Thompson Sampling and Compositions in Knowledge Bases with Uncertainty

Abstract

We are concerned with the problem of knowledge base completion, or inferring missing facts from known relations. Human can readily infer a new fact through the composition of known facts in knowledge base, whereas the current statistical relational models lack the consideration of active knowledge acquisition through the composition of knowledge. We propose a probabilistic model to bridge this gap. We start from a new formulation of vector space embedding for knowledge tensors, that explicitly model distributions of relations. This enables us to extend Thompson sampling approach to knowledge bases, and also incorporate additive and multiplicative approaches for composing relations. On synthetic and real world datasets, we find that learning with composition is helpful when training data is sparse, and that Thompson sampling provide effective exploit-explore strategies that balance recall and reconstruction accuracy.

1. Introduction

Relational knowledge bases support reasoning, information retrieval, or question-answering tasks about entities and their relations. Most of them contact facts in the form of (entity1, relation, entity2) triples, such as (CarlFriedrichGauss, BornIn, Braunschweig). Automatically acquiring, maintaining, and reasoning in knowledge bases is a very active topic area with many important and challenging research questions. Even the largest of such databases are known to be incomplete (Dong et al., 2014). There are two main ways to fill in the missing facts, the first learning new relations from large collections of text or hyper-text, known as knowledge extraction, the second is inferring facts from existing relations, known as knowledge completion.

One challenge in knowledge base completion is its disconnection from the knowledge extraction setting. It would be nice, for example, to know which question to query for in a search-based method for gathering triples (West et al., 2014). Recently Kajino et al. (2015) propose an active learning strategy for completing knowledge triples, however the algorithm find it difficult to achieve high recall and high reconstruction at the same time. Having an active exploit-explore strategy would more effectively connect knowledge completion and extraction problems. Another challenge is leveraging compositional knowledge from existing triples. Also known as paths in knowledge graphs, composition of facts is key to achieving common reasoning tasks. For example, the two triples (CarlFriedrichGauss, BornIn, Braunschweig) and (Braunschweig, LocatedIn, Germany) implies (CarlFriedrichGauss, BornIn Germany). Path ranking (Lao & Cohen, 2010), vector space traversal (Guu et al., 2015) and composition (Neelakantan et al., 2015) techniques are recently developed to leverage such information for knowledge completion. We note, however, that a principled formulation that can address both open challenges is still missing – namely, a relational model that can model knowledge compositions in an active setting.

We propose a novel probabilistic re-formulation of tensor factorisation, one of the main competitive variants for knowledge completion (Nickel et al., 2015). We name our model Bayesian RESCAL (BRESCAL). The probabilistic model provides a natural way of embracing uncertainty of triples that is crucial to develop an active triple selection for knowledge completion, using Thompson sampling (Scott, 2010) – an approach for solving the multi-armed bandit problem, which allows us to trade-off exploration and exploitation when identifying new triples. We also perform compositional training for the probabilistic model, by modeling relation compositions as algebraic operations in the probabilistic embedding space. For inference, we design Gibbs sampling for BRESCAL with and without and compositions. We employ a sequential Monte-Carlo method for the active querying of new triples, called particle Thompson sampling.

We first test the proposed models with synthetic datasets. We observe that Thompson sampling provide significant gain over random sampling of triples, we also observe a clear gain in tensors with known composition structure. We then evaluated the model on three real-world relation datasets. In passive learning setting, we find BRESCAL outperforming the non-probabilistic version of tensor factorisation, and that compositions help when training set

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

is sparse. In the active learning scenario, we find that BRESCAL achieves the highest cumulative gain across all datasets. It is encouraging to see the exploit-explore strategy with uncertainty outperforms active learning strategies that focus on exploit or favours uncertainty.

We are pleased to be able to learn a vector-space model for entities and relations with uncertainty, bridging the gaps of probabilistic active sampling and knowledge compositions. We look forward to follow-on work with better sampling strategies and computational scalability.

2. Related Work

The literature on data factorisation and vector space models for relational data is vast. We give a brief overview of related work along three design choices: the method, the learning strategy, and the data representation. We then use these dimensions to help position our work.

Bayesian/Non-Bayesian This refers to two broad classes of model formulation, whether the obtained model is a point estimate or posterior distribution, whether to incorporate priors, quantification of uncertainty, and so on.

Passive/Active This refers to two different learning strategies, of passively learning a model given labeled data points, or actively requesting data points to be labeled.

Matrix/Tensor/Composition Relational learning problems for operate on different data representations. Matrix representation is common when dataset can be represented as a bi-partite graph, such as in (user, item) tuples in recommenders systems setting. Tensor representation is handy when edges in the graph has labels, i.e. (entity1, relation, entity2). We can think of compositions as paths in the graph, i.e. entity1 – relation1 – entity2 – relation2 – entity3.

In Table 1, we summarise a sample of related recent work along all combinations in each dimension. Note that N,A,C, or Non-Bayesian Active Composition model, can be done by simply using the query strategies from Kajino et al. (2015) on the compositional model by Guu et al. (2015). Our work address a critical gap in Bayesian Tensor factorisation capable of learning in the Active setting with relation Compositions.

Given this position, our work is inspired by, and most closely related to: active multi-relational data construction (AMDC) with tensor factorisation (Kajino et al., 2015), Thomson sampling for matrix factorisation (Kawale et al., 2015), and compositions objectives in vector space (Guu et al., 2015). Note that we re-formulate the compositions objectives in the probabilistially such that it can be used in an active setting; our approach for active learning is a generalisation of Thomson sampling from matrixes to tensors; AMDC find that reconstruction accuracy and recall cannot be achieved at the same time with strategies geared towards either exploit or reducing uncertainty, we show that the two objectives can be achieved at the same time with a properly designed exploration and exploitation scheme.

