知識グラフ埋め込み学習における損失関数の統一的解釈

上垣外 英剛 *
東京工業大学
kamigaito@lr.pi.titech.ac.jp

林 克彦 *
群馬大学
khayashi0201@gmail.com

1 はじめに

知識グラフとはエンティティ間の関係を記述したグラフであり，対話や質問応答などに利用される．しかし，知識グラフを作成するには，膨大なエンティティの組み合わせとその関係を考慮する必要があり，人手・半自動で完全なグラフを構築することは困難である．そのため，自動でエンティティ間のリンク予測を行うことは重要な課題である．

現在，エンティティ間のリンク予測は主に知識グラフの埋め込み表現に基づいたスコアリング法 [1] を用いて行われている．この方法では，各リンクに対するスコアをエンティティ及び関係の埋め込み表現によって計算するが，埋め込み表現の学習は様々な損失関数を用いて行われている．特に，表現学習において一般的に使用されていることもあり，**ソフトマックス関数に対する交差エントロピー（SCE; Softmax Cross Entropy）**を用いる方法 [2] や，その近似法の一つである**負例サンプリング（NS; Negative Sampling）**による方法 [3] が主流である．

近年，これらの損失関数と既存のスコアリング方法との組み合わせを適切に選択することにより，リンク予測の性能が大きく変化することが文献 [4] で経験的に示されている．その一方で，SCE 損失と NS 損失に対する理論的な関係性はあまり探求されていない．そのため，これらの異なる損失関数を用いた際の結果を比較することが公平であるのか否か，またどのような条件であればそれが可能であるか，などの議論を困難にしている．さらに，スコアリング手法と損失関数の関係が明らかでないことは，予測精度を向上させる上で様々な組み合わせを試行して，経験的に良好な組み合わせを導く必要があり，ハイパーパラメータを探索するための計算量を膨大なものとしている．

本稿では，これらの問題を解決するために，**Bregman 距離（Bregman Divergence）**[5] を用いて SCE 損失と NS 損失の統一的な解釈を試みた．この解釈の下で，どのような条件において両損失関数での最適解が同一になり得るかを理論的に確認し，両関数の背後に存在する距離の違いが学習結果にどのような影響を及ぼすかを実験的に検証した．

2 Bregman 距離と SCE 損失

知識グラフにおけるエンティティ e_i と e_j の関係 r_k によるリンクを (e_i, r_k, e_j) と表記する．与えられたクエリ $(e_i, r_k, ?)$ や $(?, r_k, e_j)$ に対して，リンク予測モデルは ? に対応するエンティティを予測する．このようなクエリを入力 x，予測するべきエンティティを y とするとき，モデルパラメータ θ に基づくスコア関数 $f_\theta(x, y)$ の下，x から y が予測される確率 $p_\theta(y|x)$ は，ソフトマックス関数を用いて次のように定義される:

$$p_\theta(y|x) = \frac{\exp\left(f_\theta(x,y)\right)}{\sum_{y' \in Y} \exp\left(f_\theta(x,y')\right)}. \quad (1)$$

Y は予測候補となる全エンティティを表す．

次に Bregman 距離についての説明を行う．入力 x とその出力ラベル y のペアを (x, y) として書く．観測データを $D = \{(x_1, y_1), \cdots, (x_{|D|}, y_{|D|})\}$ とし，これは分布 $p_d(x, y)$ に従うとする．$\Psi(z)$ を微分可能な関数とすると，分布 f と g の間の Bregman 距離は以下のように定義される:

$$d_{\Psi(z)}(f, g) = \Psi(f) - \Psi(g) - \Delta\Psi(g)^T (f - g). \quad (2)$$

$\Psi(z)$ を変えることによって，様々な距離を表現することが可能となる．本稿では，文献 [6] と同様に，観測データ全体における距離の最小化を考えるため，f を固定した上で，$d_\Psi(f, g)$ の期待値を

$$B_{\Psi(z)}(f, g) = \sum_{(x,y) \in D} \left[-\Psi(g) + \Delta\Psi(g)^T g - \Delta\Psi(g)^T f\right] p_d(x, y) \quad (3)$$

