Belief Propagation in Monoidal Categories

Jason Morton

Penn State

June 5, 2014
QPL 2014

Computational category theory with diagrams

- *diagram*: equivalence class of monoidal words over a finite tensor scheme, usually with certain additional properties X.
- *interpretation:* an X-monoidal functor "assigning values"

Questions of a diagram interpreted in a particular category:

1. compute a (possibly partial) contraction,
2. solve the word problem (are two diagrams equivalent, i.e. do they have the same interpretation) or compute a normal form for a diagram,
3. solve the implementability problem (construct a word equivalent to a target using a library of allowed morphisms), and
4. choose morphisms in a diagram to best approximate a more general diagram (possibly allowing the approximating diagram itself to vary).

Computational category theory is hard

- *diagram*: equivalence class of monoidal words over a finite tensor scheme, usually with certain additional properties X.
- *interpretation:* an X-monoidal functor "assigning values"

Questions of a diagram interpreted in a particular category:

1. compute a (possibly partial) contraction, (#P-hard)
2. solve the word problem (are two diagrams equivalent, i.e. do they have the same interpretation) or compute a normal form for a diagram, (undecidable)
3. solve the implementability problem (construct a word equivalent to a target using a library of allowed morphisms) (undecidable)
4. choose morphisms in a diagram to best approximate a more general diagram (possibly allowing the approximating diagram itself to vary). (NP-hard)

(Limited) practical algorithms anyway

- Many practical questions are instances of one of these problems
  - quantum programming and logic
  - probabilistic graphical models,
  - tensor network state approach to quantum condensed matter,
  - computational complexity theory: circuits, CSP, #CSP
  - even databases
- So many tractable special cases, approximate algorithms, and heuristics exist

(Limited) practical algorithms anyway

- Many practical questions are instances of one of these problems
  - quantum programming and logic
  - probabilistic graphical models,
  - tensor network state approach to quantum condensed matter,
  - computational complexity theory: circuits, CSP, #CSP
  - even databases
- So many tractable special cases, approximate algorithms, and heuristics exist
- Let's **turn these into categorical algorithms** (see also [MT13]). Formalize analogies among procedures.

Example: belief propagation

- A message-passing algorithm (Pearl 1982), for contraction, marginalization, and optimization problems
- Many extensions, analogs (survey propagation, turbo coding)
- These should be the same abstract categorical algorithm, varying the category (e.g. prob. graphical models vs. sets and relations).

Example: belief propagation

- A message-passing algorithm (Pearl 1982), for contraction, marginalization, and optimization problems
- Many extensions, analogs (survey propagation, turbo coding)
- These should be the same abstract categorical algorithm, varying the category (e.g. prob. graphical models vs. sets and relations).

To make this precise, first describe set-up

Tensor schemes and monoidal languages

Definition ([JS91, Sel09])

A (finite) *tensor scheme* $\mathscr{T}$ (or *monoidal signature*) is

- a finite set $\mathrm{Ob}_V(\mathscr{T})$ of object variables (including a monoidal identity object I),
- a finite set $\mathrm{Mor}(\mathscr{T})$ of morphism variables, and
- functions $\mathrm{dom}, \mathrm{cod} : \mathrm{Mor}(\mathscr{T}) \to \mathrm{Ob}(\mathscr{T}) = \otimes$-words in obj vars.

May also add relations.

Tensor schemes and monoidal languages

Definition ([JS91, Sel09])

A (finite) *tensor scheme* $\mathscr{T}$ (or *monoidal signature*) is

- a finite set $\mathrm{Ob}_V(\mathscr{T})$ of object variables (including a monoidal identity object I),
- a finite set $\mathrm{Mor}(\mathscr{T})$ of morphism variables, and
- functions $\mathrm{dom}, \mathrm{cod} : \mathrm{Mor}(\mathscr{T}) \to \mathrm{Ob}(\mathscr{T}) = \otimes$-words in obj vars.