Table 1. The categorisation of factorisation problems with respect to three design considerations. The column headings are Bayesian(B)/Non-Bayesian(N) method, Passive(P)/Active(A) learning, and Matrix(M)/Tensor(T)/Compositional(C) structure. In this work, we tackle the problems denoted by an asterisk.

B/N	P/A	M/T/C	References
N	P	M	Lee & Seung (1999)
N	A	M	Ruchansky et al. (2015)
N	P	T	Nickel et al. (2011) Kolda & Bader (2009)
N	A	T	Kajino et al. (2015)
N	P	C	Neelakantan et al. (2015) Guu et al. (2015)
N	A	C	–
B	P	M	Mnih & Salakhutdinov (2007)
B	A	M	Kawale et al. (2015) Sutherland et al. (2013)
B	P	T	*, Xiong et al. (2010) Schmidt & Mohamed (2009)
B	A	T	*
B	P	C	*
B	A	C	*

3. Bayesian RESCAL

A relational knowledge base consists of a set triples in the form of (i, k, j) where i, j are entities, and k is a relation. A triple can be distinguished in a valid triple and invalid triple based on a semantic meaning of a triple. An example of valid triple in Freebase is (BarackObama, PresidentOf, U.S.), and an example of invalid triple is (BarackObama, PresidentOf, U.K.). A knowledge base can be represented in a three-way tensor $\mathcal{X} \in \{0, 1\}^{N \times K \times N}$, where K is a number of relations, N is a number of entities, and $x_{ikj} \in \mathcal{X}$ indicates whether the triple is valid.

We model the entities i as vectors e_i and the relations k as matrices R_k with an appropriately chosen latent dimension D. This follows a popular model for statistical relational learning, which is to factorise the tensor into a set of latent vector representations, such as the bilinear model RESCAL (Nickel et al., 2011). RESCAL aims to factorise each relational slice $X_{:k:}$ into a set of rank-D latent features

Table 2. Parameters for Gibbs updates. The posterior of e_i and R_k follows the normal distribution with mean μ and precision matrix Λ. $\otimes$ is the Kronecker product.

var	μ	Λ	ξ
e_i	$\frac{1}{\sigma_x^2}\Lambda_i^{-1}\xi_i$	$\frac{1}{\sigma_x^2}\sum_{jk:x_{ikj}\in\mathcal{X}^t}(R_k e_j)(R_k e_j)^\top$	$\sum_{jk:x_{ikj}\in\mathcal{X}^t} x_{ikj}R_k e_j + \sum_{jk:x_{jki}\in\mathcal{X}^t} x_{jki}R_k^\top e_j.$
$vec(R_k)$	$\frac{1}{\sigma_x^2}\Lambda_k^{-1}\xi_k$	$\frac{1}{\sigma_x^2}\sum_{ij:x_{ikj}\in\mathcal{X}^t}(e_i\otimes e_j)(e_i\otimes e_j)^\top + \frac{1}{\sigma_r^2}I_{D^2}$	$\sum_{ij:x_{ikj}\in\mathcal{X}^t} x_{ikj}(e_i\otimes e_j).$

as follows:

$$\mathcal{X}_{:k:} \approx E R_k E^\top, \qquad \text{for } k = 1, \ldots, K$$

Here, $E \in \mathbb{R}^{N\times D}$ contains the latent features of the entities $e_1, \ldots, e_N$ and $R_k \in \mathbb{R}^{D\times D}$ models the interaction of the latent features between entities in relation k.

We propose a probabilistic framework that directly generalises RESCAL by placing priors over the latent features. For each entity i, the latent feature of an entity $e_i \in \mathbb{R}^D$ is drawn from an isotropic multivariate-normal distribution.

$$e_i \sim N(\mathbf{0}, \sigma_e^2 I_D) \tag{1}$$

For each relation k, we draw matrix R_k from a zero-mean isotropic matrix normal distribution.

$$R_k \sim \mathcal{MN}_{D\times D}(\mathbf{0}, \sigma_r I_D, \sigma_r I_D) \tag{2}$$
$$\text{or equivalently } \; r_k = \text{vec}(R_k) \sim N(\mathbf{0}, \sigma_r^2 I_{D^2})$$

where $\text{vec}(R_k)$ denotes the flattening of the matrix.

We consider two models for x_{ikj}: a real or random variable. By placing a normal distribution over x_{ikj},

$$x_{ikj}|e_i, e_j, R_k \sim \mathcal{N}(e_i^\top R_k e_j, \sigma_x^2) \tag{3}$$

we can control the confidence on different observations through the variance parameter σ_x^2. The role of this parameter will be further discussed in the compositional model section.

We develop an efficient Gibbs sampler to perform inference for Bayesian RESCAL (BRESCAL). The key for achieving efficiency are the two conditional posteriors for latent features. The Gibbs updates are given by:

$$p(e_i|E_{-i}, \mathcal{R}, \mathcal{X}^t, \sigma_e, \sigma_x) = \mathcal{N}(e_i|\mu_i, \Lambda_i^{-1}) \tag{4}$$
$$p(R_k|E, \mathcal{X}, \sigma_r, \sigma_x) = \mathcal{N}(\text{vec}(R_k)|\mu_k, \Lambda_k^{-1}) \tag{5}$$

where the negative subscript $-i$ indicates the every other entity variables except e_i. The means and precision matrices are listed in Table 2, where we have used the identity $e_i^\top R_k e_j = r_k^\top e_i \otimes e_j$.

