The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

潜在変数モデルとナレッジグラフ埋め込みの融合

Incorporating Knowledge Graph Embeddings into Latent Variable Models

武石 直也 *1　秋元 康佑 *1
Naoya Takeishi　Kosuke Akimoto

*1東京大学 航空宇宙工学専攻
Department of Aeronautics and Astronautics, The University of Tokyo

In this talk, we introduce an idea for incorporating information encoded in a knowledge graph to a latent variable model (LVM). We propose an extension of an LVM by two-view modeling, where the parameters and the latent variables of the LVM are shared between the original LVM and a probabilistic model for knowledge graph embedding. We specifically introduce how to incorporate a knowledge graph into probabilistic principal component analysis and show preliminary experimental results.

1. 背景

混合モデルや因子分析などの（特に、教師なしの）潜在変数モデル（latent variable model, LVM）は、次元削減やクラスタリングによる探索的データ分析をはじめとする種々のタスクに活用されている。LVM の潜在変数やパラメタはデータをもとに推定されるが、その推定量がどのような実際的・物理的な意味合いを持つかは多くの場合定かではない。例えば、主成分分析で得られた主成分方向の意味は必ずしも専門家から見てわかりよいとは限らない。無論それこそがデータ分析の意義のひとつといえる（常に既存の知識で説明できてしまう結果が出ていては新しい知見はない）が、一方で専門家の知識を LVM に事前知識として「入れ込む」ことができれば、データと既存知識との整合性／非整合性を定量的にとらえるのに役立つ可能性がある。一般的な LVM のモデル設計や推論・学習の過程においても、特徴量、仮説空間、正則化、初期値などの選択において事前知識といえるものが役立っている。また、データドメインに関してより直接的な形の事前知識として制約や論理式を LVM で考慮する研究もある [Fung 03, Varol 12, Andrzejewski 11, Mei 14, Foulds 15]。

本稿では、論理式などよりは間接的な形の事前知識として、**ナレッジグラフ**の形で表された知識を LVM に利用するための方法を提案する。ここでいうナレッジグラフとは、（通常、真の）タプル (h, r, t) の集合である。あるタプル (h, r, t) について、例えば

$$h = \text{Kagoshima}, \quad r = \text{is-located-in}, \quad t = \text{Japan},$$

であれば、そのタプルは真である。h や t はエンティティ、r はリレーションと呼ばれる。

ナレッジグラフは離散的なデータであるが、近年はエンティティやリレーションを連続値で表現するナレッジグラフ埋め込み（knowledge graph embedding, KGE）が盛んに研究され、リンク予測等のナレッジグラフ上のタスクのほか、question answering 等のナレッジグラフを利用するタスクに活用されている。よく知られた KGE として TransE [Bordes 13] とよばれる方法や DistMult [Yang 15] とよばれる方法がある。より複雑な最近の方法としては ConvE [Dettmers 17] などがある。本稿で紹介するのは、LVM にナレッジグラフによる知識を入れ込むために KGE を活用するという発想である。

連絡先: {takeishi,akimoto}@ailab.t.u-tokyo.ac.jp

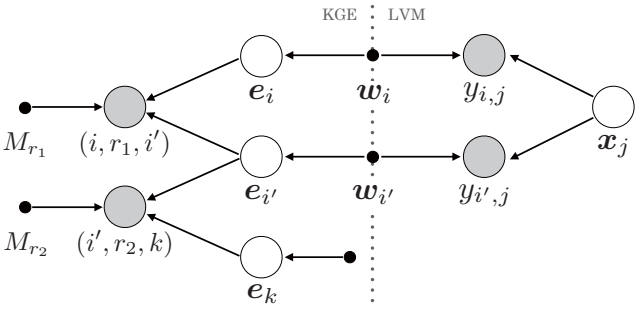

図 1: 提案手法のグラフィカルモデル表示の一部

2. 提案手法

まず、LVM および KGE 一般についてそれらを融合する方法論を紹介する。その後、確率的 PCA [Tipping 99] および KGE の一手法である DistMult [Yang 15] を組み合わせる場合についてのモデルや推論を特に記述する。

