

化学構造式のためのハイパーグラフ文法

Molecular Hypergraph Grammar

梶野 洸^{*1}

Hiroshi Kajino

^{*1}日本アイ・ビー・エム株式会社東京基礎研究所

IBM Research - Tokyo

This paper is concerned with a graph grammar that can be inferred from data and always generates valence-consistent molecular graphs. Our result is that the requirement above can be satisfied by a hyperedge replacement grammar inferred from *molecular hypergraphs*, which we call a *molecular hypergraph grammar* (MHG). By substituting MHG for SMILES (a grammar generating a string representation of a molecule) in a generative model of molecules, we can generate novel molecules without decoding errors, which have been one of the common issues when using SMILES.

1. はじめに

化学分子の最適設計問題は、分子に関する所望の性質を入力したとき、それに最も近い性質を持つ分子を出力する問題である。例えば、なるべく効率の高い有機発光ダイオードを設計する問題などが考えられる [Gómez-Bombarelli 16]。この問題では、分子集合という構造が明らかではない離散集合の上で最適化を行わなければいけない点で難しいと考えられている。近年では変分自己符号化器 (VAE) を用いてこの最適化問題に取り組みやすくする研究が行われている [Gómez-Bombarelli 18]。この研究では、分子グラフとその潜在ベクトルを繋ぐ符号化器と復号化器に加えて、潜在連続空間から分子の性質を予測する予測器を学習する方法を提案している。3つめのモデルを使うことで分子の最適設計を連続空間上での最適化問題として定式化でき、復号化器を用いることで、最適化問題を解いて得られた潜在ベクトルに対応する分子グラフを得ることができる。

この研究を元に様々な方向性の拡張が行われる中で、本研究では分子の表現方法に着目する。ほとんどすべての研究では分子のグラフ表現を元に行っているが、VAE の入出力としてグラフを直接使った研究はほとんどなく、その代わりとして分子グラフを文字列表現した SMILES [Weininger 88] という表現が用いられる。その理由として、文字列を出力できるニューラルネットワークとして LSTM や GRU など様々なものが知られている一方で、グラフを出力できるニューラルネットワークがあまり一般的でないことが考えられる。

SMILES を用いた VAE の欠点として、出力された SMILES 文字列が必ずしも分子グラフにならないという問題が上げられる。これを**復号化エラー問題**と呼ぶ。この問題をフェノールを例に説明する。フェノールは SMILES では c1ccccc1O と表される。ここで数字は環の始点・終点を、括弧は枝分かれを表し、水素原子は省略される。例えば復号化器が最後の数字を出力しなかった場合、c1ccccc1 という文字列が得られるが、この文字列では環が閉じないため文字列を分子グラフに変換できない。また復号化器が複数の括弧を出力して c1ccccc1(O)ccccc1 という文字列を得た場合、炭素原子の原子価を超えてしまうため、不正な分子グラフとなる。このような問題を解決するため、出力が分子グラフとなることが保証されるような手法の開発が必要である。

本研究では、原子価を守った分子グラフを必ず生成できる**分子ハイパーグラフ文法** (*molecular hypergraph grammar*; MHG) と MHG を分子の集合から学習するアルゴリズムを提案する。MHG を文法利用型変分自己符号化器 [Kusner 17] と組合せることで、復号化エラー問題のない分子グラフの生成モデルが実現できる。MHG では、分子を**分子ハイパーグラフ**という形で表現する。通常の分子グラフとは異なり、原子をハイパーエッジ、結合をノードとして表現する。ハイパーグラフの集合からハイパーエッジ置換文法を学習するアルゴリズム [Aguinaga 16] にある前処理を加えたアルゴリズムを用いることで、分子ハイパーグラフの集合から MHG を学習できるアルゴリズムが得られる。本研究の主な貢献は、上のアルゴリズムが MHG を正しく学習できること、つまり上のアルゴリズムで得られたハイパーエッジ置換文法は、原子価を守った分子グラフを必ず生成するという点を示した点にある。

