

# 知識グラフ埋め込みのための二値化 CP 分解

岸本広輝<sup>\*a</sup> 林克彦<sup>a</sup> 赤井元紀<sup>a</sup> 新保仁<sup>b</sup> 駒谷和範<sup>a</sup>

<sup>a</sup> 大阪大学 産業科学研究所

<sup>b</sup> 奈良先端科学技術大学院大学 情報科学研究科

<sup>\*</sup>kishimoto@ei.sanken.osaka-u.ac.jp

## 1 はじめに

Freebase [2] のような巨大な知識グラフが構築されて以降、知識グラフ上で関係予測 (知識グラフ補完) を行う手法が活発に研究されてきた。特に、知識グラフ補完を潜在的な特徴空間で行う知識グラフ埋め込みは、推論規則や特徴設計を必要とせず、高い予測精度を実現できるため、関係抽出 [6] や推薦システム [5] などへの応用が急速に進められている。

知識グラフ  $G = (\mathcal{E}, \mathcal{R}, \mathcal{F})$  はエンティティの集合  $\mathcal{E} = \{e_1, e_2, \dots, e_{N_e}\}$ 、関係の集合  $\mathcal{R} = \{r_1, r_2, \dots, r_{N_r}\}$ 、事実  $(e_i, e_j, r_k)$  の集合  $\mathcal{F} \subseteq \mathcal{E} \times \mathcal{E} \times \mathcal{R}$  から成る ( $e_i, e_j \in \mathcal{E}$  は主語と目的語,  $r_k \in \mathcal{R}$  はそれらの関係)。知識グラフは各要素  $x_{ijk}$  が  $(e_i, e_j, r_k) \in \mathcal{F}$  の場合、1、それ以外は 0 をとる三次隣接テンソル  $\mathcal{X} \in \{0, 1\}^{N_e \times N_e \times N_r}$  で表すことができる。知識グラフ埋め込みでは、エンティティと関係をベクトル空間へと埋め込み、その間の演算を使って、事実の信頼度スコア関数  $\theta_{ijk}$  をモデル化する。

これまで様々な知識グラフ埋め込みモデルが提案されたが、表 1 で示すように計算量や表現力の問題が残されてきた<sup>1</sup>。しかし、昨年発表された研究 [4] では古典的なテンソル分解法の 1 つである CANDECOMP/PARAFAC (CP) 分解でも完全な表現力を持ち、高い予測精度を達成できることが報告されている。CP 分解では主語、目的語それぞれに対するエンティティを  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N_e \times D}$ 、関係を  $\mathbf{C} \in \mathbb{R}^{N_r \times D}$  へと埋め込み ( $D$  は潜在空間の次元数)、信頼度スコア  $\theta_{ijk}$  を  $\langle \mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_k \rangle$  として計算する (記法は 2.1 節で述べる)。CP 分解は  $32D(2N_e + N_r)$  のメモリ消費量、 $O(D)$  の計算量であり、簡潔なモデルとなる。

一方、エンティティ数  $N_e$  や次元数  $D$  が大きな場合、CP 分解モデルでも実運用時のメモリ・計算コストは依然大きな問題として残る<sup>2</sup>。本稿ではこの問題を解

表 1: モデルの表現力: 任意の三次隣接テンソル  $\mathcal{X} \in \{0, 1\}^{N_e \times N_e \times N_r}$  は表記の次元数で再現可能。\* 二値化 CP では論理演算によって高速にスコア計算が行える。

モデル	計算量	スコア関数	表現力
TransE 等	$O(D) - O(KD)$	$-\ p(\mathbf{a}_i) + \mathbf{b}_k - p(\mathbf{a}_j)\ $	$\mathcal{X}$
RESCAL	$O(D^2)$	$\mathbf{a}_i^T \mathbf{B} \mathbf{a}_j$	$N_e$
DistMult	$O(D)$	$\langle \mathbf{a}_i, \mathbf{a}_j, \mathbf{b}_k \rangle$	$\mathcal{X}$
CP	$O(D)$	$\langle \mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_k \rangle$	$N_e N_r$
二値化 CP	$O(D)^*$	$h(\bar{\mathbf{a}}_i, \text{XNOR}(\bar{\mathbf{b}}_j, \bar{\mathbf{c}}_k))$	$8N_e N_r$