として定義する．文献 [6] より，$\Psi(z)$ が厳密に凸で微分可能なとき[1] に $B_{\Psi(z)}(f, g) = 0$ が満たされてい

* 共同責任著者

1） なお，本稿で扱う $\Psi(z)$ は全てこの性質を満たす．

れば，f と g は等価である．本稿では，分布と損失関数の間における関係性を調べるため，$B_{\Psi(z)}$ の最小化を考える．

後述する NS との比較のために，式 (3) を用いて，我々はまず SCE 損失関数の導出を行う．式 (3) において f に $p_d(\mathbf{y}|x)$ を[2)]，g に $p_\theta(\mathbf{y}|x)$ を入力し，ベクトルの次元数を表す関数 len を用いて，$\Psi(\mathbf{z}) = \sum_{i=1}^{len(\mathbf{z})} z_i \log z_i$ とした際に，SCE 損失関数は次のように導出される．

$$B_{\Psi(\mathbf{z})}(p_d(\mathbf{y}|x), p_\theta(\mathbf{y}|x)) \tag{4}$$

$$= -\sum_{(x,y)\in D}\left[\sum_{i=1}^{|Y|} p_d(y_i|x)\log p_\theta(y_i|x)\right] p_d(x,y) \tag{5}$$

$$= -\frac{1}{|D|}\sum_{(x,y)\in D}\log p_\theta(y|x). \tag{6}$$

この導出から $B_{\Psi(z)}(p_d(\mathbf{y}|x), p_\theta(\mathbf{y}|x))$ が 0 となる最小化を通じて $p_\theta(y|x)$ が $p_d(y|x)$ に等しくなることが分かる．本稿では以降，このように式 (3) が 0 となる際の $p_\theta(y|x)$ を**目的分布**と呼ぶ．

3 NS の解釈、SCE との関係性

Bregman 距離を使って NS の性質について議論する．分布 $p_d(x,y)$ に従う観測データ D の各サンプル $(x,y) \in D$ に対して，NS では既知の雑音分布 p_n から ν 個の雑音サンプルを抽出し，分布 $G(y|x;\theta) = \exp(-f_\theta(x,y))$ に対するモデルパラメータ θ の推定を考える．まず，(x,y) が観測データから抽出されたサンプルであれば，二値クラスラベル $C=1$ とし，雑音分布 p_n から抽出されたサンプルであれば，$C=0$ とすると，クラスラベル C に対する事後確率は以下のように定義できる．

$$p(C=1,y|x;\theta) = \frac{1}{1+\exp(-f_\theta(x,y))} = \frac{1}{1+G(y|x;\theta)},$$
$$p(C=0,y|x;\theta) = 1-p(C=1,y|x;\theta) = \frac{G(y|x;\theta)}{1+G(y|x;\theta)}.$$

さらに，NS の目的関数 $\ell^{NS}(\theta)$ は以下のように定義できる．

$$\ell^{NS}(\theta) = -\frac{1}{|D|}\sum_{(x,y)\in D}\Big[\log(P(C=1,y|x;\theta)) + \sum_{i=1,y_i\sim p_n}^{\nu}\log(P(C=0,y_i|x;\theta))\Big]. \tag{7}$$

ここで Bregman 距離を使って目的関数 $\ell^{NS}(\theta)$ について以下の性質を導くことができる．

2) 本項では，ラベル y に対する確率を $p(y)$，全てのラベル $\mathbf{y}$ に対する確率値のベクトルを $p(\mathbf{y})$ のように記述する．

命題 1 $\Psi(z) = z\log(z) - (1+z)\log(1+z)$ とすることで，式 (3) から $\ell^{NS}(\theta)$ を導くことができ，$\ell^{NS}(\theta) = 0$ のとき，次の式が成立する:

$$G(y|x;\theta) = \frac{p_d(y|x)}{\nu p_n(y|x)}. \tag{8}$$

命題 2 $P_\theta(y|x)$ に対して，$\ell^{NS}(\theta)$ の目的分布は:

$$\frac{p_d(y|x)}{p_n(y|x)\sum_{y_i\in Y}\frac{p_d(y_i|x)}{p_n(y_i|x)}}. \tag{9}$$