2. 導入

本章ではハイパーエッジ置換文法 (*hyperedge replacement grammar*; HRG) をデータから学習するアルゴリズム [Aguinaga 16] の紹介をする。ハイパーグラフの木分解と HRG の構文木に対応関係があることに着目したアルゴリズムで、入力したハイパーグラフのすべてを生成できる HRG が得られるのが特徴である。

2.1 ハイパーグラフと木分解

ハイパーグラフ $H = (V_H, E_H)$ は、ノード集合 V_H とハイパーエッジ集合 E_H からなる。ハイパーエッジとは V_H の非空部分集合である。ハイパーグラフの木分解 (定義 1) は、ハイパーグラフをなるべく木に近いグラフに分解したものである。図 3 は、図 2b のハイパーグラフの木分解の一つである。

定義 1. ハイパーグラフ $H = (V_H, E_H)$ の木分解は、木 $T = (V_T, E_T)$ と二つのラベル付け関数 $\ell_T^{(V)}: V_T \rightarrow 2^{V_H}$, $\ell_T^{(E)}: V_T \rightarrow 2^{E_H}$ で、以下の条件を満たすものである。

1. 各 $v_H \in V_H$ について、 $v_H \in \ell_T^{(V)}(v_T)$ を満たすノード $v_T \in V_T$ が少なくとも一つ存在する。
2. 各 $e_H \in E_H$ について、 $e_H \subseteq \ell_T^{(E)}(v_T)$ かつ $e_H \in \ell_T^{(E)}(v_T)$ を満たすノード $v_T \in V_T$ がただ一つ存在する。

連絡先: 梶野 洸, KAJINO@jp.ibm.com

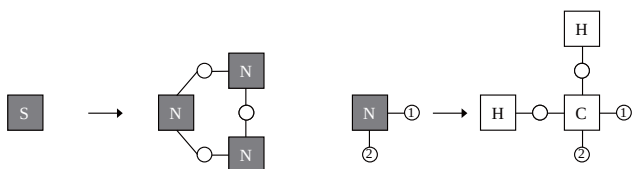


図 1: 図 2b のハイパーグラフから抽出された生成規則の一部。塗りつぶした四角は非終端記号、塗りつぶしていない四角は終端記号、丸はノード、丸と四角の間の線はハイパーエッジに含まれるノードを示している。ノードの中の番号は、取り除かれるハイパーエッジと置換されるハイパーグラフのノードの対応を示す。

3. 各 $v_H \in V_H$ について、 $\{v_T \in V_T \mid v_H \in \ell_T^{(V)}(v_T)\}$ が T 内で連結である。

ノード $v_T \in V_T$ 上のハイパーグラフを $H(v_T) := (\ell_T^{(V)}(v_T), \ell_T^{(E)}(v_T))$ と書く。

2.2 ハイパーエッジ置換文法 (HRG)

HRG は、ラベル付きハイパーエッジを持つハイパーグラフを生成する文脈自由文法である (定義 2, 図 1)。開始記号 S からなるハイパーグラフ H からはじめ、非終端記号 A でラベル付けされたハイパーエッジをハイパーグラフ R で置き換えることを繰り返し、すべてのハイパーエッジが終端記号でラベル付けされた時点で終了する。

定義 2. ハイパーエッジ置換文法 $\mathcal{G} = (N, T, S, P)$ は、以下の四つの要素で定義される。

1. N は、非終端記号の集合である。
2. T は、終端記号の集合である。
3. $S \in N$ は開始記号である。
4. P は生成規則の集合で、

- $p = (A, R)$ は生成規則、
- $A \in N$ は非終端記号、
- R は、 $|A|$ 個の外部ノードを持ち、 $T \cup N$ でラベル付けされたハイパーエッジを持つハイパーグラフである。ハイパーエッジ A の各ノードと外部ノードは対応付けられており、その対応付けに従ってハイパーエッジ A がグラフ R に置換される。

