Probabilistic Knowledge Graph Construction: Compositional and Incremental Approaches

ABSTRACT

A knowledge base construction consists of a two-step approach; extracting information from external sources, known as knowledge population, and inferring missing information through a statistical analysis on the extracted information, known as knowledge completion. In many cases, however, there is not enough external sources to extract information. An incremental knowledge population via labelling of human experts can help to reduce the gap between two processes. In this paper, we propose a new probabilistic knowledge base factorisation method that benefits from the path structure of existing knowledge (e.g. syllogism). This explicit probabilistic formulation enable us to develop an incremental population model based on exploitation-exploration strategies. We demonstrate that the factorisation model with the path structure performs better on the knowledge completion task. Whereas the model without the path structure performs better in the incremental population. The result leads to a counter-intuitive conclusion; a better predictive model does not guarantee to have a better performance in incremental population. An additional experiment explains the degeneracy under the model uncertainty.

1. INTRODUCTION

Relational knowledge graphs structuralise our understanding about the world and help us reason and infer in a wide range of tasks such as information retrieval, question answering, and semantic parsing [6, 11, 14]. A construction of a knowledge graph is a very active research area with many important and challenging research questions. The early stage of knowledge graph construction relies on **knowledge population** task where the goal is to maximise the number of facts in the form of `(entity1, relation, entity2)` triples. External sources such as Wikipedia are used to extract the triples [10], or human experts encode a prior knowledge manually [2]. Despite the endeavour toward to construct a complete knowledge graph, even the largest commercialised knowledge graph is still far from complete [7]. **Knowledge completion** task has emerged as a complement of the knowledge population to scale up the knowledge graph construction. Unlike the knowledge population task, the goal of knowledge completion is to maximise the prediction performance on unseen triples. Several statistical relational models has been proposed to infer missing facts from known facts [16, 21].

One major obstacle of knowledge graph construction is a gap between knowledge population and completion. For example, when there is no external source to extract knowledge, the construction relies solely on the contribution of human experts. Manual labelling is often slow and costly. Therefore, we need a systematic way of selecting triples that will be labelled, in order to maximise the performance of the following knowledge completion. Recently, Kajino et al. propose an active learning strategy for populating knowledge graph [12]; however the algorithm have problems simultaneously achieving both construction goals.

We provide statistical relational models, which can be used for the active query selection. We first reformulate a bilinear tensor factorisation [20] in a probabilistic way, where entities and relations are embedded into a latent feature space. And then we propose a tensor factorisation model that incorporates the graph structure of knowledge graph into the factorisation. These two probabilistic models provide a natural way of exploit uncertainty of triples that is crucial to develop an active triple selection for the incremental knowledge population. We employ a Thompson sampling [24] – an approach for solving the multi-armed bandit problem, to find an optimal trade-off between exploration and exploitation during active selection.

Based on experiments with three commonly used benchmark datasets, we find that the additional graph structure helps predict unobserved triples, while the model without graph structure is more helpful in the incremental knowledge population task. Because, for the knowledge completion, it is important to find a better latent structure, for the incremental population, however, it is more important to measure an uncertainty of a model in order to explore the latent space efficiently over time. As far as we know, this is the first study that explicitly reveals how the knowledge completion models result in different conclusions in the incremental knowledge population.

The contributions of this paper can be summarised as follows:

- We propose a probabilistic reformulation of bilinear tensor factorisation that allows us to predict and measure the uncertainty of unobserved triples (Section 3).

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

- We incorporate a graph path structure of a knowledge graph into the proposed factorisation by modelling a composition of relations as an algebraic operation in the probabilistic embedding space (Section 4).

- We propose an incremental knowledge population method that efficiently explores the factorised space, guided by a principled way of exploration-exploitation via Thompson sampling algorithm (Section 5).

- Experiments on the knowledge completion with three real-world dataset show that the compositional model predicts unseen triples better than the bilinear factorisation model (Section 6).

- Experiments show the importance of uncertainty in the incremental population. Especially, the better predictive model does not guarantee to have a better knowledge population due to an improper uncertainty measure (Section 7).

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.

Probabilistic/Non-Probabilistic This refers to two broad classes of model formulation, whether an obtained model has a probabilistic interpretation. The probabilistic models are capable of placing priors and measuring uncertainty.

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 labelled.

Matrix/Tensor/Composition Relational learning problems can operate on different data representations. Matrix representation is common when a dataset can be represented as a bi-partite graph, such as (user, item) tuples in the recommender systems setting. Tensor representation is handy when edges in the graph have 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-Pr, A, C, or Non-Probabilistic Active Composition model, can be done by simply using the query strategies from [12] on the compositional model by [9]. Our work address a critical gap in probabilistic 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 [12], Thomson sampling for matrix factorisation [13], and compositional objectives in vector space [9]. Note that we reformulate the compositional objectives probabilistically 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 knowledge population 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-exploitation scheme.

Table 1: The categorisation of factorisation problems with respect to three design considerations. The column headings are Probabilistic(Pr)/Non-Probabilistic(N-Pr) 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.

