algorithmically, easiest way is to
  - “fix” $x$ to 0 in the formula $F$, compute $\#\text{clusters}$ in new formula $(F|_{x=0})$
  - So, use as approximation: $\#\text{clusters}(F)|_{x=0} \approx \#\text{clusters}(F|_{x=0})$

- **Risk?**
  - Potential *over-counting*: a cluster of $F$ may break/fragment into several smaller, disconnected clusters when $x$ is fixed to 0

- Interestingly: Clusters often do not fragment!

- In particular, *provably* no fragmentation in 2-SAT and 3-COL* instances! (any instance, i.e., worst-case).

- Also, *empirically* holds for almost all clusters in random 3-SAT, logistics, circuits, …
Theoretical Results: Exactness of $Z_{(-1)}$

On what kind of solution spaces does $Z_{(-1)}$ count clusters exactly?

- **Theorem**: $Z_{(-1)}$ is exact for any 2-SAT problem.
- **Theorem**: $Z_{(-1)}$ is exact for a 3-COL problem on $G$, if every connected component of $G$ has at least one triangle.

- **Theorem**: $Z_{(-1)}$ is exact if the solution space decomposes into “recursively-monotone subspaces”.
Empirical Results: $Z_{(-1)}$ for SAT

Random 3-SAT, $n=90$, $\alpha=4.0$
One point per instance

Random 3-SAT, $n=200$, $\alpha=4.0$
One point per variable
One instance
Empirical Results: $Z_{(-1)}$ for SAT

- $Z_{(-1)}$ is remarkably accurate even for many structured formulas (formulas encoding some real-world problem):

<table>
<thead>
<tr>
<th>Instance Name</th>
<th># solutions</th>
<th># clusters</th>
<th>$Z_{(-1)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>v32r250p1</td>
<td>52081218</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>v32r500p5</td>
<td>1543304664</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>driverlog1_ks99i</td>
<td>856152</td>
<td>338100</td>
<td>338100</td>
</tr>
<tr>
<td>rovers1_ks99i</td>
<td>17850294</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>rovers1_v01a</td>
<td>83200608</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>rovers1_v01i</td>
<td>266000</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>rovers2_ks99i</td>
<td>531360</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>rovers2_v01a</td>
<td>52107696</td>
<td>316</td>
<td>308</td>
</tr>
<tr>
<td>rovers2_v01i</td>
<td>21504</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>rovers4_ks99i</td>
<td>13794198600</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>rovers4_v01a</td>
<td>2592794880</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>rovers4_v01i</td>
<td>28447200</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>
BP for Estimating $Z_{(-1)}$

- Recall that the number of clusters is very well approximated by
  \[
  Z_{(-1)} = \sum_{\mathbf{y} \in \{0, 1, *\}^n} (-1)^{\#*\mathbf{y}} \prod_{\alpha} f'_\alpha(\mathbf{y}_\alpha)
  \]

- This expression is in a form that is very similar to the standard partition function of the original problem, which we can approximate with BP.

- $Z_{(-1)}$ can also be approximated with “BP”: the factor graph remains the same, only the semantics is generalized:
  - Variables: $\mathbf{y} \in \{0, 1, *\}^n$
  - Factors: $f'_\alpha(\mathbf{y}_\alpha) = \prod_{\mathbf{x}_\alpha \in \mathbf{y}_\alpha} f_\alpha(\mathbf{x}_\alpha)$

- And we need to adapt the BP equations to cope with (-1).
BP Adaptation for (-1)

- Standard BP equations can be derived as **stationary point conditions** for continuous constrained optimization problem [Yedidia et al. '05]
  - Let \( p(\mathbf{x}) \) be the uniform distribution over solutions of a problem
  - Let \( b(\mathbf{x}) \) be a unknown parameterized distribution from a certain family
  - The goal is to **minimize** \( D_{KL}(b||p) \) over parameters of \( b(.) \)
  - Use \( b(.) \) to approximate answers about \( p(.) \)

- The BP adaptation for \( Z_{(-1)} \) follows exactly the same path, and generalizes where necessary.

One can derive a message passing algorithm for inference in factor graphs with (-1)

- We call this adaptation \( BP_{(-1)} \)
The Resulting $\text{BP}^{(-1)}$

- The $\text{BP}^{(-1)}$ iterative equations:

\[
\begin{align*}
    n_{i \rightarrow \alpha}(y_i) & \propto \prod_{\beta \ni i \setminus \alpha} m_{\beta \rightarrow i}(y_i) \\
    m_{\alpha \rightarrow i}(y_i) & \propto \sum_{y_{\alpha \setminus i} \in \{0,1,*\}^{|\alpha|-1}} f'_{\alpha}(y_{\alpha}) \prod_{j \in \alpha \setminus i} (-1)^{\delta(y_{j} = *)} n_{j \rightarrow \alpha}(y_j)
\end{align*}
\]

Relation to SP:

- For SAT: $\text{BP}^{(-1)}$ is equivalent to SP
  - The instantiation of the $\text{BP}^{(-1)}$ equations can be rewritten as SP equations
- For COL: $\text{BP}^{(-1)}$ is different from SP
  - $\text{BP}^{(-1)}$ estimates the total number of clusters
  - SP estimates the number of clusters with most frequent size
BP\(_{(-1)}\): Results for COL

Experiment: rescaling number of clusters and Z\(_{(-1)}\)

1. for 3-colorable graphs with various average degrees
2. count log(Z\(_{(-1)}\))/N and log(Z\(_{BP(-1)}\))/N

The rescaling assumes that 
\(\#\text{clusters} = \exp(N \sum(c))\)

\(\Sigma(c)\) is so called \textbf{complexity}
and is instrumental in various physics-inspired approaches
to cluster counting (will see later)

Sketch of SP results:
Nonzero between 4.42 and 4.69
Summary

- Truly combinatorial framework for cluster counting: $Z^{-1}$
  - Applicable to structured problems (contrast with original SP clusters)
  - With theoretical exactness results

- Algorithm for approximate inference over clusters: $BP^{-1}$
  - Direct derivation of SP for SAT
  - Allows derivation of new algorithms for other combinatorial problems