2.1 表記法

データは n 個のオブジェクトからなり、各々のオブジェクトは m 属性の観測からなるベクトル $\boldsymbol{y}_j \in \mathbb{R}^m$ である（ただし $j = 1, \ldots, n$）。$\boldsymbol{y}_j$ の i 番目の要素、すなわち属性 i のオブジェクト j での観測値を $y_{i,j} \in \mathbb{R}$ と表す（ただし $i = 1, \ldots, m$）。データ行列を $\boldsymbol{Y} = [\boldsymbol{y}_1 \ \cdots \ \boldsymbol{y}_n] \in \mathbb{R}^{m \times n}$ とする。

ある LVM が、パラメタの集合 θ および n 個の潜在変数ベクトル $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ を持っているとする（$\boldsymbol{x}_j \in \mathbb{R}^d$ は $\boldsymbol{y}_j$ に対応する）。$\theta = \{\boldsymbol{\pi}, \{\boldsymbol{w}\}\}$ の元には大域的パラメタと局所的パラメタの二種類があって、大域的パラメタ $\boldsymbol{\pi}$ が属性インデクス i によらないのに対して、m 個ある局所的パラメタ $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m\} \in \{\mathbb{R}^d\}^m$ は属性毎に存在するとする。*1 例えば確率的 PCA では、$\boldsymbol{\pi}$ には観測ノイズ分散などがあたり、$\boldsymbol{w}_i$ には因子負荷行列の i 行目があたる。また、LVM には観測モデル $p(\boldsymbol{Y} \mid \boldsymbol{X}, \theta)$ および事前分布 $p(\boldsymbol{X})$ が定まっているとする。パラメタの事前分布 $p(\theta)$ があってもよい。

KGE に関する量は次のように表す。エンティティの埋め込みは任意の添字とともに $\boldsymbol{e}_\cdot \in \mathbb{R}^q$ で表す。埋め込みのスコア付けのために用いるリレーション r に対応するパラメタは一般に

*1　$\boldsymbol{x}$ と $\boldsymbol{w}$ の次元は異なってもよいが、簡単のため両方 d とする。

M_r で表し、KGE のモデルによって行列 $\boldsymbol{M}_r$ やベクトル $\boldsymbol{m}_r$ として取られる。タプル (h,r,t) のスコアは $\psi(\boldsymbol{e}_h,\boldsymbol{e}_t,M_r)$ と書くことにする。また、タプルの負例は $(h,r,\tilde{t})$ などと表す。この場合は、タプル (h,r,t) に対して t だけを取り替えて作った負例という意味である。

2.2 LVM と KGE の two-view モデル

提案手法の主要な考え方を表すため、図 1 にグラフィカルモデルを示す。同図点線より右側は、ある LVM の観測モデルの一部（オブジェクト j と属性 i,i' に対応する部分）、すなわち

$$p(y_{i,j} \mid \boldsymbol{x}_j,\boldsymbol{w}_i)\cdot p(y_{i',j} \mid \boldsymbol{x}_j,\boldsymbol{w}_{i'}), \tag{1}$$

を表す。[*2] 同図点線より左側は KGE の一部を確率モデル化したものとその事前分布に対応していて、次の確率分布を表す：

$$\begin{gathered} p(\boldsymbol{e}_i \mid \boldsymbol{w}_i),\quad p(\boldsymbol{e}_{i'} \mid \boldsymbol{w}_{i'}),\quad p(\boldsymbol{e}_k),\\ p((i,r_1,i') \mid \boldsymbol{e}_i,\boldsymbol{e}_{i'},M_{r_1}),\quad p((i',r_2,k) \mid \boldsymbol{e}_{i'},\boldsymbol{e}_k,M_{r_2}). \end{gathered} \tag{2}$$

