Improving Belief Propagation via Graphical Model Transformation

Thomas R. Halford
Communication Sciences Institute
University of Southern California

May 1, 2007

Algorithms, Inference, and Statistical Physics

Work supported in part by:
- US Army Research Office, MURI Contract DAAD19-01-1-0477
- TrellisWare Technologies, Inc.
- Powell Foundation
Improving Belief Propagation

Generalized Belief Propagation
- regions of nodes pass messages
- regions may overlap

Loop Calculus
- view BP as a truncation of a series expansion for exact inference
- improve performance by including more terms

Graphical Model Transformation
- standard model, non-standard BP
- non-standard model, standard BP
$C : (n, k) \text{ over } \mathbb{F}_q$

$C_i : (n_i, k_i) \text{ over } \mathbb{F}_q$
Graphical Model Extraction - One Code Many Models

**Code Definition**

\[ H = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0
\end{bmatrix} \]

**Graphical Models**

**Decoding Algorithms**

Many Many More!
Towards a Formalization of Extraction
Question 1: *Can the space of graphical models for a code be searched?*
Question 1: Can the space of graphical models for a code be searched?

**YES** - We’ll show how
Towards a Formalization of Extraction

Question 1: *Can the space of graphical models for a code be searched?*

  *YES* - We’ll show how

Question 2: *What are meaningful optimization constraint functions?*
Question 1: *Can the space of graphical models for a code be searched?*

*YES* - We’ll show how

Question 2: *What are meaningful optimization constraint functions?*

Inclusion in some model complexity class
Towards a Formalization of Extraction

Question 1: *Can the space of graphical models for a code be searched?*
   
   **YES** - We’ll show how

Question 2: *What are meaningful optimization constraint functions?*
   
   Inclusion in some model complexity class

Question 3: *What are meaningful optimization cost functions?*
Towards a Formalization of Extraction

Question 1: *Can the space of graphical models for a code be searched?*

YES - We’ll show how

Question 2: *What are meaningful optimization constraint functions?*

Inclusion in some model complexity class

Question 3: *What are meaningful optimization cost functions?*

Cost function $\Leftrightarrow$ good graphical model
Question 1: *Can the space of graphical models for a code be searched?*
   
   **YES** - We’ll show how

Question 2: *What are meaningful optimization constraint functions?*
   
   Inclusion in some model complexity class

Question 3: *What are meaningful optimization cost functions?*
   
   Cost function $\leftrightarrow$ *good* graphical model

   Difficult and open problem in general...

   Our Approach - *Short Cycle Structure*
Towards a Formalization of Extraction

Question 1: *Can the space of graphical models for a code be searched?*

YES - We’ll show how

Question 2: *What are meaningful optimization constraint functions?*

Inclusion in some model complexity class

Question 3: *What are meaningful optimization cost functions?*

Cost function $\Leftrightarrow$ *good* graphical model

Difficult and open problem in general...

Our Approach - *Short Cycle Structure*

Question 4: *What are good heuristics for this hard combinatorial optimization?*
Towards a Formalization of Extraction

Question 1: *Can the space of graphical models for a code be searched?*

*YES* - We’ll show how

Question 2: *What are meaningful optimization constraint functions?*

Inclusion in some model complexity class

Question 3: *What are meaningful optimization cost functions?*

Cost function $\Leftrightarrow$ *good* graphical model

Difficult and open problem in general...

Our Approach - *Short Cycle Structure*

Question 4: *What are good heuristics for this hard combinatorial optimization?*

Difficult and open problem in general...

Our Approach - *Greedy Extraction Heuristic*
Searching the Model Space: Basic Operations

- **Constraint Merging / Splitting**
- **Inserting / Removing Degree-2 Repetition Constraint**
- **Inserting / Removing Isolated Partial Parity Constraints**
- **Inserting / Removing Trivial Constraints**

References:
- [Pe88], [Fo01], [KsFrLo01]
Theorem: Let $\mathcal{G}_C$ and $\tilde{\mathcal{G}}_C$ be two graphical models for $\mathcal{C}$. Then $\mathcal{G}_C$ can be transformed into $\tilde{\mathcal{G}}_C$ via a finite number of basic operations.

Proof:
1) Maximum hidden variable alphabet size: $q^m$.

2) Each local constraint $C_i$ satisfies:

$$\min (k_i, n_i - k_i) \leq m$$

Wolf’s bound on local trellis complexity

(or is a direct product of codes which do).
Cost Functions - Short Cycle Structure

Candidate Proxies:
- stopping sets
- trapping / absorbing sets
- pseudo-codewords
- short cycles
Cost Functions - Short Cycle Structure

Candidate Proxies:

- stopping sets   ✘
- trapping / absorbing sets
- pseudo-codewords
- short cycles
Candidate Proxies:

- stopping sets ✗
- trapping / absorbing sets ✗
- pseudo-codewords
- short cycles
Cost Functions - Short Cycle Structure

Candidate Proxies:

- stopping sets  
- trapping / absorbing sets  
- pseudo-codewords  
- short cycles
Cost Functions - Short Cycle Structure

Candidate Proxies:

- stopping sets
- trapping / absorbing sets
- pseudo-codewords
- short cycles

\(\Rightarrow\) Can count cycles of length \(g\), \(g+2\) and \(g+4\) in time \(O(gn^3)\).

(Halford & Chugg, “An algorithm for counting short cycles in bipartite graphs”, IEEE Trans. IT, 52(1) 2006.)
A Greedy Heuristic for Model Extraction

**Motivation:** Tanner graphs for many block codes *necessarily* contain *many* short cycles


**Idea:** Greedily reduce cycles via model transformation

**Allowed Moves:**
1) Tanner graph search - *row operations*
2) $2^m$-ary search - *local constraint merging*

**Cost Function:** Short cycle structure $(N_4, N_6, N_8)$
Greedy Heuristic: Experimental Results

<table>
<thead>
<tr>
<th>Model</th>
<th>$N_4$</th>
<th>$N_6$</th>
<th>$N_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclic Tanner</td>
<td>7251</td>
<td>717 K</td>
<td>74 M</td>
</tr>
<tr>
<td>Improved Tanner</td>
<td>5415</td>
<td>466 K</td>
<td>43 M</td>
</tr>
<tr>
<td>$m = 2$, cyclic</td>
<td>3465</td>
<td>230 K</td>
<td>15 M</td>
</tr>
<tr>
<td>$m = 4$, cyclic</td>
<td>706</td>
<td>16 K</td>
<td>292 K</td>
</tr>
<tr>
<td>$m = 6$, cyclic</td>
<td>126</td>
<td>657</td>
<td>0</td>
</tr>
</tbody>
</table>

(63,45) BCH Code

[HaCh06b], [JiNa06]
Synthesis & Open Problems

GBP vs. Model Tx:
- similar if GBP regions don’t overlap
- model transformation allows redundancy

Loop Calculus vs. Model Tx:
- similar problem of how to transform / what loop terms to use
- model transformation improves dense models, loop calculus improves sparse models

Major Open Problems:
- better cost functions & search heuristics
- model transformation + GBP / loop calculus