May also add relations. The monoidal language $\mathscr{T}^{\otimes,\circ}$ comprises all valid morphism words built from $\mathrm{Mor}(\mathscr{T})$, and identity morphisms. It generates the free monoidal category over $\mathscr{T}$. Constructively:

1. For all $A \in \mathrm{Ob}(\mathscr{T})$, id_A is a word.
2. Each $f \in \mathrm{Mor}(\mathscr{T})$ is a word.
3. Given words u, u', $u \otimes u'$ is a word with domain $\mathrm{dom}(u) \otimes \mathrm{dom}(u')$ and codomain $\mathrm{cod}(u) \otimes \mathrm{cod}(u')$.
4. Given words w, w' with $\mathrm{dom}(w') = \mathrm{cod}(w)$, $w \circ w'$ is a word.

Words in a monoidal language

- *Interpret* a word to define a morphism is a particular category.
  - Factors through free monoidal category, which imposes equivalences such as $\mathrm{id}_A \circ f = f$ and $(f \otimes g) \circ (f' \otimes g') = (f \circ f') \otimes (g \circ g')$,
  - Words equivalent if represent same morphism in the free (X-) monoidal category.

Definition

An equivalence class of words in the free (X-) monoidal category over a tensor scheme is called a *diagram*.

- Further notions of equivalence arise with additional relations.
- So far, no normal form for words

I-valued points: the messages

- Important word problem for Belief Propagation: equality of morphisms of type $\mathrm{Mor}(I, A)$ for objects A (I-valued points).
- Why?
  - Want to generalize algorithms (e.g. belief propagation in the category of vector spaces and linear transformations)
  - Can't assume objects A are sets with points (such as probability distributions in the classical belief propagation algorithm).

I-valued points: the messages

- Important word problem for Belief Propagation: equality of morphisms of type $\mathrm{Mor}(I, A)$ for objects A (I-valued points).
- Why?
  - Want to generalize algorithms (e.g. belief propagation in the category of vector spaces and linear transformations)
  - Can't assume objects A are sets with points (such as probability distributions in the classical belief propagation algorithm).
- But, messages are still morphisms of type $\mathrm{Mor}(I, A)$ for each object A; equate these for belief propagation equations
  - Deciding if two vectors are equal up to numerical tolerance becomes deciding a **word problem** in $\mathrm{Mor}(I, A)$.
  - These messages must also be stored somehow.

Word problems in monoidal languages

- Coherent graphical languages for some types of monoidal categories means those word problems can be reduced to e.g. graph isomorphism [DK13], and produces normal forms by word$\mapsto$graph$\mapsto$word.
- Hence the word problem for the free closed category and free compact closed category over a finite tensor scheme are in LOGSPACE and P [Luk82] respectively.
- Adding adjectives (X-monoidal categories) and relations, or fixing values by applying a functor F, so that the category is no longer free may make it easier or harder.

Word problems in monoidal languages

- Coherent graphical languages for some types of monoidal categories means those word problems can be reduced to e.g. graph isomorphism [DK13], and produces normal forms by word$\mapsto$graph$\mapsto$word.
- Hence the word problem for the free closed category and free compact closed category over a finite tensor scheme are in LOGSPACE and P [Luk82] respectively.
- Adding adjectives (X-monoidal categories) and relations, or fixing values by applying a functor F, so that the category is no longer free may make it easier or harder.

Proposition

The word problem and implementability problem in a monoidal category over a finite tensor scheme are undecidable.

Word problems, term rewriting, normal forms

- Preferable to have a confluent terminating rewriting system that attached a direction to the equalities of the X-category.
- Term rewriting and computing normal forms in monoidal categories is a field in its infancy [Kis12, Mim14]
- Or, exploit completeness
  - Finite dimensional vector spaces over a field of characteristic zero are complete for traced symmetric monoidal categories [HHP08] and finite dimensional Hilbert spaces are complete for dagger compact closed categories [Sel11].

I-valued points: the messages

- Anyway for an efficient algorithm, need representation and word problem for I-valued points to be efficient.
- Classical belief propagation: have a monoid homomorphism, $\text{size}\colon \text{Ob}(\mathscr{T})^{\otimes} \to \mathbb{N}$, from the free monoid generated by the objects of our tensor scheme to the natural numbers.
- Monoidal product $\mapsto$ multiplication of vector space dimensions
- Then words in $\text{Mor}(I, A)$ can be stored and compared in $O(\text{size}(A))$.