ここで、r_1 は属性 i と属性 i' との間のリレーション、r_2 は属性 i' とあるナレッジグラフ上のエンティティ k との間のリレーションを表す。このような two-view モデリングでは、LVM の局所的パラメタ $\boldsymbol{w}$ がエンティティの埋め込み $\boldsymbol{e}$ と関連づいている。このモデルの動機は、$\boldsymbol{w}$ を（$\boldsymbol{e}$ を通じて）ナレッジグラフのモデリングにも用いることによって、LVM の事前分布 $p(\boldsymbol{w})$ だけでなくナレッジグラフの構造も考慮した形で $\boldsymbol{w}$ の推定値や事後分布を得ることにある。

さて、式 (1) はオリジナルの LVM で定まっている一方、式 (2) の各分布は都合よく定めなくてはいけない。本稿では、

$$p(\boldsymbol{e} \mid \boldsymbol{f}(\boldsymbol{w})) = \mathcal{N}_{\boldsymbol{e}}(\boldsymbol{f}(\boldsymbol{w}),\boldsymbol{V}), \tag{3}$$

$$p(\boldsymbol{e}) = \mathcal{N}_{\boldsymbol{e}}(\boldsymbol{0},\boldsymbol{I}), \tag{4}$$

$$p((h,r,t)=\mathrm{true} \mid \boldsymbol{e}_h,\boldsymbol{e}_t,M_r) = \sigma(\psi(\boldsymbol{e}_h,\boldsymbol{e}_t,M_r)), \tag{5}$$

のようにモデリングすることを提案する。ただし、$\boldsymbol{f}:\mathbb{R}^d\to\mathbb{R}^q$ はユーザが設計すべき関数、σ はシグモイド関数である。また、$\boldsymbol{V}$ は大域的パラメタで $d\times d$ の実対称行列である。

推論や学習は基本的に通常の LVM と同様の手続きで行えばよい。ただし、KGE 部分の学習のためには予めナレッジグラフ上の負例を作成しておく前処理が重要となる（KGE に関して確率モデルを考えている先行研究だと [He 15, Xiao 16, Zhang 16] などを参照するとよい）。これは KGE 学習では一般的に行われている処理で、一種の data augmentation といえる。負例は正例タプルの一部をランダムサンプリングで変更して作成する（例えば、(h,r,t) から $(h,r,\tilde{t})$ を作る）ことが多いが、その詳細な方法は KGE のスコア関数設計と同様に本稿の対象の外となる。また、$\boldsymbol{e}$ の事後分布は解析的には計算できないので、サンプリングや近似推論が必要となる。しかし本稿では、ひとまず式 (3) のかわりに

$$p(\boldsymbol{e} \mid \boldsymbol{f}(\boldsymbol{w})) = \delta(\boldsymbol{e}-\boldsymbol{f}(\boldsymbol{w})), \tag{3'}$$

として（すなわち $\boldsymbol{e}=\boldsymbol{f}(\boldsymbol{w})$ として）、属性に対応しないエンティティについては式 (4) の事前分布は考えず、$\boldsymbol{e}$ をパラメタとして点推定する妥協策を用いる。$\boldsymbol{e}$ を確率変数としてみなすことでデータやナレッジグラフの不確かさを考慮することができるが、今後の課題とする。

*2 簡単のため j について独立な観測モデルであるとし、局所的パラメタの事前分布は考えず、図中では大域的パラメタを省略している。

2.3 ナレッジグラフ＋PCA

前節では LVM や KGE 一般について、それらを融合させるための two-view modeling に基づく方法論を提案した。この節では、そのような two-view modeling の例として、基本的な LVM のひとつである確率的 PCA [Tipping 99] および KGE のベースラインとしてよく参照される DistMult [Yang 15] を融合したモデルを定義し、学習方法を示す。