証明 命題 1, 2 の証明は付録に記す □

上記の結果に対して，例えば，$p_n(y|x) = p_d(y)$ とすれば，式 (8) は自己相互情報量から定数 $\log\nu$ を減じた形となる．これは文献 [7] の結論と同様であり，我々の結論はその一般化となっている．

3.1 様々な雑音分布

SCE 損失の目的分布とは異なり，式 (9) は雑音分布 $p_n(y|x)$ の影響を受ける．よって，特定の雑音分布 $p_n(y|x)$ を考えることで，式 (9) のより詳細な分析を行う．また，そこから SCE との関係性も導く．

3.1.1 一様雑音分布

知識グラフ埋め込みの学習において最もよく使われている離散の一様雑音分布 $u\{1,|Y|\}$ を考える．$p_n(y|x)$ が一様のとき，以下の性質を導出できる．

命題 3 式 (9) は $p_d(y|x)$ となる．

証明 $p_n(y|x)\sum_{y_i\in Y}\frac{p_d(y_i|x)}{p_n(y_i|x)} = \sum_{y_i\in Y} p_d(y_i|x) = 1$ であり，式 (9) は $p_d(y|x)$ となる． □

この結論は $P_n(y|x)$ が一様であるとき，ℓ^{NS} の目的分布が SCE 損失と一致することを示している．

3.1.2 自己敵対雑音分布

Sun ら [8] は自己敵対負例サンプリング（SANS）を提案している．SANS は $p_\theta(y|x)$ を雑音分布として利用するので，式 (9) は次のように書ける:

$$\frac{p_d(y|x)}{p_{\hat\theta}(y|x)\sum_{y_i\in Y}\frac{p_d(y_i|x)}{p_{\hat\theta}(y_i|x)}}. \tag{10}$$

ここで $\hat\theta$ は最後に更新された結果に基づくパラメータである．式 (10) を解析的に理解することは難しいが，$p_\theta(y|x)$ の特殊な場合について考えることで

分析を進める．学習前，パラメータ θ はランダムに初期化されるため，$p_\theta(y|x)$ は $u\{1,|Y|\}$ に従う．その際，式 (10) は $p_d(y|x)$ となる．逆に，$p_{\hat{\theta}}(y|x)$ を $p_d(y|x)$ に設定すると，式 (10) は $u\{1,|Y|\}$ となる．このように，式 (10) では，$p_{\hat{\theta}}(y|x) \to u\{1,|Y|\}$ となるとき，$p_\theta(y|x) \to p_d(y|x)$，$p_{\hat{\theta}}(y|x) \to u\{1,|Y|\}$ となるとき，$p_\theta(y|x) \to p_d(y|x)$ と収束する．

実際のミニバッチ学習において，θ は各バッチデータに対して逐次的に更新が行われるため，SANS の目的分布は $p_d(y|x)$ と $u\{1,|Y|\}$ の間での均衡に基づいて決定されていると考えられる．この考えに基づいて，$p_d(y|x)$ と $u\{1,|Y|\}$ の重み付き和を SANS の目的分布と仮定する．SANS では $p_{\hat{\theta}}(y|x)$ を以下のように定式化している:

$$p_{\hat{\theta}}(y|x) = \frac{\exp(\alpha f_\theta(x,y))}{\sum_y \exp(\alpha f_\theta(x,y))}. \tag{11}$$

α は温度パラメータであり，これも上記の均衡を調整する役割がある．従って，$p_\theta(y|x)$ に対する SANS の目的分布を以下として表す:

$$p_\theta(y|x) \approx (1-\lambda)p_d(y|x) + \lambda u\{1,|Y|\}. \tag{12}$$

λ は $p_\theta(y|x)$ を $p_d(y|x)$ または $u\{1,|Y|\}$ のどちらに近づけるかを決定するハイパーパラメータである．

式 (12) の結果を用いて同様の目的分布を持つ SCE 損失を確認する．$q(y|x) = (1-\lambda)p_d(y|x) + \lambda u\{1,|Y|\}$ と $\Psi(\mathbf{z}) = \sum_{i=1}^{len(\mathbf{z})} z_i \log z_i$ と置き，式 (5) から式 (6) への変換に基づいて，新しい SCE 損失の形式