I-valued points: the messages

- Anyway for an efficient algorithm, need representation and word problem for I-valued points to be efficient.
- Classical belief propagation: have a monoid homomorphism, $\text{size}\colon \text{Ob}(\mathscr{T})^{\otimes} \to \mathbb{N}$, from the free monoid generated by the objects of our tensor scheme to the natural numbers.
- Monoidal product $\mapsto$ multiplication of vector space dimensions
- Then words in $\text{Mor}(I, A)$ can be stored and compared in $O(\text{size}(A))$.

Now look at type of category BP will work in. Need something like variables.

Spiders: generalized variables

Definition

A *spidered category* is a strict symmetric monoidal category equipped with a special commutative (†) Frobenius structure [CPV08] $(A, m, u, \delta, \epsilon, \sigma^F)$ on each object A.

- Note: morphisms of a spidered category not generally monoid or comonoid homomorphisms.
- Now add duals for objects to obtain a compact closed category with additional structure.

Dungeon category

Call a compact closed category spidered in a compatible way a

Definition

A *dungeon category* is a compact closed category $(\mathcal{C}, \sigma^{\mathcal{C}}, i, e)$ s.t.

(i) Each object has a special commutative Frobenius structure $(A, m, u, \delta, \epsilon, \sigma^F)$ with $\sigma^F_{A^{(*)},A^{(*)}} = \sigma^{\mathcal{C}}_{A^{(*)},A^{(*)}}$, and

(ii) Any two morphisms which are

- Constructed from the identity id_A, the symmetric braiding $\sigma_{A,A}$, the Frobenius morphisms, and the dualizing cup and cap morphisms i_A, e_A for A, and
- Have the same domain (tensor product of zero or more or copies of A and A^*) and the same codomain (another such tensor product)
- Are equal.

Dungeon category

So a directed spider morphism depends only on

- the number of inward and outward directed arrows,
- which way they point,
- and their order

Good setting for generalized belief propagation because we can

- bend wires to choose inputs and outputs of any morphism and
- have spiders that play the role of variables in the probabilistic setting for belief propagation.

Sum-product and belief propagation for contraction

- The sum product algorithm [KFL01]: if a diagram is a tree, can perform contraction according to the tree.
- If not, use a tree decomposition [Hal76] to force it to be a tree, then run sum-product.
  - This is the *junction tree algorithm* [LS88], also extended to the quantum case [MS08].

Sum-product and belief propagation for contraction

- The sum product algorithm [KFL01]: if a diagram is a tree, can perform contraction according to the tree.
- If not, use a tree decomposition [Hal76] to force it to be a tree, then run sum-product.
  - This is the *junction tree algorithm* [LS88], also extended to the quantum case [MS08].
- Can improve on the abstract sum-product algorithm by using an optimized message-passing version, which among other benefits permits parallelization.

Sum-product and belief propagation for contraction

- The sum product algorithm [KFL01]: if a diagram is a tree, can perform contraction according to the tree.
- If not, use a tree decomposition [Hal76] to force it to be a tree, then run sum-product.
  - This is the *junction tree algorithm* [LS88], also extended to the quantum case [MS08].
- Can improve on the abstract sum-product algorithm by using an optimized message-passing version, which among other benefits permits parallelization.

this is belief propagation

Belief propagation in factor graphs

- The algorithm operates on a *factor graph*, a bipartite graph with
  - one part discrete random variables $v \in V$ and
  - one part factors $u \in U$.
- Each factor (potential) assigns a real number to each combination of states of the variables it is connected to.
- Multiplying factors and normalizing if needed gives a joint probability distribution.
- Belief propagation is a message passing algorithm.
  - Each message is a probability distribution over the states one variable v can take: a vector in the associated vector space V_v.
- Each factor f_U at node u is a tensor in $\bigotimes_{v \in \text{nbhd}(u)} V_v$, defines valence($u$) reshaped linear maps

$$f_{u,v} : \bigotimes_{i \in \text{nbhd}(u) \setminus v} V_i \to V_v,$$

one for each $v \in \text{nbhd}(u)$.

Messages at variables.

- Compute the pointwise (Hadamard) product of the incoming messages, and output it as the outgoing message along e.
- In a probabilistic category, Hadamard product rescales so the out message is a probability distribution.
- If there are no incoming messages, output the uniform message.