まず、確率的 PCA の観測モデルと事前分布は次の通り：

$$p(\boldsymbol{y}_j \mid \boldsymbol{x}_j,\theta) = \prod_{i=1}^{m}\mathcal{N}_{y_{i,j}}(\boldsymbol{w}_i^\top\boldsymbol{x}_j+\boldsymbol{\mu},\sigma^2), \tag{6}$$

$$p(\boldsymbol{x}_j) = \mathcal{N}_{\boldsymbol{x}_j}(\boldsymbol{0},\boldsymbol{I}). \tag{7}$$

ただし、$\boldsymbol{\mu}$, σ^2 は大域的パラメタである。次に、DistMult [Yang 15] のスコア関数は次のように定義される：

$$\psi(\boldsymbol{e}_h,\boldsymbol{e}_t,\boldsymbol{m}_r) = \boldsymbol{e}_h^\top\,\mathrm{diag}(\boldsymbol{m}_r)\boldsymbol{e}_t, \tag{8}$$

ただし、$\boldsymbol{m}_r\in\mathbb{R}^q$ はリレーション r に対応するパラメタである。$\boldsymbol{e}$ と $\boldsymbol{w}$ を関連させる関数 $\boldsymbol{f}$ の形は簡単にアフィン変換とする。すなわち、大域的パラメタ $\boldsymbol{A}\in\mathbb{R}^{q\times d}$, $\boldsymbol{b}\in\mathbb{R}^q$ を用いて

$$\boldsymbol{f}(\boldsymbol{w}) = \boldsymbol{A}\boldsymbol{w}+\boldsymbol{b}. \tag{9}$$

なお、$\boldsymbol{f}$ の設計には試行錯誤の余地が大いにある。確率的 PCA より複雑なモデルでは、j によって異なる $\boldsymbol{f}$ が必要な場合もある。

前述のように、$\boldsymbol{e}$ の事後分布推論は諦めて（というより、$\boldsymbol{e}$ を確率変数としないで）、$\boldsymbol{x}$ だけを潜在変数と見なす妥協策を採用する。このとき、学習時に最大化すべき不完全データ対数尤度は次の通り：

$$\begin{aligned} \mathcal{L} = &\sum_j \ln p(\boldsymbol{y}_j \mid \boldsymbol{W})\\ &+\sum_{(h,r,t)\in\mathcal{T}_1}\ln p((h,r,t) \mid \boldsymbol{e}_h,\boldsymbol{e}_t,\boldsymbol{m}_r)\\ &+\sum_{(i,r,t)\in\mathcal{T}_2}\ln p((i,r,t) \mid \boldsymbol{f}(\boldsymbol{w}_i),\boldsymbol{e}_t,\boldsymbol{m}_r)\\ &+\sum_{(h,r,i)\in\mathcal{T}_3}\ln p((h,r,i) \mid \boldsymbol{e}_h,\boldsymbol{f}(\boldsymbol{w}_i),\boldsymbol{m}_r)\\ &+\sum_{(i,r,i')\in\mathcal{T}_4}\ln p((i,r,i') \mid \boldsymbol{f}(\boldsymbol{w}_i),\boldsymbol{f}(\boldsymbol{w}_{i'}),\boldsymbol{m}_r). \end{aligned} \tag{10}$$

$\boldsymbol{x}$ は解析的に周辺化できるので、式 (10) の第一項は

$$\sum_j \ln\mathcal{N}_{\boldsymbol{y}_j}\left(\boldsymbol{\mu},\boldsymbol{W}\boldsymbol{W}^\top+\sigma^2\boldsymbol{I}\right),$$

と書ける。第二項以降は、例えば

$$\begin{aligned} \sum_{(i,r,t)\in\mathcal{T}_2}\Big\{&z_{(i,r,t)}\ln\left[\sigma(\psi(\boldsymbol{f}(\boldsymbol{w}_i),\boldsymbol{e}_t,\boldsymbol{m}_r))\right]\\ &+\left(1-z_{(i,r,t)}\right)\ln\left[1-\sigma(\psi(\boldsymbol{f}(\boldsymbol{w}_i),\boldsymbol{e}_t,\boldsymbol{m}_r))\right]\Big\}, \end{aligned}$$