Alternatively, we may want to more closely model the fact that the observations are binary. Therefore we model x_{ikj} as a binomial distributed random variable whose probability is determined by logistic regression.

$$p(x_{ikj} = 1) = \sigma(e_i^\top R_k e_j),$$

where σ is a sigmoid function. We approximate the conditional posterior of E and R by Laplace approximation (Bishop, 2006). The maximum a posterior estimate of e_i or R_k given the rest can be computed through the standard logistic regression solvers with regularisation parameters. Given the maximum a posteriori parameters e_i^*, the posterior covariance S_i of entity i takes the form

$$S_i^{-1} = \sum_{x_{ikj}} \sigma(e_i^{*\top} R_k e_j)(1 - \sigma(e_i^{*\top} R_k e_j)) R_k e_j (R_k e_j)^\top$$
$$+ \sum_{x_{jki}} \sigma(e_j^\top R_k e_i^*)(1 - \sigma(e_j^\top R_k e_i^*)) R_k^\top e_i^* (R_k^\top e_i^*)^\top + I\sigma_e^{-1}.$$

The posterior covariance of R_k can be computed in the same way, and is shown in the appendix.

There are many advantages to a Bayesian view of tensor factorisation, such as the quantification of uncertainty by the predictive distribution, the ability to utilise priors, and the availability of principled model selection. We show in the empirical experiments that BRESCAL outperforms standard RESCAL. In the following, we focus on the predictive distribution, which enables us to improve sequential knowledge acquisition.

4. Particle Thompson Sampling

Thompson sampling has been gaining an increasing attention because of a competitive empirical performance as well as its conceptual simplicity (Scott, 2010; Chapelle & Li, 2011). Let $y_{1:t}$ be a sequence of rewards up to time t, and θ is an underlying parameter governing the rewards. With Thompson sampling, an agent choose action a according to its probability of being optimal:

$$\arg\max_a \int \mathbb{I}\left[\mathbb{E}(r|a,\theta) = \max_{a'} \mathbb{E}(r|a',\theta)\right] p(\theta|y_{1:t-1}) d\theta,$$

where $\mathbb{I}$ is an indicator function. Note that it is sufficient to draw a random sample from the posterior instead computing the integral.

We formulate Thompson sampling for knowledge base construction system as follows. First, we assume there are optimal latent features E^* and R^*, and the triples are generated through Equation 1– 3. At time t, the system draws samples E^t and R^t from the posterior distribution, and then chooses an optimal triple $(i,k,j)^* = \arg\max_{i,k,j} e_i^\top R_k e_j$

to be queried. Finally, with the newly observed triple $x_{(i,k,j)^*}$, the system updates the posterior of the latent features.

The main difficulty of applying Thompson sampling to this task is a sequential update of the posterior distribution of the latent features with new observations over time. Unlike the point estimation algorithms such as the maximum likelihood estimate, computing a full posterior with MCMC requires extensive computational cost. To make the algorithm feasible, we employ a sequential Monte-Carlo (SMC) method for online posterior inference, generalising an algorithm proposed in (Kawale et al., 2015) to tensors.

The SMC starts with H number of particles, each of which starts with likelihood weight $w_h = 1/H$, and a set of randomly sampled latent features E^{h_0} and $\mathcal{R}^{h_0}$. With a slight abuse of notation, let $\mathcal{X}^t$ be a set of observed triples up to time t. At time t, the system chooses one particle according to the particle weights, and then generates a new query via Thompson sampling from the selected particle. After observing a new variable, the system updates the posterior samples of every particle through the MCMC kernels with the new observation. We first sample the relation matrices using Equation 5, and sample the entity vectors using Equation 4. Under the mild assumption where $p(\Theta|\mathcal{X}^{t-1}) \approx p(\Theta|\mathcal{X}^t)$, $\Theta = \{E, \mathcal{R}\}$, the weight of each particle at time t can be computed as follows (Del Moral et al., 2006; Chopin, 2002):

$$w_h^t = \frac{p(\mathcal{X}^t|\Theta)}{p(\mathcal{X}^{t-1}|\Theta)} = p(x^t|\Theta, \mathcal{X}^{t-1}) \tag{6}$$

To keep the posterior samples on regions of high probability mass, we resample the particles whenever an effective sample size (ESS) is less than a predefined threshold. The ESS can be computed as $(\sum_h w_h^2)^{-1}$, and we set the threshold to $N/2$ (Doucet & Johansen, 2011). Resampling removes low weight particles with high probability, while keeping samples from the posterior. We summarise the particle Thompson sampling for BRESCAL with the Gaussian output variable in Algorithm 1.

We show that the Thompson sampling approach improves over passive BRESCAL in experiments with real and synthetic data. We also investigated the extension of the Rao-Blackwellisation approach as proposed in (Kawale et al., 2015), but we did not observe any significant performance improvements. We describe our extension in the appendix.