決するため、バイナリベクトルによる埋め込み (二値化) を考える。同じ次元数の実数ベクトルと比較して、バイナリベクトルはメモリコストを 1/32 に削減でき、スコア計算をハミング距離によって高速に実行できる。

提案手法 (二値化 CP 分解) では CP 分解モデルのパラメータを量子化しながら、確率的勾配降下法 (SGD) に基づいてロジスティック回帰学習することで二値化を実現する。量子化の導入により目的関数の勾配がほとんど 0 となるため、Hinton's Straight Through Estimator [1] (ここでは Hinton 法と呼ぶ) に基づいて勾配計算時には量子化を考慮せずにパラメータ更新を行うことで対処した。本稿の貢献を以下にまとめる。

- 量子化に基づく二値化 CP 分解を提案し、知識グラフ埋め込みが高精度に行えることを実験的に示した。
- 二値化 CP 分解が任意の 3 次隣接テンソルを表現できることを証明した。

## 2 CP 分解による埋め込み

### 2.1 記法の定義

ベクトル, 行列, テンソルをそれぞれ太字かつ小文字 ( $\mathbf{a}$ ), 太字かつ大文字 ( $\mathbf{A}$ ), 太字かつカリグラフィック体 ( $\mathcal{X}$ ) で表す。行列  $\mathbf{A}$  の  $i$  番目の行ベクトルを  $\mathbf{a}_i$ 、 $j$  番目の列ベクトルを  $\mathbf{a}_j$  または  $\mathbf{a}_j$  と表す。ベクトル, 行列, テンソルの各要素を  $a_i, a_{ij}, x_{ijk}$  と表す。

<sup>1</sup>TransE や TransH などでは主語と目的語に写像  $p: \mathbb{R}^D \rightarrow \mathbb{R}^K$  をかけた後、平行移動と距離によりスコアを計算する。しかし、このようなモデルでは  $x_{iik} = 1$  の場合、 $x_{jjk} = 0$  が表現できないため、任意の三次隣接テンソル  $\mathcal{X}$  を表せない。

<sup>2</sup>関係抽出 [6] や推薦システム [5] では学習時よりも実運用時のメモリ・計算コスト削減が重要である。

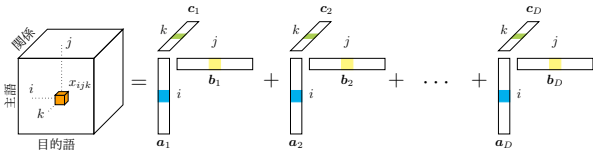


図 1: 三次隣接テンソルの CP 分解.

記号  $\circ$  はアダマール積,  $\otimes$  は外積を表す. 3つの (行または列) ベクトル  $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^D$  が与えられたとき, 三重内積を  $\langle \mathbf{a}, \mathbf{b}, \mathbf{c} \rangle = \sum_{d \in [D]} a_d b_d c_d$  として定義する.  $[D]$  は  $1, 2, \dots, D$  の自然数の集合を表す.

三次テンソル  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$  が 3つのベクトルの外積  $\mathcal{X} = \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}$  で表せるとき, これをランク 1 テンソルと呼ぶ. このとき,  $\mathcal{X}$  の要素  $x_{i_1 i_2 i_3}$  は

$$x_{i_1 i_2 i_3} = a_{i_1} b_{i_2} c_{i_3} \text{ for } i_1 \in [I_1], i_2 \in [I_2], i_3 \in [I_3]$$

となる. また, テンソル  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_k}$  のノルムは

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1 \in [I_1]} \sum_{i_2 \in [I_2]} \dots \sum_{i_k \in [I_k]} x_{i_1 i_2 \dots i_k}^2}$$

として定義される.