$$\begin{aligned} &B_{\Psi(\mathbf{z})}(p_d(\mathbf{y}|x), p_\theta(\mathbf{y}|x)) \\ &= -\sum_{(x,y)\in D}\Bigg[\sum_{i=1}^{|Y|}(1-\lambda)p_d(y_i|x_t)\log p_\theta(y_i|x_t) \\ &\qquad + \sum_{i=1}^{|Y|}\lambda u\{1,|Y|\}\log p_\theta(y_i|x_t)\Bigg]p_d(x,y) \\ &= -\frac{1}{|D|}\sum_{(x,y)\in D}\Bigg[(1-\lambda)\log p_\theta(y|x) \\ &\qquad + \sum_{i=1}^{|Y|}\frac{\lambda}{|Y|}\log p_\theta(y_i|x)\Bigg] \end{aligned} \tag{13}$$

が導出される．この形式は SCE 損失に対するラベルスムージング [9] と等価であり，SANS がラベルスムージングと同様の効果を持つことがわかる．

4 SCE と NS 損失の統一的解釈

3) なお，w / LS はラベルスムージングを，w/ Uni は雑音として離散一様分布を使用したことをそれぞれ表す．

表 1 各損失関数間の関係[3)]．

損失	目的分布	$\Psi(z)$ または $\Psi(\mathbf{z})$
NS w/ Uni	$p_d(y\|x)$	$z\log z - (1+z)\log(1+z)$
SANS	$\lambda p_d(y\|x) + (1-\lambda)u\{1,\|Y\|\}$	$z\log z - (1+z)\log(1+z)$
SCE	$p_d(y\|x)$	$\sum_{i=1}^{len(\mathbf{z})} z_i \log z_i$
SCE w/ LS	$\lambda p_d(y\|x) + (1-\lambda)u\{1,\|Y\|\}$	$\sum_{i=1}^{len(\mathbf{z})} z_i \log z_i$

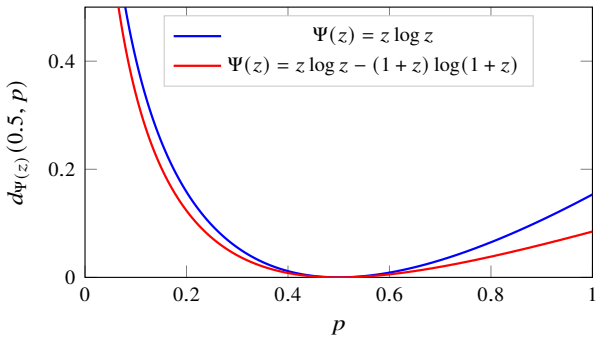

図 1 確率 p と 0.5 に対し，各 Ψ で $d_{\Psi(z)}$ が示す距離．

各損失関数の関係を表 1 に示す．この表から読み取れるように，一様雑音分布の下での NS 損失と SCE 損失の目的分布が等価なものであることが分かる．また，自己敵対雑音分布を用いた際の NS 損失とラベルスムージングを用いた際の SCE 損失の目的分布が極めて類似していることも分かる．これらの知見は NS 損失と SCE 損失に基づく手法を公平に比較する際に重要である．なぜならば，対象とするデータセットに疎なエンティティが含まれている場合，ラベルスムージングを用いた SCE 損失と SANS 損失では，モデルによる性能改善が存在しなくとも，スムージングにより性能が改善する可能性があるためである．このことから，目的分布の観点に基づいて NS 損失に基づくモデルと SCE 損失に基づくモデルを公平に比較するためには，NS 損失で一様雑音分布が使用されている場合には，SCE 損失をそのまま使用し，NS 損失で一様雑音分布が使用されている場合には，SCE 損失でラベルスムージングを使用することが必要である．

NS 損失と SCE 損失における $\Psi(z)$ を比較することは目的分布に着目することと同等以上に重要である．これは $\Psi(z)$ が損失における距離を決定し，その距離がモデルのデータセットに対する振る舞いを決定する上で重要な役割を果たすためである．

図 1 に表 1 で示されている NS 損失と SCE 損失のそれぞれにおける Ψ を用いて式 (2) の距離を計算し，確率 p のそれぞれの値と確率 0.5 との距離を示した．この図から読み取れるように，SCE 損失の方が NS 損失よりも大きな距離を示すことが分かる．

表 2　それぞれのモデルと損失関数の組み合わせを用いた場合の FB15k-237 と WN18RR での実験結果.