2.3 HRG 学習アルゴリズム

最後に、ハイパーグラフの集合から HRG を学習するアルゴリズム [Aguinaga 16] を紹介する。ノード $v_T \in V_T$ について、 $\text{pa}(v_T)$ を v_T の親ノード、 $\text{ch}(v_T)$ を v_T の子ノード集合とする。このアルゴリズムの基本的なアイデアは、各ペア $(v_T, \text{pa}(v_T))$ を共通するノードで繋いでいくと元のハイパーグラフを作れるということにある。例えば図 3 で各ノード上のハイパーグラフを繋いでいくと図 2b のハイパーグラフが得られる。これを HRG の言葉で言い換えると、木分解は HRG の構文木と解釈できる [Chiang 13] ということである。実際、アルゴリズムでは三つ組 $(\text{pa}(v_T), v_T, \text{ch}(v_T))$ から一つの生成規則を抽出するが、その生成規則は、「 $H(\text{pa}(v_T))$ に $H(v_T)$ を貼り付けつつ、子ノードから得られる生成規則を適用可能なように非終端記号を追加する」という操作に対応する。アルゴリズムの詳細に関しては原論文を参照されたい。

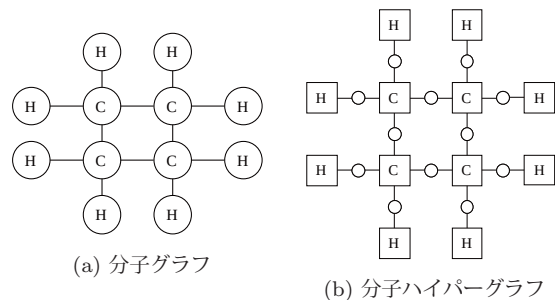


図 2: シクロブタンの二つのグラフ表現。

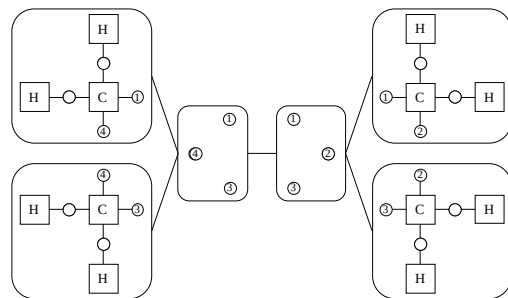


図 3: 図 2b のハイパーグラフの非冗長木分解。

3. 分子ハイパーグラフ文法

本論文の主貢献である、分子ハイパーグラフ文法 (molecular hypergraph grammar; MHG) とその学習アルゴリズムを紹介する。以下の二つのアイデアからなる。(i) 分子は通常分子グラフ (図 2a) ではなく、分子ハイパーグラフ (図 2b) で表現する。(ii) HRG の学習の際に、通常の本分解ではなく非冗長木分解を用いる。この二つのアイデアを用いることで、学習された文法は原子価を守った分子グラフを必ず生成するということが示される。

3.1 分子ハイパーグラフ

分子ハイパーグラフ (定義 3) では、原子をハイパーエッジで表現し、二つの原子間の結合を二つのハイパーエッジで共有されたノードとしてモデル化する。実際の分子との対応については、 $L^{(E)}$ は原子集合に対応し (例えば $C, H \in L^{(E)}$)、 $c^{(E)}$ は各原子の原子価に対応する (例えば $c^{(E)}(C) = 4$)。

定義 3. $L^{(E)}$ をハイパーエッジのラベル集合とし、 $c^{(E)}: L^{(E)} \rightarrow \mathbb{N}$ をノード数制約関数とする。ハイパーエッジラベル付きハイパーグラフ $H = (V_H, E_H, \ell_H^{(E)})$ が以下の条件を満たすとき分子ハイパーグラフであるという。

1. H の各頂点は必ず二つのハイパーエッジに含まれる。
2. 各 $e \in E_H$ について、 $|e| = c^{(E)}(\ell_H^{(E)}(e))$ が成り立つ。ここで $|e|$ はハイパーエッジ e に含まれるノード数を表す。