のように書ける。ただし、$\mathcal{T}_1$ は head と tail の両者がデータの属性に関係しないタプルの集合、$\mathcal{T}_2$ は head だけがデータの属性に対応しているタプルの集合、以下同様に定義する。

3. 数値例

3.1 トイデータの用意と実験手順

次のように三次元の連続値データを生成した。まず、$\bar{y}_{1,1:n}$ および $\bar{y}_{2,1:n}$（ただし n は平方数）を $[-0.5, 1.5]^2$ 内の $\sqrt{n} \times \sqrt{n}$ 格子上の点としてとる。これをもとに $\bar{y}_{3,1:n}$ を次のように生成する：

$$\bar{y}_{3,j} = \bar{y}_{1,j}^3 + \bar{y}_{2,j}^3, \quad j = 1, \ldots, n. \tag{11}$$

このように生成した $\bar{y}$ に正規分布に従うノイズを加えてデータ $\boldsymbol{y}$ をつくる。

$$\boldsymbol{y}_j = \begin{bmatrix} \bar{y}_{1,j} \\ \bar{y}_{2,j} \\ \bar{y}_{3,j} \end{bmatrix} + \boldsymbol{v}_j, \quad \boldsymbol{v}_j \sim \mathcal{N}_{\boldsymbol{v}_j}(\boldsymbol{0}, 0.01\boldsymbol{I}) \tag{12}$$

また、利用するナレッジグラフとして単に

$$\mathcal{T} = \{(h = w_1, r = r_1, t = e_1), (w_2, r_1, e_1)\} \tag{13}$$

を用意する。w_1, w_2 はそれぞれデータの 1 次元目、2 次元目の特徴に対応するエンティティ、e_1 は何か任意のエンティティである。

以上をもとに、$q = 5, d = 2$ として次のような実験を行った。

1. 上記の手順でデータを生成する（$n = 400$）。
2. 訓練・検証用およびテスト用にデータを分割する。このとき、訓練・検証用データは範囲 $[-0.5, 1.5] \times [-0.5, 0.5]$ の 200 点、テスト用データは $[-0.5, 1.5] \times [0.5, 1.5]$ の 200 点を使用する（つまり、訓練時とテスト時で共変量シフトのようなことが起こっている設定）。訓練・検証用データは更にランダムに分割し、訓練用データ $\boldsymbol{Y}_{\mathrm{tr}} \in \mathbb{R}^{3\times 160}$、検証用データ $\boldsymbol{Y}_{\mathrm{val}} \in \mathbb{R}^{3\times 40}$ を用意する。テスト用データを $\boldsymbol{Y}_{\mathrm{te}} \in \mathbb{R}^{3\times 200}$ と表す。
3. $\boldsymbol{Y}_{\mathrm{tr}}$ から確率的 PCA のパラメタを学習する。このとき、次の三つの方法を試す：
   (a) データ共分散行列の固有値分解に基づく最尤推定 [Tipping 99]（$\mathcal{T}$ は使わない）
   (b) 勾配法による $-\mathcal{L}$ の最小化（$\mathcal{T}$ は使わない）
   (c) 勾配法による $-\mathcal{L}$ の最小化（$\mathcal{T}$ を使う）

   ただし、(b) および (c) では (a) の解を初期値とする。(c) で $\boldsymbol{e}$ の初期値はランダムに得る。さらに、(b) および (c) では勾配法の各ステップで $\boldsymbol{Y}_{\mathrm{val}}$ に対する再構成誤差を監視し、最小の検証再構成誤差を達成するパラメタを採用する。
4. 上記 (a)–(c) それぞれについて、$\boldsymbol{Y}_{\mathrm{te}}$ に対してテスト再構成誤差 $\boldsymbol{Y}_{\mathrm{te}}$ を計算して比較する。
5. 上記 1–3 を異なる乱数シードのもとで 100 回繰り返す。