Model	Loss	FB15k-237		WN18RR	
		MRR	Hits@3	MRR	Hits@3
TuckER	NS	0.257	0.297	0.431	0.440
	SANS	0.330	0.365	0.445	0.455
	SCE	0.338	0.372	0.453	0.465
	SCE w/ LS	0.343	0.378	0.472	0.483
RESCAL	NS	0.337	0.368	0.385	0.405
	SANS	0.339	0.372	0.389	0.404
	SCE	0.352	0.387	0.451	0.470
	SCE w/ LS	0.363	0.400	0.469	0.485
ComplEx	NS	0.296	0.323	0.394	0.403
	SANS	0.299	0.327	0.432	0.442
	SCE	0.297	0.325	0.463	0.473
	SCE w/ LS	0.317	0.348	0.477	0.491
RotatE	NS	0.301	0.333	0.469	0.484
	SANS	0.333	0.371	0.472	0.487
	SCE	0.315	0.347	0.452	0.463
	SCE w/ LS	0.315	0.346	0.447	0.461

なお，文献 [10] では，二値ラベルにおける Bregman 距離の上界は $\Psi(z) = z\log(z)$ とした際であることが示されており，我々の観測はこの結果に沿っている．SCE 損失と NS 損失が分布間に与える距離の違いは，モデルの差異に起因する問題を生じさせる原因となりうる．SCE を用いた場合には，自由度が高く表現力が高いモデルを訓練データに適合させることに寄与する一方，事前分布や制約に基づく表現力が低いモデルでは学習を妨げる可能性がある．また，NS を用いた場合に，自由度が高く表現力が高いモデルでは過少適合を起こす可能性がある一方，事前分布や制約により表現力が制限されるモデルでは，訓練事例への過度な適合を回避し，学習の進行に寄与する可能性がある．これらの問題は対象とするモデルやデータによって振る舞いが決定されるため，現在主流となっている設定に基づいて影響を実験的に検証する．

5　実験

本節では SCE 損失と NS 損失の性質を検証するための実験を行う．データセットには FB15k-237 [11] と WN18RR [12] を使用し，評価には MRR，Hits@3 を用いる．比較対象のモデルとして TuckER [13]，RESCAL [1]，ComplEx [14]，RotatE [8] を選択し，実装は LibKGE [15][4] のものを利用した．これらのモデルのハイパーパラメータは RESCAL，ComplEx については先行研究 [4] で最高精度を達成した設定を，TuckER，RotatE については元の論文の設定を使用した．SANS を適用する際には RotatE 以外のモデルでは LibKGE の初期値である 1.0 を使用し，RotatE では元の論文で SANS が使用されているため，その設定に従った．ラベルスムージングを適用する際には，RotatE 以外のモデルでは LibKGE の初期値である 0.3 を使用したが，RotatE では比較対象である SANS の値がチューニングされているため，公平性のために開発データを用いて{0.3, 0.1, 0.01}から値を選択した．他の詳細な設定は付録に記載した．

表 2 に実験結果を示す．SCE w/ LS で性能向上する時に，多くの場合で SANS でも性能が向上している．この結果からいずれのデータセットでも疎なエンティティが問題となっていることが分かる．またこの結果は SCE w/ LS と SANS が対応しているという我々の解釈に沿うものである．なお，RotatE ではこの関係が成立していないが，後述するようにこの結果は SCE が訓練データに適合できていないためであると考えられる．訓練データに適合していないのであれば出力が既に平滑化されており，SCE w/ LS の効果が抑制されるからである．

スコアリング法と損失関数の組み合わせの点からは，TuckER，RESCAL，ComplEx といった表現力が高い手法では SCE と比較して NS の結果が高くないことから，SCE の過剰適合ではなく，むしろ NS の過少適合が問題であると考えられる．さらに，モデル上の制約が多い RotatE では NS でより高い性能を発揮していることから，我々が予想したように自由度が高いモデルでは SCE が有用で，制約に基づくモデルでは NS が有用であることが分かる．

6　まとめ

本稿では Bregman 距離を用いて知識グラフの埋め込み学習における SCE 損失と NS 損失の二つの関数を統一的に解釈した．その結果，目的分布の点からは SCE と一様雑音分布に基づく NS が等価であり，また SCE でラベルスムージングを用いた場合と自己敵対的雑音に基づく NS が類似していることを示した．さらに，SCE は NS よりも大きな距離を持つことを明らかにし，実験により SCE は表現力の高いスコアリング法と相性が良く，NS は制約をもつ自由度の低いスコアリング法との相性が良いことを示した．

4)　https://github.com/uma-pi1/kge

参考文献

[1] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, p. 301–306. AAAI Press, 2011.