一つめの条件は、分子ハイパーグラフが通常分子グラフに変換できることを保証する。また二つめの条件が成立している場合、ハイパーグラフが原子価一致していると定義する。HRG で、常に分子ハイパーグラフを生成するものを分子ハイパーグラフ文法 (MHG) と定義する。

3.2 非冗長木分解

本論文で用いる木分解の性質である非冗長性 (定義 4) について紹介する。この性質は、HRG で生成されたハイパーグラ

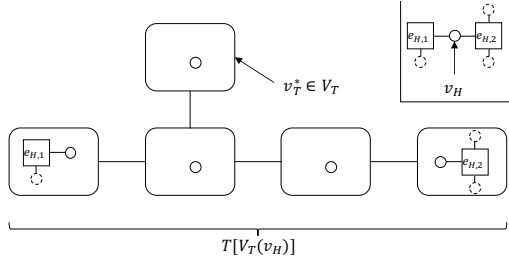


図 4: 冗長木分解 T の例. 右上の図は元のハイパーグラフ H の一部を表し, 左下にその木分解の一部 ($V_T(v_H)$ によって誘導された部分グラフ) を図示した. 非冗長にするためには v_T^* から v_H を取り除く必要がある.

フを通常のグラフに変換できることを保証するのに必要である. 直感的には, 木分解の各ノードのハイパーグラフに自明に取り除けるノードが含まれていない場合に非冗長であるという. 図 3, 図 4 はそれぞれ非冗長, 冗長な木分解を図示している.

定義 4. $H = (V_H, E_H)$ をハイパーグラフとし, $(T, \ell_T^{(V)}, \ell_T^{(E)})$ をその木分解とする. $V_T(v_H) = \{v_T \in V_T \mid v_H \in \ell_T^{(V)}(v_T)\}$ を T のノードで $v_H \in V_H$ を含むノード集合とする. 木分解が次の条件を満たすとき, 非冗長であるという.

$$\forall v_H \in V_H, \forall v_T \in V_T(v_H) \\ \ell_T^{(E)}(v_T) \neq \emptyset \Leftrightarrow v_T \text{ が } T[V_T(v_H)] \text{ の葉.} \quad (1)$$

ここで $T[V_T(v_H)]$ は $V_T(v_H)$ により誘導される部分グラフである.

任意の木分解は多項式時間で非冗長なものに変換可能である. 各 v_H について, 各 $\ell_T^{(E)}(v_T)$ ($v_T \in V_T(v_H)$) が条件 (1) を満たしていなければ v_H を取り除くことを繰り返せばよい.

3.3 MHG 学習アルゴリズム

2.3 章で議論した HRG 学習アルゴリズムと 3.2 章で議論した木分解を非冗長木分解に変換するアルゴリズムを組合せたものが MHG 学習アルゴリズムである. つまり, HRG 学習アルゴリズムで木分解から生成規則を抽出する前に木分解を非冗長木分解に変換するというものが提案アルゴリズムである.

3.4 アルゴリズムの正当性

最後に, 提案アルゴリズムの正当性に関する結果を定理 5 に示す. この定理は, 提案アルゴリズムから学習された HRG は入力分子ハイパーグラフすべてを生成でき, その HRG によって生成されたハイパーグラフは必ず分子ハイパーグラフになる (つまりこの HRG は MHG である) ということを述べている.