## 2.2 ロジスティック CP 分解

CP 分解はテンソルを  $D$  個のランク 1 テンソルの線形和によって分解する. ある知識グラフを表した三次隣接テンソル  $\mathcal{X} \in \mathbb{R}^{N_e \times N_e \times N_r}$  に対する CP 分解は

$$\mathcal{X} \approx \sum_{d \in [D]} \mathbf{a}_d \otimes \mathbf{b}_d \otimes \mathbf{c}_d$$

として表せる. ここで,  $\mathbf{a}_d \in \mathbb{R}^{N_e}, \mathbf{b}_d \in \mathbb{R}^{N_e}, \mathbf{c}_d \in \mathbb{R}^{N_r}$  とする. 図 1 はその CP 分解を表したものである. 行列  $\mathbf{A} = [\mathbf{a}_1 \mathbf{a}_2 \dots \mathbf{a}_D]$  は因子行列とし,  $\mathbf{B}, \mathbf{C}$  も同様に定義する.  $\mathcal{X}$  の要素  $x_{ijk}$  は  $\langle \mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_k \rangle$  で近似される.  $\mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_k$  はそれぞれ主語, 目的語, 関係に対する  $D$  次元ベクトルを表す.

本稿では CP 分解を二値類問題として考え, ロジスティック回帰による定式化を行う. まず, 各  $x_{ijk}$  が因子行列  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  から独立に生起すると仮定し,

$$p(\mathcal{X} | \mathbf{A}, \mathbf{B}, \mathbf{C}) = \prod_{i \in [N_e]} \prod_{j \in [N_e]} \prod_{k \in [N_r]} p(x_{ijk} | \theta_{ijk})$$

を考える. ある事実  $(e_i, e_j, r_k) \in \mathcal{F}$  に対するスコア関数は  $\theta_{ijk} = \langle \mathbf{a}_i, \mathbf{b}_j, \mathbf{c}_k \rangle$  として定義される. また, 確率変数  $x_{ijk}$  が  $x_{ijk} \sim \text{Bernoulli}(\sigma(\theta_{ijk}))$  と仮定すると, 事後確率は以下のように定義される.

$$p(x_{ijk} | \theta_{ijk}) = \begin{cases} \sigma(\theta_{ijk}) & \text{if } x_{ijk} = 1, \\ 1 - \sigma(\theta_{ijk}) & \text{if } x_{ijk} = 0. \end{cases}$$

$\sigma(x) = 1/(1 + \exp(-x))$  でシグモイド関数を表す.

ここで, 最大事後確率の負の対数尤度を最小化することで学習を行う. このとき最適化する目的関数は

$$E = \sum_{i \in [N_e]} \sum_{j \in [N_e]} \sum_{k \in [N_r]} E_{ijk}$$

$$E_{ijk} = \underbrace{-x_{ijk} \log \sigma(\theta_{ijk}) + (x_{ijk} - 1) \log(1 - \sigma(\theta_{ijk}))}_{\text{ロジスティック損失 } \ell_{ijk}} + \underbrace{\lambda_A \|\mathbf{a}_i\|^2 + \lambda_B \|\mathbf{b}_j\|^2 + \lambda_C \|\mathbf{c}_k\|^2}_{\text{L2 正則化}}$$

である. 実験では SGD と負例サンプリングにより目的関数を最小化した.  $E_{ijk}$  の  $\mathbf{a}_i$  に関する偏微分は

$$\frac{\partial E_{ijk}}{\partial \mathbf{a}_i} = -x_{ijk} \exp(-\theta_{ijk}) \sigma(\theta_{ijk}) \mathbf{b}_j \circ \mathbf{c}_k + (1 - x_{ijk}) \sigma(\theta_{ijk}) \mathbf{b}_j \circ \mathbf{c}_k + 2\lambda_A \mathbf{a}_i$$

となる.  $\mathbf{b}_j, \mathbf{c}_k$  も同様である.