3.2 実験結果

まず、方法 (b) および (c) による再構成誤差が方法 (a) によるものより明らかに小さくなったため、方法 (a) による再構成誤差については省略する。方法 (b) および (c) を比較するため、図 2 に、方法 (b) によるテストデータに対する平均再構成誤差から方法 (c) によるテストデータに対する平均再構成誤差を引いた値のヒストグラムを示す（なお、差をとる前の元の再構成誤差は 1.0 程度というオーダーである）。このとこから、ほとんどの場合にナレッジグラフを用いた方が用いない場合よりも再構成誤差が 10%前後小さくなることがわかる。参考までに、図 3 にある試行における訓練・検証用データとテスト用データ、および学習により得られた $\boldsymbol{W}$ に対応する平面をナレッジグラフを利用しない場合と利用する場合についてそれぞれ示す。

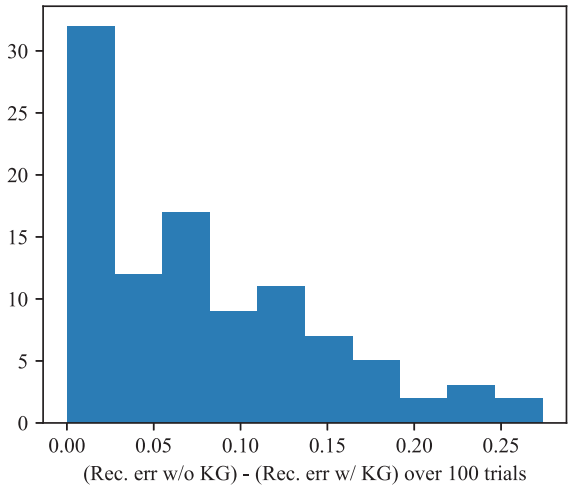

図 2: ナレッジグラフを使用した場合と使用しなかった場合のテスト再構成誤差の差に関するヒストグラム（正の値はナレッジグラフを使用した場合のテスト再構成誤差の方が使用しなかった場合より小さいことを示す）

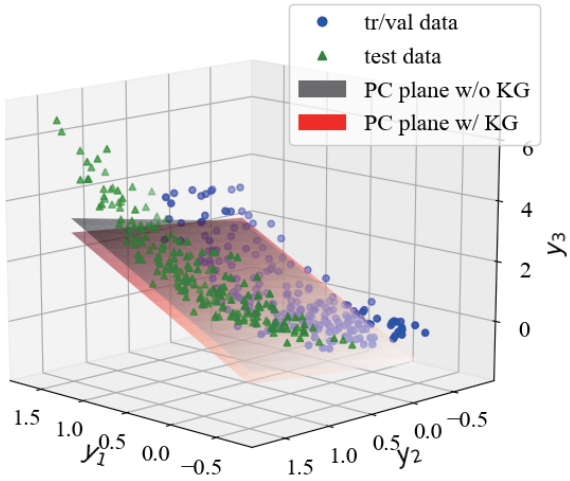

図 3: ある試行における訓練・検証データとテストデータ、および得られた $\boldsymbol{W}$ の示す平面

4. 今後の課題

本稿で紹介した方法では、$\boldsymbol{e}$ の事後分布推論についての方略が未完成である。局所的パラメタ $\boldsymbol{w}$ を点推定することにして、各局所的パラメタに対応するようなエンティティとナレッジグラフ中のエンティティとの関係のみを考慮するなら本稿で紹介した学習方法で十分であるかもしれないが、例えば各オブジェクトとナレッジグラフのエンティティとの関係も考慮したい際には、（オブジェクトに対応する $\boldsymbol{x}$ を確率変数として事後分布推論したい限りは）$\boldsymbol{e}$ の事後分布推論を考えなくてはな