[2] Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. Knowledge base completion: Baselines strike back. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 69–74, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 26, pp. 2787–2795. Curran Associates, Inc., 2013.

[4] Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*, 2020.

[5] L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, Vol. 7, No. 3, pp. 200 – 217, 1967.

[6] Michael U. Gutmann and Jun-ichiro Hirayama. Bregman divergence as general framework to estimate unnormalized statistical models. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI' 11, p. 283–290, Arlington, Virginia, USA, 2011. AUAI Press.

[7] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 27, pp. 2177–2185. Curran Associates, Inc., 2014.

[8] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *CoRR*, Vol. abs/1902.10197, , 2019.

[9] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Vol. 32, pp. 4694–4703. Curran Associates, Inc., 2019.

[10] A. Painsky and G. W. Wornell. Bregman divergence bounds and universality properties of the logarithmic loss. *IEEE Transactions on Information Theory*, Vol. 66, No. 3, pp. 1658–1673, 2020.

[11] Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pp. 57–66, Beijing, China, July 2015. Association for Computational Linguistics.

[12] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, pp. 1811–1818, February 2018.

[13] Ivana Balazevic, Carl Allen, and Timothy Hospedales. TuckER: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5185–5194, Hong Kong, China, November 2019. Association for Computational Linguistics.

[14] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, pp. 2071–2080, 2016.

[15] Samuel Broscheit, Daniel Ruffinelli, Adrian Kochsiek, Patrick Betz, and Rainer Gemulla. LibKGE - A knowledge graph embedding library for reproducible research. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 165–174, 2020.

A 付録

A.1 命題 1, 2 の証明

ℓ^{NS} は以下のように定式化できる:

$$
\begin{aligned}
\ell^{NS}(\theta) &= -\frac{1}{|D|}\sum_{(x,y)\in D}\left(\log(P(C=1,y|x;\theta)) + \sum_{i=1,y_i\sim p_n}^{\nu}\log(P(C=0,y_i|x;\theta))\right)\\
&= -\frac{1}{|D|}\sum_{(x,y)\in D}\log(P(C=1,y|x;\theta)) - \frac{1}{|D|}\sum_{(x,y)\in D}\sum_{i=1,y_i\sim p_n}^{\nu}\log(P(C=0,y_i|x;\theta))\\
&= -\frac{1}{|D|}\sum_{(x,y)\in D}\log(\frac{1}{1+G(y|x;\theta)}) - \frac{1}{|D|}\sum_{(x,y)\in D}\sum_{i=1,y_i\sim p_n}^{\nu}\log(\frac{G(y_i|x_t;\theta)}{1+G(y_i|x_t;\theta)})\\
&= \frac{1}{|D|}\sum_{(x,y)\in D}\log(1+G(y|x;\theta)) + \frac{\nu}{\nu|D|}\sum_{(x,y)\in D}\sum_{i=1,y_i\sim p_n}^{\nu}\log(1+\frac{1}{G(y_i|x;\theta)})\\
&= \sum_{(x,y)\in D} p_d(y|x)\log(1+G(y|x;\theta))p_d(x) + \sum_{(x,y)\in D}\nu p_n(y|x)\log(1+\frac{1}{G(y|x;\theta)})p_d(x) \qquad (14)
\end{aligned}
$$