## 3 提案手法

### 3.1 二値化 CP 分解

提案手法 (二値化 CP 分解) では CP 分解における因子行列  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  の要素を二値に変換する. まず, スコア関数を  $\theta_{ijk}^{(b)} = \langle \mathbf{a}_i^{(b)}, \mathbf{b}_j^{(b)}, \mathbf{c}_k^{(b)} \rangle$  として再定義する. ここで  $a_{id}^{(b)}, b_{jd}^{(b)}, c_{kd}^{(b)}$  は以下で定義される量子化関数によって,  $a_{id}, b_{jd}, c_{kd}$  を二値化した値である.

$$x^{(b)} = Q_\Delta(x) = \begin{cases} \Delta & \text{if } x \geq 0, \\ -\Delta & \text{if } x < 0. \end{cases}$$

ただし,  $\Delta$  は正の実数である. 量子化関数は  $\mathbf{x}^{(b)} = Q_\Delta(\mathbf{x})$  ( $x_i^{(b)} = Q_\Delta(x_i)$ ) としてベクトルに拡張できる.

2.2 節の損失関数を以下のように再定義する.

$$\ell_{ijk}^{(b)} = -x_{ijk} \log \sigma(\theta_{ijk}^{(b)}) + (x_{ijk} - 1) \log(1 - \sigma(\theta_{ijk}^{(b)})).$$

これに基づいて学習を行う際, 量子化関数  $Q_\Delta$  に関する勾配の値のほとんどが 0 となり, パラメータの更新がうまく行えない. 本稿ではこの問題を解決するため, 量子化関数の勾配を以下のように定義する.

$$\frac{\partial Q_\Delta(\mathbf{x})}{\partial \mathbf{x}} \approx \mathbf{I}.$$

これは深層学習の量子化によく利用される Hinton 法である [1]. このとき損失関数の  $\mathbf{a}_i$  に関する偏微分は

$$\frac{\partial \ell_{ijk}^{(b)}}{\partial \mathbf{a}_i} = \frac{\partial Q_\Delta(\mathbf{a}_i)}{\partial \mathbf{a}_i} \frac{\partial \ell_{ijk}^{(b)}}{\partial Q_\Delta(\mathbf{a}_i)} \approx \mathbf{I} \frac{\partial \ell_{ijk}^{(b)}}{\partial Q_\Delta(\mathbf{a}_i)} = \frac{\partial \ell_{ijk}^{(b)}}{\partial \mathbf{a}_i^{(b)}}$$

となり,  $E_{ijk}$  の偏微分を以下のように表せる.

$$\frac{\partial E_{ijk}}{\partial \mathbf{a}_i} = -x_{ijk} \exp\left(-\theta_{ijk}^{(b)}\right) \sigma(\theta_{ijk}^{(b)}) \mathbf{b}_j^{(b)} \circ \mathbf{c}_k^{(b)} \\ + (1 - x_{ijk}) \sigma(\theta_{ijk}^{(b)}) \mathbf{b}_j^{(b)} \circ \mathbf{c}_k^{(b)} + 2\lambda_A \mathbf{a}_i.$$

$\mathbf{b}_j, \mathbf{c}_k$ : も同様である.

また, 詳細は省略するが, バイナリベクトルによるスコア計算は論理演算で高速に実行できる.

$$\theta_{ijk}^{(b)} = \Delta^3 \{-2h(\bar{\mathbf{a}}_i^{(b)}, \text{XNOR}(\bar{\mathbf{b}}_j^{(b)}, \bar{\mathbf{c}}_k^{(b)})) + D\}$$

ここで  $h(\cdot, \cdot)$  はハミング距離,  $\text{XNOR}(\cdot, \cdot)$  は排他的論理和の否定,  $\bar{\mathbf{a}}_i^{(b)}$  は  $a_{id}^{(b)} = \Delta$  のとき真, それ以外, 偽とするベクトルを表す.  $\bar{\mathbf{b}}_j^{(b)}, \bar{\mathbf{c}}_k^{(b)}$  も同様である.