Algorithm 1 Particle Thompson sampling for Bayesian RESCAL with Gaussian output variable

Input: $\mathcal{X}^0, \sigma_x, \sigma_e, \sigma_r$.
for $t = 1, 2, \ldots$ **do**
  Thompson Sampling:
  $h_t \sim \text{Cat}(\mathbf{w}^{t-1})$
  $(i, k, j) \leftarrow \arg\max p(x_{ikj}|E^{h_{t-1}}, \mathcal{R}^{h_{t-1}})$
  Query (i, k, j) and observe x_{ikj}
  $\mathcal{X}^t \leftarrow \mathcal{X}^{t-1} \cup \{x_{ikj}\}$
  Particle Filtering:
  $\forall h, w_h^t \propto p(x_{ikj}|E^h, \mathcal{R}^h)$ ▷ Reweighting
  if $\text{ESS}(\mathbf{w}^t) \leq N$ **then**
    resample particles
    $w_h^t \leftarrow 1/H$
  end if
  for $h = 1$ **to** H **do**
    $\forall k, R_k^{h_t} \sim p(R_k|\mathcal{X}^t, E^{h_{t-1}}, \mathcal{R}_{-k})$ ▷ see Table (2)
    $\forall i, e_i^{h_t} \sim p(e_i|\mathcal{X}^t, E_{-i}, \mathcal{R}^{h_t})$ ▷ see Table (2)
  end for
end for

5. Compositional Relations

In this section, we propose a compositional relation model that exploit the compositional structure of knowledge graph to capture the latent semantic structure of the entities and relations. While previously suggested vector space models provide a statistical way to infer the latent semantic structure of entities and relations, but lack consideration of a graph structure of a knowledge base itself.

The compositionality represents a semantic meaning of a path over a knowledge graph that corresponds to a sequence of composable triples. For example, given two triples, "Barack Obama is a 44th president of U.S." (BarackObama, PresidentOf, U.S) and "Joe Biden was a running mate of Barack Obama" (JoeBiden, RunningMateOf, BarackObama), one can naturally deduce that the "Joe Biden is a vice president of U.S." (JoeBiden, VicePresidentOf, U.S.). Here the composition of two relations, president of, and running mate of, yield to a compositional relation, vice president of. More formally, if there is a sequence of triples where the target entity of a former triple is a source entity of a latter triple in a consecutive pair of triples in the sequence, then we can form a compositional triples as follows. Given the sequence of triples (i_1, k_1, j_1), (i_2, k_2, j_2), (i_2, k_2, j_2) $\ldots$ (i_n, k_n, j_n), where $i_k = j_{k+1}$ for all k, we form a compositional triple $(i_1, c(k_1, k_2, \ldots, k_n), j_n)$, where c denotes the compositional relation of the sequence of relations.

Let $\mathcal{C}^L$ be a set of all possible compositions of which length is up to L, $c \in \mathcal{C}$ be a sequence of relations, $c(i)$ be ith index of a relation in sequence c and $|c|$ be the length of the sequence. With set of compositions $\mathcal{C}^L$, we can expand set of observed triples $\mathcal{X}^t$ to set of compositional triples $\mathcal{X}^{\mathcal{C}^L(t)}$ in which compositional triple x_{icj} is an indicator variable that show the existence of the path from entity i to entity j through sequence of relations c in $\mathcal{X}^t$. Note that the compositional relation c is an abstract relation, and

there might be a multiple possible paths from i_1 to j_n.

With these extended compositional triples, we again model x_{icj} with a bilinear Gaussian distribution,

$$x_{(i,c(k_1,k_2),l)} \sim \mathcal{N}(e_i^\top R_{c(k_1,k_2)} e_j, \sigma_c^2), \tag{7}$$

where $R_{c(k_1,k_2)} \in \mathbb{R}^{D\times D}$ is a latent matrix of compositional relation c, and σ_c^2 is a covariance of the compositional triples. We keep the same latent vector e for each entity to model both normal triples and compositional triples. In the subsequent sections, we provide two different ways of modelling the compositional relation R_c.

5.1. Additive Compositionality

First, we define an additive compositional relation R_c as a sequence of normalized summation over relation matrices in composition c, i.e., $R_c = \frac{1}{|c|}(R_{c(1)} + R_{c(2)} + \cdots + R_{c(|c|)})$, then compositional triple x_{icj} is modeled as

$$\begin{aligned} x_{(i,c,j)} &\sim \mathcal{N}(e_i^\top R_c e_j, \sigma_c^2) \\ &= \mathcal{N}(e_i^\top \frac{1}{|c|}(R_{c(1)} + R_{c(2)} + \cdots + R_{c(|c|)})e_j, \sigma_c^2). \end{aligned} \tag{8}$$

The conditional distribution of e_i given $E_{-i}, \mathcal{R}, \mathcal{X}^t, \mathcal{X}^{L(t)}$ is expanded from the posterior of BRESCAL by incorporating compositional triples.

$$p(e_i|E_{-i}, \mathcal{R}, \mathcal{X}^t, \mathcal{X}^{L(t)}) = \mathcal{N}(e_i|\mu_i, \Lambda_i^{-1}). \tag{9}$$

To compute the conditional distribution of R_k, we first decompose R_c into two part where $R_c = \frac{1}{|c|}R_k + \frac{|c|-1}{|c|}R_{c/k}$, where $R_{c/k} = \sum_{k'\in c/k} R_{k'}$. The distribution of compositional triple is decomposed as follows:

$$x_{(i,c,l)} \sim \mathcal{N}(e_i^\top(\frac{1}{|c|}R_k + \frac{|c|-1}{|c|}R_{c/k})e_j, \sigma_c^2). \tag{10}$$

Then, the conditional distribution R_k given $R_{-k}, E, \mathcal{X}^t, \mathcal{X}^{L(t)}$ is

$$p(R_k|E, \mathcal{X}^t, \mathcal{X}^{L(t)}, \sigma_r, \sigma_x) = \mathcal{N}(\text{vec}(R_k)|\mu_k, \Lambda_k^{-1}). \tag{11}$$

The mean and precision are obtained by expanding out the sum across $\mathcal{X}^t$ and $\mathcal{X}^{L(t)}$. The details of the parameters are shown in the appendix.