ここで $u=(x,y)$, $f(u)=\frac{\nu p_n(y|x)}{p_d(y|x)}$, $g(u)=G(y|x;\theta)$, $p_d(x)=\frac{1}{p_d(y|x)}p_d(x,y)$ と置くと，式 (14) を以下のように書き換えることができる:

$$
\begin{aligned}
\ell^{NCE}(\theta) &= \left(\sum_{(x,y)\in D} p_d(y|x)\log(1+g(u))\frac{1}{p_d(y|x)}p_d(x,y) + \sum_{(x,y)\in D}\nu p_n(y|x)\log(1+\frac{1}{g(u)})\frac{1}{p_d(y|x)}p_d(x,y)\right)\\
&= \sum_{(x,y)\in D}\left[\log(1+g(u)) + \log(1+\frac{1}{g(u)})f(u)\right]p_d(x,y)\\
&= \sum_{(x,y)\in D}\left[\log(1+g(u)) - \log(g(u))f(u) + \log(1+g(u))f(u)\right]p_d(x,y)\\
&= \sum_{(x,y)\in D}[-g(u)\log(1+g(u)) + (1+g(u))\log(1+g(u)) + \log(g(u))g(u) + \log(1+g(u))g(u)\\
&\quad - \log(g(u))f(u) + \log(1+g(u))f(u)\ ]\,p_d(x,y) \qquad (15)
\end{aligned}
$$

$\Psi(g(u)) = g(u)\log(g(u)) - (1+g(u))\log(1+g(u))$ と $\Psi'(g(u)) = \log(g(u)) - \log(1+g(u))$ に基づいて，式 (15) をさらに変形できる:

$$
\ell^{NCE}(\theta) = \sum_{(x,y)\in D}[-\Psi(g(u)) + \Psi'(g(u))g(u) - \Psi'(g(u))f(u)]\,p_d(x,y) = B_{\Psi}(g(u), f(u)). \qquad (16)
$$

式 (16) から，$g(u) = f(u)$ で $\ell^{NS}(\theta)$ が最小化されるとき，$G(y|x;\theta)$ は $\frac{\nu p_n(y|x)}{p_d(y|x)}$ となる．そして，$\exp(f_\theta(x,y))$ は

$$
\exp(f_\theta(x,y)) = \frac{p_d(y|x)}{\nu p_n(y|x)} \qquad (17)
$$

となる．よって，ソフトマックス関数の定義から，$p_\theta(y|x)$ の目的分布は以下となる:

$$
p_\theta(y|x) = \frac{p_d(y|x)}{p_n(y|x)\sum_{y_i\in Y}\frac{p_d(y_i|x)}{p_n(y_i|x)}}. \qquad (18)
$$

A.2 使用したハイパーパラメータ

表 3 に使用したハイパーパラメータの一覧を示す．. Dim はエンティティ埋め込みの次元数を，LS はラベルスムージングを AS は敵対雑音サンプリングの温度パラメータをそれぞれ表す．各モデルは両方向のクエリに対応できるようエンティティの埋め込みのみを共有した逆側モデルも学習し，最大エポック数を 800 とした．また，開発データにて 5 エポック毎に MRR を計算し，その最高値が 10 回更新されなかった際に学習を打ち切った．