定理 5. $\mathcal{L}_H(L^{(E)}, c^{(E)})$ を分子ハイパーグラフ全体の集合とし, $\hat{\mathcal{L}}_H \subseteq \mathcal{L}_H(L^{(E)}, c^{(E)})$ をその部分集合とする. 提案アルゴリズムを $\hat{\mathcal{L}}_H$ に適用して学習される HRG が生成するハイパーグラフ全体の集合を $\mathcal{L}_{\text{HRG}}(\hat{\mathcal{L}}_H)$ とする. $\mathcal{L}_{\text{HRG}}(\hat{\mathcal{L}}_H)$ は次の関係を満たす.

$$\hat{\mathcal{L}}_H \subseteq \mathcal{L}_{\text{HRG}}(\hat{\mathcal{L}}_H) \subseteq \mathcal{L}_H(L^{(E)}, c^{(E)}).$$

$\hat{\mathcal{L}}_H \subseteq \mathcal{L}_{\text{HRG}}(\hat{\mathcal{L}}_H)$ に関してはすでに示されている [Aguíñaga 16] ため, 定理 5 を証明するためには $\mathcal{L}_{\text{HRG}}(\hat{\mathcal{L}}_H) \subseteq$

$\mathcal{L}_H(L^{(E)}, c^{(E)})$ が成り立つことを示せばよい. $\mathcal{L}_{\text{HRG}}(\hat{\mathcal{L}}_H) \subseteq \mathcal{L}_H(L^{(E)}, c^{(E)})$ は, 補題 7, 8, 9 を組合せることで示される. 補題 7, 8 は, 提案アルゴリズムが分子ハイパーグラフの第一条件を保存することを示しており, 補題 9 は, 提案アルゴリズムが分子ハイパーグラフの第二条件を保存することを示す.

条件 6. 各生成規則 $p = (A, R)$ について, R の外部ノードの次数は 1 で, 内部ノードの次数は 2 である.

補題 7. すべてのノードの次数が 2 のハイパーグラフ集合に対して提案アルゴリズムを適用して得られた HRG は条件 6 を満たす.

補題 8. HRG が条件 6 を満たすならば, 生成されるハイパーグラフのすべてのノードの次数は 2 である.

補題 9. 原子価一致したハイパーグラフ集合に対して提案アルゴリズムを適用して得られた HRG は常に原子価一致したハイパーグラフを生成する.

本稿では証明は省略する.

4. 関連研究

最適分子設計に関する既存研究を紹介し, その分野における本研究の貢献を議論する.

最適分子設計は特に創薬の分野において長い歴史を持つ. 数理的な手法の研究では組合せ的最適化に基づく手法を用いて所望の性質を持つ新しい分子を発見する研究が行われてきた [Jorgensen 09]. 近年の研究では, VAE を用いて分子グラフの空間と連続潜在空間との間の符号化器と復号化器を学習し, 連続空間上で分子の最適化を行った後, 復号化器を用いて分子グラフを出力するという手法が取られている [Gómez-Bombarelli 18]. 2016 年にプレプリントが公開されて以来, この枠組に則った手法が盛んに研究されている.

この手法の欠点の一つに分子の表現方法がある. 1 章で議論したように, 殆どの研究では SMILES という分子グラフの文字列表現を用いており, 最適化の結果得られた分子の連続ベクトルが正しい SMILES 文字列に復号化できないという復号化エラー問題がある. この問題に対する解決策の方針は主に二つある. 一つの方針は, 復号化器のニューラルネットワークに工夫を施しなるべく正しい SMILES 文字列を生成するようにする方法である [Kusner 17, Dai 18]. もう一つの方針は, SMILES 以外の分子の表現方法で, 出力が必ず正しい分子グラフになることが保証できるものを用いる方法である [Jin 18]. 本研究は後者に属する. 出力が必ず原子価を守った分子グラフになるようなグラフ文法とその学習方法を提案した.