5.2. Multiplicative Compositionality

Second, we define an multiplicative compositional relation R_c as a sequence of multiplication over relations in composition c, i.e. $R_c = R_{c(1)}R_{c(2)}\dots R_{c(|c|)}$, and the compositional triple as a bilinear Gaussian distribution with the compositional relation R_c,

$$x_{(i,c,j)} \sim \mathcal{N}(e_i^\top R_{c(1)}R_{c(2)}\dots R_{c(|c|-1)}R_{c(|c|)}e_j, \sigma_c^2) \tag{12}$$

The multiplicative compositionality can be understood as a sequence of linear transformation from the original entity i with the compositional relations, and the inner product between the transformed entity and target entity will form a value of the compositional triple.

Given a sequence of relations including relation k, R_k is placed in the middle of the compositional sequence, i.e., $e_i^\top R_{c(1)}R_{c(2)}\dots R_{c(\delta_k)}\dots R_{c(|c|-1)}R_{c(|c|)}e_j$, where δ_k is the index of relation k. For notational simplicity, we will denote the left side $e_i^\top R_{c(1)}R_{c(2)}\dots R_{c(\delta_k-1)}$ as $\bar{e}^\top_{ic(:\delta_k)}$, and the right side $R_{c(\delta_k+1)}\dots R_{c(|c|-1)}R_{c(|c|)}e_j$ as $\bar{e}_{ic(\delta_k:)}$, therefore we can rewrite the mean parameter as $\bar{e}^\top_{ic(:\delta_k)}R_k\bar{e}_{ic(\delta_k:)}$. With the simplified notations, the conditional of R_k is

$$p(R_k|E, \mathcal{X}, \sigma_r, \sigma_x) = \mathcal{N}(\text{vec}(R_k)|\mu_k, \Lambda_k^{-1}), \tag{13}$$

where the mean and precision are obtained by expanding out the product across $\mathcal{X}^t$ and $\mathcal{X}^{L(t)}$. The details of the parameters are shown in the appendix. The conditional distribution of e_i given the rest is the same as Equation 9.

As the length of sequence c increases, a small error in the first few multiplication will result a large differences in the final compositional relation. One way to mitigate the cascading error is to increase variance of compositional triples σ_c as the length of the sequence increases.

With the conditional distributions, both compositional models can use the same particle Thompson sampling scheme described in Algorithm 1. However, the model can only query the triples in the original tensor and not in the expanded tensor because the triples are not observable.

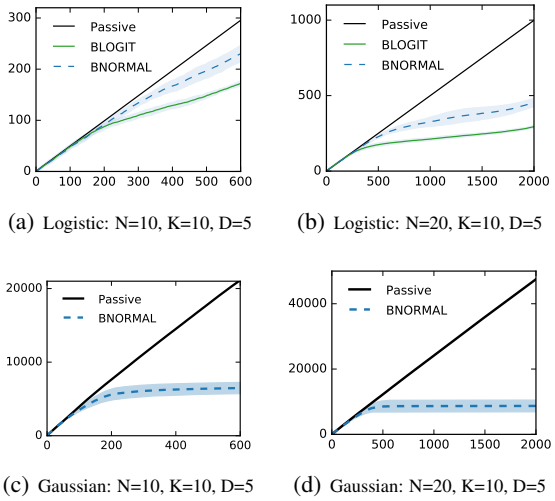
6. Experiments

In this section, we show results for Thomson sampling and compositions for BRESCAL. First on two synthetic datasets, and then on three common benchmarks for knowledge graphs.

6.1. Thompson Sampling on synthetic data

We first synthesise two datasets following the model assumptions in Eq. 1 to 3. First, entities and relations are generated from zero-mean isotropic multivariate normal distribution, with variance parameters $\sigma_e = 1, \sigma_r = 1$, respectively. We generate two sets of output triples, with the logistic output and the Gaussian with σ_x set to 0.1, respectively (Sec 3).

To measure performance, we compute cumulative regret at each time n as $R(n) = \sum_{t=1}^n x_t - x_t^*$, where x_t^* is the highest-valued triple among triples that have not been chosen up to time t. Unlike the general bandit setting where one can select a single item multiple times, in our formula-



(a) Logistic: N=10, K=10, D=5 (b) Logistic: N=20, K=10, D=5

(c) Gaussian: N=10, K=10, D=5 (d) Gaussian: N=20, K=10, D=5

Figure 1. Cumulative regret of particle Thompson sampling with Gaussian and logistic output (BNORMAL, BLOGIT) against Passive learning on synthetic datasets with logistic (top row, a, b) and Gaussian (bottom row, c, d) output variables. The averaged cumulative regrets over 10 runs are plotted with one standard error. As the model obtained more and more labeled samples from Thompson sampling, the cumulative regrets increase sub-linearly.

tion, we can select one triple only once. So after selecting a triple at time t, the selected triple will be removed from a set of candidate triples.

Figure 1 shows the cumulative regret of the algorithm on the synthetic data with varying size of entities and relations. We compare the cumulative regret of the particle Thompson sampling with the passive learning method where the model choose a random triple at each time. All results are averaged over 10 individual runs with different initialisations. Note that the dataset with binary logistic output variables can be used to train both Bayesian logistic RESCAL (BLOGIT) and Bayesian Gaussian RESCAL (BNORMAL) whereas the dataset with the Gaussian output can only be trained by BNORMAL. Figure 1(a) and 1(b) show that with the logistic synthetic dataset both models are capable to learn the latent features of the generated triples, with logistic outperforming the Gaussian; Figure 1(c) and 1(d) show that BNORMAL outperform passive learning in the real valued dataset.