表 3 使用したハイパーパラメータの一覧

FB15k-237

Model		Batch	Dim	Initialize	Regularize Type	Regularize Entity	Regularize Relation	Dropout Entity	Dropout Relation	Optimizer Type	LR	Decay	Patience	Sample subject	Sample object	LS	AS
Tucker	SCE	128	200	xavier normal: 1.0	-	-	-	0.3	0.4	Adam	0.0005	-	-	All	All	-	-
	SCE w/ LS	128	200	xavier normal: 1.0	-	-	-	0.3	0.4	Adam	0.0005	-	-	All	All	0.3	-
	NS	128	200	xavier normal: 1.0	-	-	-	0.3	0.4	Adam	0.0005	-	-	All	All	-	-
	SANS	128	200	xavier normal: 1.0	-	-	-	0.3	0.4	Adam	0.0005	-	-	All	All	-	1.0
Rescal	SCE	512	128	normal, mean:0.0, std: 0.123	-	-	-	0.427	0.159	Adam	0.000739	0.95	1	All	All	-	-
	SCE w/ LS	512	128	normal, mean:0.0, std: 0.123	-	-	-	0.427	0.159	Adam	0.000739	0.95	1	All	All	0.3	-
	NS	256	128	xavier normal: 1.0	lp: 3	1.22E-12	4.80E-14	0.347	-	Adagrad	0.0170	0.95	5	22	155	-	-
	SANS	256	128	xavier normal: 1.0	lp: 3	1.22E-12	4.80E-14	0.347	-	Adagrad	0.0170	0.95	5	22	155	-	1.0
ComlEx	SCE	512	128	uniform: [-0.311,0.311]	-	-	-	0.0476	0.443	Adagrad	0.503	0.95	5	All	All	-	-
	SCE w/ LS	512	128	uniform: [-0.311,0.311]	-	-	-	0.0476	0.443	Adagrad	0.503	0.95	5	All	All	0.3	-
	NS	512	256	normal, mean:0.0, std: 0.000481	lp: 2	6.34E-9	9.08E-18	0.182	0.0437	Adagrad	0.241	0.95	7	1	48	-	-
	SANS	512	256	normal, mean:0.0, std: 0.000481	lp: 2	6.34E-9	9.08E-18	0.182	0.0437	Adagrad	0.241	0.95	7	1	48	-	1.0
Rotate	SCE	1024	1000	xavier uniform: 1.0	-	-	-	-	-	Adam	0.00005	0.95	5	All	All	-	-
	SCE w/ LS	1024	1000	xavier uniform: 1.0	-	-	-	-	-	Adam	0.00005	0.95	5	All	All	0.01	-
	NS	1024	1000	xavier uniform: 1.0	-	-	-	-	-	Adam	0.00005	0.95	5	256	256	-	-
	SANS	1024	1000	xavier uniform: 1.0	-	-	-	-	-	Adam	0.00005	0.95	5	256	256	-	1.0

WN18RR

Model		Batch	Dim	Initialize	Regularize Type	Regularize Entity	Regularize Relation	Dropout Entity	Dropout Relation	Optimizer Type	LR	Decay	Patience	Sample subject	Sample object	LS	AS
Tucker	SCE	128	200	xavier normal: 1.0	-	-	-	0.2	0.2	Adam	0.0005	-	-	All	All	-	-
	SCE w/ LS	128	200	xavier normal: 1.0	-	-	-	0.2	0.2	Adam	0.0005	-	-	All	All	0.3	-
	NS	128	200	xavier normal: 1.0	-	-	-	0.2	0.2	Adam	0.0005	-	-	All	All	-	-
	SANS	128	200	xavier normal: 1.0	-	-	-	0.2	0.2	Adam	0.0005	-	-	All	All	-	1.0
Rescal	SCE	512	256	xavier normal: 1.0	-	-	-	-	-	Adam	0.00246	0.95	9	All	All	-	-
	SCE w/ LS	512	256	xavier normal: 1.0	-	-	-	-	-	Adam	0.00246	0.95	9	All	All	0.3	-
	NS	512	128	normal, mean:0.0, std:0.00164	-	-	-	-	-	Adam	0.00152	0.95	1	6	8	-	-
	SANS	512	128	normal, mean:0.0, std:0.00164	-	-	-	-	-	Adam	0.00152	0.95	1	6	8	-	1.0
ComlEx	SCE	512	128	uniform:[-0.281,0.281]	lp: 2	4.52E-6	4.19E-10	0.359	0.311	Adagrad	0.526	0.95	5	All	All	-	-
	SCE w/ LS	512	128	uniform:[-0.281,0.281]	lp: 2	4.52E-6	4.19E-10	0.359	0.311	Adagrad	0.526	0.95	5	All	All	0.3	-
	NS	1024	128	xavier normal: 1.0	-	-	-	0.0466	0.0826	Adam	0.000332	0.95	7	6	6	-	-
	SANS	1024	128	xavier normal: 1.0	-	-	-	0.0466	0.0826	Adam	0.000332	0.95	7	6	6	-	1.0
Rotate	SCE	512	500	xavier uniform: 1.0	-	-	-	-	-	Adam	0.00005	0.95	5	All	All	-	-
	SCE w/ LS	512	500	xavier uniform: 1.0	-	-	-	-	-	Adam	0.00005	0.95	5	All	All	0.01	-
	NS	512	500	xavier uniform: 1.0	-	-	-	-	-	Adam	0.00005	0.95	5	1024	1024	-	-
	SANS	512	500	xavier uniform: 1.0	-	-	-	-	-	Adam	0.00005	0.95	5	1024	1024	-	0.5