Messages at factors.

- Compute the tensor product of the incoming messages,
- apply reshaped $f_{u,v} : \bigotimes_{i \in \mathrm{nbhd}(u) \setminus v} V_i \to V_v$, and output the result as the outgoing message along the edge to v.

Resulting algorithm.

- *BP equations* describe fixed points of the update rules.
- Initial messages can be uniform distributions.
- Tree factor graph: done in two "passes," leaves to root then root to leaves, updating messages only as they change.
- Belief propagation is exact on trees

Messages at spiders.

- Apply the reshaped spider to incoming messages, and output the result as the outgoing message.
- If there are no incoming messages, treat the spider as a Frobenius unit.

Messages at “factor” morphisms.

- Compute the monoidal product of the incoming messages,
- apply the reshaped f,
- output the result as the outgoing message.

Resulting algorithm.

- System of BP equations are equalities of I-valued points describing the fixed points of the update rules.
- Initial messages can be chosen to be units at the spiders.
- Nice behavior on trees preserved

A spider is just a special kind of morphism. To get the **general bipartite** version, replace the message procedure at spiders with another copy of the factor message procedure.

To solve a problem, just reduce to category theory

- Goal: general tools that work for any category with suitable properties
  - specialize automatically by giving a monoidal category interface
- Rapidly expanding universe of applied problems given categorical interpretations
  - a problem-solving abstraction with the potential to be as useful as convex programming or numerical linear algebra.

- Bob Coecke and Ross Duncan, *Interacting quantum observables: categorical algebra and diagrammatics*, New Journal of Physics **13** (2011), no. 4, 043016.
- B. Coecke, D. Pavlovic, and J. Vicary, *A new description of orthogonal bases*, Mathematical Structures in Computer Science **13** (2008), no. 1.
- Lucas Dixon and Aleks Kissinger, *Open-graphs and monoidal theories*, Mathematical Structures in Computer Science **23** (2013), no. 02, 308–359.
- Rudolf Halin, *S-functions for graphs*, Journal of Geometry **8** (1976), no. 1-2, 171–186.
- Masahito Hasegawa, Martin Hofmann, and Gordon Plotkin, *Finite dimensional vector spaces are complete for traced symmetric monoidal categories*, Pillars of computer science, Springer, 2008, pp. 367–385.

A. Joyal and R. Street, *The geometry of tensor calculus. I*, Advances in Mathematics **88** (1991), no. 1, 55–112.

Frank R Kschischang, Brendan J Frey, and H-A Loeliger, *Factor graphs and the sum-product algorithm*, Information Theory, IEEE Transactions on **47** (2001), no. 2, 498–519.

Aleks Kissinger, *Pictures of processes: Automated graph rewriting for monoidal categories and applications to quantum computing*, arXiv preprint arXiv:1203.0202 (2012).

Steffen L Lauritzen and David J Spiegelhalter, *Local computations with probabilities on graphical structures and their application to expert systems*, Journal of the Royal Statistical Society. Series B (Methodological) (1988), 157–224.

Eugene M Luks, *Isomorphism of graphs of bounded valence can be tested in polynomial time*, Journal of Computer and System Sciences **25** (1982), no. 1, 42–65.

- Samuel Mimram, *Towards 3-dimensional rewriting theory*, http://www.pps.univ-paris-diderot.fr/~smimram/docs/mimram_3drt.pdf, 2014.

- Igor L Markov and Yaoyun Shi, *Simulating quantum computation by contracting tensor networks*, SIAM Journal on Computing **38** (2008), no. 3, 963–981.

- Jason Morton and Jacob Turner, *Generalized counting constraint satisfaction problems with determinantal circuits*, arXiv preprint arXiv:1302.1932, to appear in Linear Algebra and its Applications (2013).

- Judea Pearl, *Reverend bayes on inference engines: A distributed hierarchical approach*, AAAI, 1982, pp. 133–136.

- P. Selinger, *A survey of graphical languages for monoidal categories*, New Structures for Physics (2009), 275–337.

Peter Selinger, *Finite dimensional hilbert spaces are complete for dagger compact closed categories*, Electronic Notes in Theoretical Computer Science **270** (2011), no. 1, 113–119.