6.2. Thompson sampling for compositional models on synthetic data

We conduct a second experiment on synthetic dataset to understand how the Thompson sampling works for the compositional data. As in the first experiment, we first generate entities and relations from zero-mean multivariate normal with variance parameter $\sigma_e = 1$ and $\sigma_r = 1$. We generate

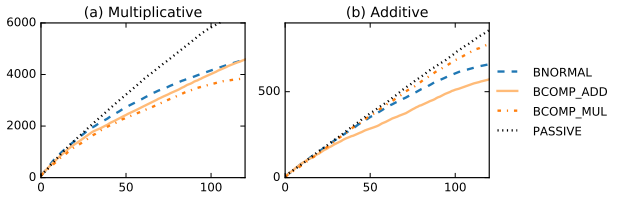


Figure 2. Cumulative regret of particle Thompson sampling of the compositional models on synthetic dataset with N=5, D=5. The synthetic dataset has three relations (K=3); the first two are independently generated, and the third relation is composed by the first two relations. The dataset used in (a) is generated by the multiplicative assumption, and the dataset used in (b) is generated by the additive assumption.

a set of triples with Gaussian output as in Eq. 3. We then synthesise two sets of expanded tensors using the previously used entities and relations based on the multiplicative and additive compositional assumptions, defined in Sec 5, respectively. So we synthesise fully observable expanded tensor $\mathcal{X}^L$ where $L = 2$. We set both variance parameter σ_x and σ_c to 0.1. Note that in real world situation, the expanded tensor can be only constructed through the observed triples, and the triples in the expanded tensor cannot be queried.

To run the particle Thompson sampling on the synthetic dataset, we let the compositional models know which relation is composed by other relations. The non-compositional BNORMAL model assumes each relation is independent to one another. Therefore, the compositional model uses much less number of parameters to model the same size of tensor to compare with the non-compositional model. With this fully observable expanded tensors, we run the Thompson sampling of the compositional models. Figure 2 shows the cumulative regrets on synthetic datasets. The multiplicative and additive compositionality are used to generate the dataset for Figure 2(a) and 2(b), respectively. The results correspond to our assumption: the multiplicative compositional model (BCOMP-MUL) shows lower regrets on the multiplicative data in Figure 2(a), and the additive compositional model (BCOMP-ADD) shows lower regrets on the additive compositional data in Figure 2(b), and both have lower regrets than passive learning or BNORMAL with no composition.

6.3. Training compositional model on real datasets

We evaluate the compositional models on three benchmark datasets and compare the performance to various baseline algorithms. We use three relational datasets: KINSHIP, UMLS, and NATION. Detailed description of each dataset is shown in Table 3 [1].

[1] https://alchemy.cs.washington.edu/papers/kok07/

Table 3. Description of datasets. Sparsity denotes the ration of valid triples to invalid triples.

Dataset	# rel	# entities	# triples	sparsity
Kinship	26	104	10,790	0.038
UMLS	49	135	6,752	0.008
Nation	56	14	2,024	0.184

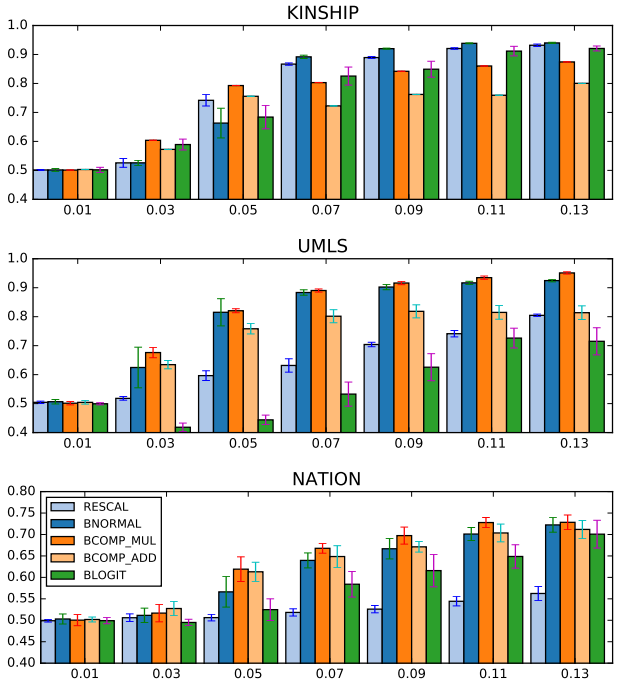


Figure 3. ROC-AUC scores of compositional models. The x-axis denotes the proportion of an observed triples including negative triples used for training models.

We first evaluate our model in a non-active setting, to measure the performance of BRESCAL with all non-compositional and compositional variants.

For all experiments, we set the compositional length L to 2, split the dataset into 20% for validation and 30% for testing. We vary the proportion of training triples from 1% to 13% of datasets. For RESCAL, we use the authors' implementation, and measure performance over 10 runs with random initialisations. For BRESCAL and all the variants, we sample triples x_{ikj} from its posterior, and measure performance over 10 different samples.