### 3.2 二値化 CP 分解の表現力

CP 分解モデルでは次元数  $D \geq N_e N_r$  であれば任意の三次隣接テンソルを表現できる [4]. ここでは二値化 CP 分解モデルの表現力について考える. ただし, 紙面の都合上, 最終的な結果のみを示す.

**命題** 二値因子行列  $\mathbf{A}^{(b)}, \mathbf{B}^{(b)} \in \{-\Delta, \Delta\}^{N_e \times 8N_e N_r}$ ,  $\mathbf{C}^{(b)} \in \{-\Delta, \Delta\}^{N_r \times 8N_e N_r}$  による二値化 CP 分解は任意の三次隣接テンソル  $\mathcal{X} \in \{0, 1\}^{N_e \times N_e \times N_r}$  を表現可能である.

**証明**  $i, j, k, h, m, n, o$  をそれぞれ  $h \leq 2N_e N_r$ ,  $m \leq 2N_r$ ,  $k, n \leq N_r$ ,  $i, j, o \leq N_e$  を満たす自然数とする. ある行列  $\mathbf{Y} \in \{-\Delta, \Delta\}^{M \times 8N_e N_r}$  ( $M$  は任意の自然数) における  $\ell$  行目の  $4h - 3$  から  $4h$  までの要素を  $\mathbf{y}_{\ell h} = [y_{\ell, 4h-3}, y_{\ell, 4h-2}, y_{\ell, 4h-1}, y_{\ell, 4h}]$  と表記する.

ここで  $e = i + 2N_e(k - 1)$ ,  $f = i + N_e + 2N_e(k - 1)$ ,  $\mathbf{p} = [\Delta, \Delta, -\Delta, -\Delta]$ ,  $\mathbf{q} = [\Delta, -\Delta, \Delta, -\Delta]$ ,  $\mathbf{r} = [\Delta, \Delta, \Delta, \Delta]$ ,  $\mathbf{s} = [-\Delta, -\Delta, -\Delta, -\Delta]$  とおく. そして, 上で導入した表記を用いて  $\mathbf{A}^{(b)}, \mathbf{B}^{(b)}, \mathbf{C}^{(b)}$  の部分行列 (ベクトル) を

$$\mathbf{a}_{ih}^{(b)} = \begin{cases} \mathbf{p} & \text{if } h = N_e(m - 1) + i \\ \mathbf{q} & \text{otherwise} \end{cases}$$

$$\mathbf{b}_{je}^{(b)}, \mathbf{b}_{jf}^{(b)} = \begin{cases} \mathbf{p} & \text{if } x_{ijk} = 1 \\ \mathbf{r} & \text{otherwise} \end{cases}$$

$$\mathbf{c}_{kh}^{(b)} = \begin{cases} \mathbf{s} & \text{if } h = 2N_e(n - 1) + N_e + o \text{ and } n \neq k \\ \mathbf{r} & \text{otherwise} \end{cases}$$

のように定義すると,

$$\mathcal{X} = \text{sign}\left(\sum_{d \in [8N_e N_r]} \mathbf{a}_d^{(b)} \otimes \mathbf{b}_d^{(b)} \otimes \mathbf{c}_d^{(b)}\right)$$

となり, 命題は示される.  $\square$

表 2: 知識グラフ補完精度の評価用データセット.

	WN18RR	FB15k-237
$N_e$	40,559	14,505
$N_r$	11	237
訓練サンプル数	86,835	272,115
開発サンプル数	3,034	17,535
テストサンプル数	3,134	20,466

## 4 実験

### 4.1 データセット・評価方法・実験設定

知識グラフ補完の精度を評価するため, 標準のデータセット WN18RR, FB15k-237 を用いた (表 2).