らない。これについては、Pólya–Gamma data augmentation [Polson 13] による Gibbs サンプリングや、または確率的勾配変分ベイズ [Kingma 14] のような方法を検討したい。

また、当然実験も行う必要がある。まず、LVM 側の性能向上を示すという観点では、本稿で紹介したように PCA の再構成誤差が小さくなるなどの汎化性能向上を示したい。なお、本稿で作ったトイデータは結局線形モデルから生成されているために、ナレッジグラフの情報を用いない場合とあまり差が出なかったと考えられる。また、提案手法によってデータとナレッジグラフを同時に考慮できることから、KGE（によるタスク）の性能向上も期待される。これについても、実際利用されている大規模なナレッジグラフやそれに関連するデータを用いて検証したい。

参考文献

[Andrzejewski 11] Andrzejewski, D., Zhu, X., Craven, M., and Recht, B.: A Framework for Incorporating General Domain Knowledge into Latent Dirichlet Allocation using First-Order Logic, in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1171–1177 (2011)

[Bordes 13] Bordes, A., Usunier, N., Gracia-Durán, A., Weston, J., and Yakhnenko, O.: Translating Embeddings for Modeling Multi-relational Data, in *Advances in Neural Information Processing Systems*, Vol. 26, pp. 2787–2795 (2013)

[Dettmers 17] Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S.: Convolutional 2D Knowledge Graph Embeddings, arXiv:1707.01476 (2017)

[Foulds 15] Foulds, J., Kumar, S. H., and Getoor, L.: Latent Topic Networks: A Versatile Probabilistic Programming Framework for Topic Models, in *Proceedings of the 32nd International Conference on Machine Learning*, pp. 777–786 (2015)

[Fung 03] Fung, G. M., Mangasarian, O. L., and Shavlik, J. W.: Knowledge-Based Support Vector Machine Classifiers, in *Advances in Neural Information Processing Systems*, Vol. 15, pp. 537–544 (2003)

[He 15] He, S., Liu, K., Ji, G., and Zhao, J.: Learning to Represent Knowledge Graphs with Gaussian Embedding, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 623–632 (2015)

[Kingma 14] Kingma, D. P. and Welling, M.: Stochastic Gradient VB and the Variational Auto-Encoder, in *Proceedings of the 2nd International Conference on Learning Representations* (2014)

[Mei 14] Mei, S., Zhu, J., and Zhu, X.: Robust RegBayes: Selectively Incorporating First-Order Logic Domain Knowledge into Bayesian Models, in *Proceedings of the 31st International Conference on Machine Learning*, No. 1, pp. 253–261 (2014)

[Polson 13] Polson, N. G., Scott, J. G., and Windle, J.: Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables, *Journal of the American Statistical Association*, Vol. 108, No. 504, pp. 1339–1349 (2013)

[Tipping 99] Tipping, M. E. and Bishop, C. M.: Probabilistic Principal Component Analysis, *Journal of the Royal Statistical Society: Series B*, Vol. 61, No. 3, pp. 611–622 (1999)

[Varol 12] Varol, A., Salzmann, M., Fua, P., and Urtasun, R.: A Constrained Latent Variable Model, in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2248–2255 (2012)

[Xiao 16] Xiao, H., Huang, M., Hao, Y., and Zhu, X.: TransG : A Generative Model for Knowledge Graph Embedding, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 2316–2325 (2016)

[Yang 15] Yang, B., Yih, tau W., He, X., Gao, J., and Deng, L.: Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in *Proceedings of the 3rd International Conference on Learning Representations* (2015)

[Zhang 16] Zhang, F., Yuan, N. J., Lian, D., Xie, X., and Ma, W.-Y.: Collaborative Knowledge Base Embedding for Recommender Systems, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 353–362 (2016)