Figure 3 shows the ROC-AUC scores of the compositional models with the various baseline models. We can see that BNORMAL or BLOGIT generally outperform RESCAL. We compare the compositional model with original RESCAL, BNORMAL, and BLOGIT. In general, BCOMP-MUL outperforms BCOMP-ADD, and performs better the other baseline models when the proportion of training set is small. For UMLS and NATION, BCOMP-MUL outperforms across the all training proportions. For KINSHIP, however, the model performs better when the training proportion is less than 7%.

6.4. Thompson sampling on real datasets

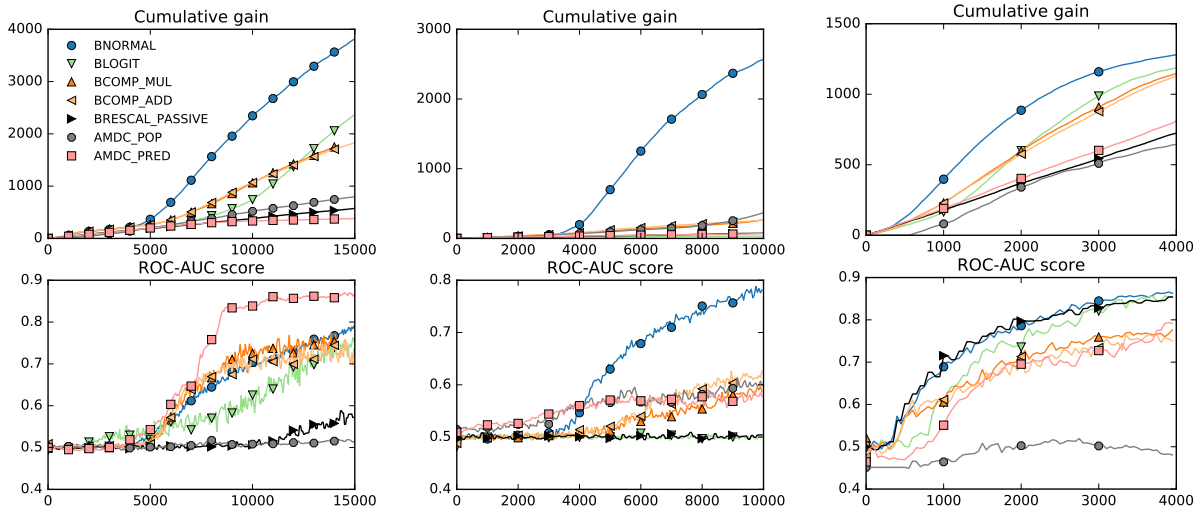
Next, we evaluate particle Thompson sampling for both compositional and non-compositional models on real datasets.

Experimental settings: We compare our model with AMDC, and passive learning with BRESCAL. AMDC model has been proposed to achieve two different active learning goals, constructing a predictive model and maximising the valid triples in a knowledge base, with two different querying strategies (Kajino et al., 2015). AMDC-PRED is a predictive model construction strategy and chooses a triple which is the most ambiguous (close to the decision boundary) at each time t. AMDC-POP is a population strategy which aims to maximise the number of valid triples in a knowledge base, choosing a triple with the highest expected value at each time. To train all models we only use the observed triples up to the current time. For the passive learning with BRESCAL, we generate a random sample at each time period. For the particle Thompson sampling models, we set variance parameter σ_e and σ_r to 1, σ_x to 0.1, and vary σ_c from 1 to 100.

We leave 30 % of triples as a test set to measure test error. At each time period, each model choose one triples to query, if the selected triple is in the test set then we choose the next highest expected triple which is not in the test set. All models start from zero observation. After every querying, the model obtains a label of the queried triples from an oracle, then the model updates the parameters.

Evaluation metric: We use two different evaluation metrics, ROC-AUC score, and cumulative gain, for the performance comparison. One goal of the Thompson sampling is to maximise the knowledge acquisition through the balanced querying strategy between exploration and exploitation. To measure how many triples are obtained through the querying stage, we fist compute the cumulative gain which is the number of valid triple obtained up to time t, and then compute the ROC-AUC score on test set to understand how this balanced querying strategy results in making a predictive model.

Exploitation and exploration: Figure 4 shows the cumulative gain and ROC-AUC scores of the Thompson sampling on three real datasets. BNORMAL performs better than other baseline models for the cumulative gain, and shows comparable result for the ROC-AUC scores. Both compositional models perform worse than BNOR-



(a) KINSHIP (b) UMLS (c) NATION

Figure 4. The cumulative gain and ROC-AUC score of the Thompson sampling with various baseline approaches. Thompson sampling with Bayesian RESCAL (BNORMAL) model achieves the highest cumulative gain to compare with other active and passive learning algorithms and shows comparable performance on ROC-AUC scores.

MAL across all datasets.

In the original AMDC work (Kajino et al., 2015), AMDC-POP model obtains more valid triples than AMDC-PRED, and AMDC-PRED shows high ROC-AUC scores than AMDC-POP. In our experiment, however, AMDC-POP shows comparable cumulative gain to AMDC-PRED and even worse than AMDC-PRED for the UMLS. We conjecture the initial observation results in the different performances: in the original experiment, the model starts from a small set of training data so this gives the model focusing on exploit and advantage, whereas in our experiment, we start from zero observation which makes the model hard to exploit the structure. This result shows the importance of balancing between exploitation and exploration.

We note that the compositional model performs worse than the model without composition in this active setting. One possible explanation is that the naive particle Thompson sampling may not capture the posterior of the complex compositional structure, so the particle degenerates over time. Recent advances in sequential Monte Carlo may help to solve the problem(Gu et al., 2015; Naesseth et al., 2014; Lindsten et al., 2014). We leave this for future work.