テストデータ中の事実  $(e_i, e_j, r_k)$  における  $e_i$  (または  $e_j$ ) を  $e_\ell \in \mathcal{E}$  に置き換え, スコアで降順にソートして, 元の事実の順位を平均逆順位 (MRR) と Hits@10 で評価した. 複数の正解事実が存在する場合, 対象とする事実以外を除いて評価した.

CP 分解のハイパーパラメータはグリッド探索により設定した. 探索範囲は  $\lambda_A, \lambda_B, \lambda_C \in \{0.0001, 0\}$ ,  $\eta \in \{0.025, 0.05\}$ ,  $D \in \{15, 50, 100, 200, 300, 400, 500\}$  で MRR を開発データ上で最大化するパラメータを求めた. 二値化 CP 分解ではパラメータの探索範囲を  $\lambda_A, \lambda_B, \lambda_C \in \{0.0001, 0\}$ ,  $\eta \in \{0.025, 0.05\}$ ,  $D \in \{100, 200, 300, 400, 500\}$  とした.  $\Delta$  は 1 に固定した.  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  は両モデルとも一様分布  $U[-\frac{\sqrt{6}}{2D}, \frac{\sqrt{6}}{2D}]$  に従う乱数によって初期化し, 負例サンプル数は 1 つの正例につき, WN18RR で 5, FB15k-237 で 10 に設定した.

### 4.2 実験結果

ここでは二値化 CP 分解を CP 分解, 及び, 他の有力な知識グラフ埋め込みモデルと比較した. 公平性を保つため, 学習後の CP 分解モデルをベクトル量子化する方法 (VQ-CP) も比較した. ベクトル量子化は実行列  $\mathbf{X} \in \mathbb{R}^{D_1 \times D_2}$  に対して,

$$\hat{\mathbf{X}}^{(b)}, \hat{\alpha} = \underset{\mathbf{X}^{(b)}, \alpha}{\text{argmin}} \|\mathbf{X} - \alpha \mathbf{X}^{(b)}\|^2$$

を最適化して二値化を行う. ここで  $\mathbf{X}^{(b)} \in \{-1, 1\}^{D_1 \times D_2}$  を二値行列,  $\alpha$  を正の実数とする. この問題の解  $\hat{\mathbf{X}}, \hat{\alpha}$  はそれぞれ  $\text{sign}(\mathbf{X})$ ,  $\frac{1}{D_1 \times D_2} \|\mathbf{X}\|_1$  で与えられる. このとき  $\|\cdot\|_1$  は  $l_1$  ノルムである.

表 3 に示すように, ほとんどの評価指標で二値化 CP ( $D = 400$ ) は精度を落とすことなく, モデルサイズの削減に成功した. 従来法と比較して, 二値化 CP, VQ-CP 共にモデルサイズは 1/10 から 1/20 になるが, VQ-CP は精度が大きく低下しており, 提案法の有効性は明らかである. ただし, 図 2 の左図で示すように,

表 3: 知識グラフ補完精度とモデルサイズ (WN18RR):

\* は文献 [3] から抜粋.

モデル ( $D$ )	モデルサイズ	WN18RR		FB15k-237	
		MRR	Hits@10	MRR	Hits@10
DistMult* (200)	32MB	43.0	49.0	24.1	41.9
ComplEx* (200)	64MB	44.0	51.0	24.7	42.8
ConvE* (200)	32MB	43.0	<b>52.0</b>	<b>32.5</b>	<b>50.1</b>
CP (15)	5MB	40.0	47.0	22.0	40.5
CP (50)	16MB	43.0	49.0	24.8	43.0
CP (200)	64MB	44.0	51.0	29.0	47.9
CP (500)	160MB	43.0	51.0	29.2	48.4
VQ-CP (200)	2MB	36.0	40.0	8.7	15.1
VQ-CP (500)	5MB	36.0	40.0	8.3	14.6
二値化 CP (100)	1MB	38.0	44.0	23.2	42.0
二値化 CP (200)	2MB	45.0	52.0	27.8	44.6
二値化 CP (400)	4MB	<b>45.0</b>	<b>52.0</b>	29.2	46.1
二値化 CP (500)	5MB	<b>45.0</b>	<b>52.0</b>	29.1	46.1