我々の知る限りでは, 後者の方針を取った研究は他に一つしか知られていない [Jin 18]. この研究では, 分子グラフの木分解を分子表現の一つとして用いている. HRG と木分解の対応関係を鑑みると, この研究と我々の研究は似たような方針を取っているが, 次の二つの点で異なる. 一点目に, Jin らの研究では通常の分子グラフに対する木分解を用いているが, 我々の研究では分子ハイパーグラフに対する木分解を用いている点が異なる. 分子グラフに対する木分解から HRG を学習することは可能だが, その HRG は原子価を破る分子グラフを生成しうるため, 本研究で得られたような保証は得られない. Jin らは文法を経由しないグラフの組上げアルゴリズムを用いてこの問題を回避しているが, その分アルゴリズムやモデルが複雑になっている. 二点目に, 研究目的が異なる. Jin らの研

究ではある固定された方法で木分解を行っている一方、ニューラルネットワークの構造に多くの工夫を加えている。我々の研究では文法をデータから学習している一方、ニューラルネットワークに関する検討を行っていない。このように我々の研究と Jin らの研究は相補的であるため、我々の手法と Jin らの手法を組合せることでさらなる性能向上が期待される。

5. 結論と今後の展望

本研究では、原子価を守った分子グラフを常に生成する HRG をデータから学習するアルゴリズムを提案した。学習された HRG を文法利用型変分自己符号化器などと組合せることで、従来手法で問題となっていた復号化エラー問題を解決することができる。

今後の展望として、MDL 原理 [Jonyer 04] やベイズ規準 [Chen 95] などに基づく文法の最適化が挙げられる。定理 5 を満たす文法の中からよりコンパクトなものを選ぶことによって、後段のニューラルネットワークの学習が容易になることが期待される。

謝辞

この研究は一部、科研・基盤 (S) 15H05711 のサポートを受けている。本研究に関して議論頂いた石畠正和氏に感謝する。

参考文献

- [Aguinaga 16] Aguinaga, S., Palacios, R., Chiang, D., and Weininger, T.: Growing graphs from hyperedge replacement graph grammars, in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*, pp. 469–478 (2016)
- [Chen 95] Chen, S. F.: Bayesian Grammar Induction for Language Modeling, in *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics (ACL '95)*, pp. 228–235 (1995)
- [Chiang 13] Chiang, D., Andreas, J., Bauer, D., Hermann, K. M., Jones, B., and Knight, K.: Parsing graphs with hyperedge replacement grammars, in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL '13)*, pp. 924–932 (2013)
- [Dai 18] Dai, H., Tian, Y., Dai, B., Skiena, S., and Song, L.: Syntax-directed variational autoencoder for structured data, in *International Conference on Learning Representations (ICLR '18)* (2018)
- [Gómez-Bombarelli 16] Gómez-Bombarelli, R., Aguilera-Iparraguirre, J., Hirzel, T. D., Duvenaud, D., Maclaurin, D., Blood-Forsythe, M. A., Chae, H. S., Einzinger, M., Ha, D.-G., Wu, T., Markopoulos, G., Jeon, S., Kang, H., Miyazaki, H., Numata, M., Kim, S., Huang, W., Hong, S. I., Baldo, M., Adams, R. P., and Aspuru-Guzik, A.: Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach, *Nat Mater*, Vol. 15, No. 10, pp. 1120–1127 (2016)
- [Gómez-Bombarelli 18] Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A.: Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules, *ACS Central Science* (2018)
- [Jin 18] Jin, W., Barzilay, R., and Jaakkola, T.: Junction Tree Variational Autoencoder for Molecular Graph Generation, Technical report, arXiv:1802.04364v1 [cs.LG] (2018)
- [Jonyer 04] Jonyer, I., Holder, L. B., and Cook, D. J.: MDL-based context-free graph grammar induction and applications, *International Journal on Artificial Intelligence Tools*, Vol. 13, No. 1, pp. 65–79 (2004)
- [Jorgensen 09] Jorgensen, W. L.: Efficient Drug Lead Discovery and Optimization, *Accounts of chemical research*, Vol. 42, No. 6, pp. 724–733 (2009)
- [Kusner 17] Kusner, M. J., Paige, B., and Hernández-Lobato, J. M.: Grammar variational autoencoder, in *Proceedings of the 35th International Conference on Machine Learning (ICML '17)* (2017)
- [Weininger 88] Weininger, D.: SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, *Journal of Chemical Information and Computer Sciences*, Vol. 28, No. 1, pp. 31–36 (1988)