7. Discussion

We have proposed a novel compositional relational model with uncertainty and presented the Thompson sampling for both compositional and non-compositional models to solve the active knowledge acquisition problem. The compositional model aims to infer the latent features of knowledge bases by incorporating an additional graph structure. In the passive learning scenario, the compositional model outperforms the other models, especially, when training size is relatively small. In the active learning scenario, Bayesian RESCAL achieves the highest cumulative gain across all datasets. Again, this result emphasise the importance of being balanced between exploration and exploitation.

Previous work such as the one by Kajino et al. (2015) views knowledge population and predictive model construction are separate problems. We find this observation true when the algorithm has a warm-start, i.e. already having a fair amount of data before active learning starts; when the information is sparse, the same strategy works for both maximizing recall and reducing uncertainty. Thompson sampling has been studied in the context of multi-armed bandit problems where the goal is to maximise cumulative gains or minimise cumulative regrets over time, whereas its performance on making a predictive model has not been widely discussed so far. Its performance on building a generalisable model was unclear. Throughout this work, we have empirically shown that maximising cumulative gain entails the predictive models as well. In the long run, we see this work as a promising step towards using a composition-aware knowledge completion system to connect with the knowledge extraction problem (Dong et al., 2014).

References

Bishop, Christopher M. Pattern recognition. *Machine Learning*, 2006.

Chapelle, Olivier and Li, Lihong. An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems*, 2011.

Chopin, Nicolas. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.

Del Moral, Pierre, Doucet, Arnaud, and Jasra, Ajay. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (3):411–436, 2006.

Dong, Xin, Gabrilovich, Evgeniy, Heitz, Geremy, Horn, Wilko, Lao, Ni, Murphy, Kevin, Strohmann, Thomas, Sun, Shaohua, and Zhang, Wei. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 601–610. ACM, 2014.

Doucet, Arnaud and Johansen, Am. A tutorial on particle filtering and smoothing: fifteen years later. *Handbook of Nonlinear Filtering*, (December):656–704, 2011. ISSN 01677152. doi: 10.1.1.157.772.

Gu, Shixiang, Ghahramani, Zoubin, and Turner, Richard E. Neural adaptive sequential monte carlo. In *Advances in Neural Information Processing Systems*, pp. 2611–2619, 2015.

Guu, Kelvin, Miller, John, and Liang, Percy. Traversing knowledge graphs in vector space. In *Proceedings of Empirical Methods in Natural Language Processing*, 2015.

Kajino, Hiroshi, Kishimoto, Akihiro, Botea, Adi, Daly, Elizabeth, and Kotoulas, Spyros. Active learning for multi-relational data construction. In *Proceedings of International Conference on World Wide Web*, pp. 560–569, 2015.

Kawale, Jaya, Bui, Hung H, Kveton, Branislav, Tran-Thanh, Long, and Chawla, Sanjay. Efficient thompson sampling for onlinematrix-factorization recommendation. In *Advances in Neural Information Processing Systems*, pp. 1297–1305, 2015.

Kolda, Tamara G and Bader, Brett W. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Lao, N and Cohen, WW. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 2010.

Lee, Daniel D and Seung, H Sebastian. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Lindsten, Fredrik, Johansen, Adam M, Naesseth, Christian A, Kirkpatrick, Bonnie, Schön, Thomas B, Aston, John, and Bouchard-Côté, Alexandre. Divide-and-conquer with sequential monte carlo. *arXiv preprint arXiv:1406.4993*, 2014.

Mnih, Andriy and Salakhutdinov, Ruslan. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 1257–1264, 2007.

Naesseth, Christian Andersson, Lindsten, Fredrik, and Schön, Thomas B. Sequential monte carlo for graphical models. In *Advances in Neural Information Processing Systems*, pp. 1862–1870, 2014.

Neelakantan, Arvind, Roth, Benjamin, and McCallum, Andrew. Compositional Vector Space Models for Knowledge Base Completion. In *Proceedings of Association for Computational Linguistics*, 2015.

Nickel, Maximilian, Tresp, Volker, and Kriegel, Hans-Peter. A three-way model for collective learning on multi-relational data. In *Proceedings of International Conference on Machine Learning*, pp. 809–816, 2011.

Nickel, Maximilian, Murphy, Kevin, Tresp, Volker, and Gabrilovich, Evgeniy. A review of relational machine learning for knowledge graphs: From multi-relational link prediction to automated knowledge graph construction. *arXiv preprint arXiv:1503.00759*, 2015.

Ruchansky, Natali, Crovella, Mark, and Terzi, Evimaria. Matrix completion with queries. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1025–1034. ACM, 2015.

Schmidt, Mikkel N and Mohamed, Shakir. Probabilistic non-negative tensor factorization using markov chain monte carlo. In *Signal Processing Conference, 2009 17th European*, pp. 1918–1922. IEEE, 2009.

Scott, Steven L. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26:639–658, 2010.

Sutherland, Dougal J, Póczos, Barnabás, and Schneider, Jeff. Active learning and search on low-rank matrices. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 212–220. ACM, 2013.

West, Robert, Gabrilovich, Evgeniy, Murphy, Kevin, Sun, Shaohua, Gupta, Rahul, and Lin, Dekang. Knowledge

base completion via search-based question answering. In *Proceedings of International Conference on World Wide Web*, pp. 515–526. ACM, 2014.

Xiong, Liang, Chen, Xi, Huang, Tzu-Kuo, Schneider, Jeff G, and Carbonell, Jaime G. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of SIAM International Conference on Data Mining*, volume 10, pp. 211–222. SIAM, 2010.