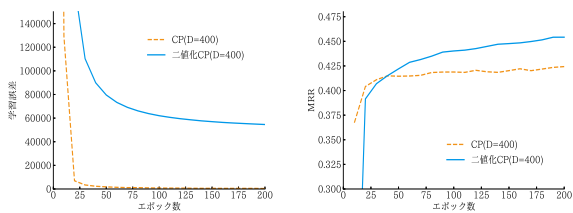


図 2: WN18RR でのエポック数 vs 学習誤差, MRR.

提案法は学習時の誤差を完全に減少させることができていない。一方、右図のように補完精度は向上しており、このデータ上では量子化が過学習を緩和させるように働いたと考えられるが、今後も他のデータを使って検証を進める必要がある。

3.1 節で述べたように二値化 CP は論理演算を用いることで高速なスコア計算を実現する。CP (Float), 二値化 CP (ハミング距離) の計算速度を比較するために次元数を  $D = 10$  から 1000 まで 10 刻みで変化させて、スコア関数の計算を各次元で 100,000 回行った。図 3 より通常の内積演算と比較して、論理演算で高速なスコア計算を実現できていることがわかる。

### 4.3 大規模データでの実験結果

より大きなデータで提案法の有効性を検証するため、Freebase-Music データ (<https://datalab.snu.ac.kr/haten2/>) から頻度が 10 より少ないエンティティと関係を排除し、残った 18,482,832 個の事実 ( $N_e = 3,025,684$ ,  $N_r = 138$ ) を実験に用いた。これを 18,462,832 個, 10,000 個, 10,000 個に分割し、訓練、開発、テストデータとし、事実の真偽に関する二値分類精度を検証した。負例としてランダム抽出した未知事実 20,000 個を半分にしてテスト、開発データに混ぜた。ハイパーパラメータは開発データで分類精度を最大にするものを 4.1 節の設定から探索した。表 4 から、二値化 CP が高い分類精度を保ちながら大幅なメ

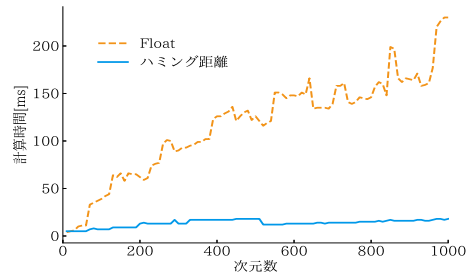


図 3: スコア計算時間の比較.

表 4: Freebase-Music での二値分類精度.

モデル ( $D$ )	モデルサイズ	分類精度
CP (15)	0.4GB	50.3
CP (200)	4.8GB	89.2
二値化 CP (400)	<b>0.3GB</b>	<b>92.8</b>

モリ削減に成功したことがわかる。これより提案法は大規模なデータでも有効と考えられる。

## 5 まとめ

本稿では二値化 CP 分解モデルを提案し、補完精度を落とさずに、メモリ・計算コスト削減を達成した。今後は提案法を関係抽出やアイテム推薦に応用することで、実用面での利点をより深く調査したい。

## 参考文献

- [1] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.
- [2] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.
- [3] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1811–1818, 2018.
- [4] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Proc. of the 32nd Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 4289–4300, 2018.
- [5] Enrico Palumbo, Giuseppe Rizzo, Raphaël Troncy, Elena Baralis, Michele Osella, and Enrico Ferro. An empirical comparison of knowledge graph embeddings for item recommendation. In *Proc. of the 1st Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies (DL4KGS)*, pages 14–20, 2018.
- [6] Jason Weston, Antoine Bordes, Oksana Yakhnenko, and Nicolas Usunier. Connecting language and knowledge bases with embedding models for relation extraction. In *Proc. of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1366–1371, 2013.