Pr/N-Pr	P/A	M/T/C	References
N-Pr	P	M	[17]
N-Pr	A	M	[22]
N-Pr	P	T	[21][15]
N-Pr	A	T	[12]
N-Pr	P	C	[19][9]
N-Pr	A	C	–
Pr	P	M	[18]
Pr	A	M	[13][25]
Pr	P	T	*, [27][23]
Pr	A	T	*
Pr	P	C	*
Pr	A	C	*

3. PROBABILISTIC RESCAL

A relational knowledge graph 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 the triple. An example of the valid triple in Freebase is `(BarackObama, PresidentOf, U.S.)`, and an example of the invalid triple is `(BarackObama, PresidentOf, U.K.)`. A knowledge graph can be represented in a three-way binary 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 $x_{ikj} \in \{0,1\}$ indicates whether the triple is valid.

We model the entity i as vectors e_i and the relation k as matrix 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 [21]. RESCAL aims to factorise each relational slice $X_{:k:}$ into a set of rank-D latent features 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

Table 2: Parameters for Gibbs updates. The conditional 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	$\frac{1}{\sigma_x^2}\Lambda_i^{-1}\xi_i$	$\frac{1}{\sigma_x^2}\sum_{jk:x_{ikj}\in\mathcal{X}^t}(R_k e_j)(R_k e_j)^\top$	$\sum_{jk:x_{ikj}\in\mathcal{X}^t} x_{ikj}R_k e_j + \sum_{jk:x_{jki}\in\mathcal{X}^t} x_{jki}R_k^\top e_j.$
$vec(R_k)$	$\frac{1}{\sigma_x^2}\Lambda_k^{-1}\xi_k$	$\frac{1}{\sigma_x^2}\sum_{ij:x_{ikj}\in\mathcal{X}^t}(e_i\otimes e_j)(e_i\otimes e_j)^\top + \frac{1}{\sigma_r^2}I_{D^2}$	$\sum_{ij:x_{ikj}\in\mathcal{X}^t} x_{ikj}(e_i\otimes e_j).$

isotropic matrix normal distribution.

$$R_k \sim \mathcal{MN}_{D\times D}(\mathbf{0}, \sigma_r I_D, \sigma_r I_D) \quad (2)$$

$$\text{or equivalently } r_k = \text{vec}(R_k) \sim N(\mathbf{0}, \sigma_r^2 I_{D^2})$$

where vec(R_k) denotes the flattening of the matrix.

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

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

we model the value of triple as a real variable. This is not a natural choice since the triple is a binary variable, however, we can control the confidence on different observations through the variance parameter σ_x^2.

We develop a Gibbs sampler to perform the posterior inference for the probabilistic RESCAL (PRESCAL). The conditional distribution of each latent variable is given by:

$$p(e_i|E_{-i}, \mathcal{R}, \mathcal{X}^t, \sigma_e, \sigma_x) = \mathcal{N}(e_i|\mu_i, \Lambda_i^{-1}) \quad (4)$$

$$p(R_k|E, \mathcal{X}, \sigma_r, \sigma_x) = \mathcal{N}(\text{vec}(R_k)|\mu_k, \Lambda_k^{-1}) \quad (5)$$

where the negative subscript $-i$ indicates the every other entity variables except entity i. Exact forms of the posterior 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 model the binary observation more precisely. Therefore we model x_{ikj} as a binomial 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 [1]. The maximum a posterior estimate of e_i or R_k given the rest can be computed through standard logistic regression solvers with the priors over e_i and R_k as 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. Let R_k^* is a maximum a posterior solution of R_k given E. Then, the conditional posterior covariance S_k of relation k has the form of:

$$S_k^{-1} = \sum_{x_{ikj}} \sigma(e_i^\top R_k^* e_j)(1 - \sigma(e_i^\top R_k^* e_j))\bar{e}_{ij}\bar{e}_{ij}^\top + I\sigma_r^{-1},$$

where $\bar{e}_{ij} = e_i \otimes e_j$.

The probabilistic view of tensor factorisation has many advantages 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 PRESCAL outperforms standard RESCAL.

4. COMPOSITIONAL RELATIONS

In this section, we propose a compositional relation model that exploits 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 graph 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_3, k_3, j_3) $\ldots$ (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 entity i_1 to j_n.

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

$$x_{(i,c(k_1,k_2),l)} \sim \mathcal{N}(e_i^\top R_{c(k_1,k_2)} e_j, \sigma_c^2), \quad (6)$$

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 non-compositional triples and compositional triples. In the subsequent sections, we provide two different ways of modelling the compositional relation R_c.

4.1 Additive Compositionality

With the compositions of relations, the PRESCAL may place a different relation matrix R_c for each composition

c. However, the number of required matrices increases exponentially, as the length of composition increases linearly. Consequently, the computational cost will also increases exponentially. To limit the required number of parameters, we propose two different ways of modelling the compositional relation R_c. First, we define an additive compositional relation R_c as a sequence of normalised 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 modelled as

$$\begin{aligned} x_{(i,c,j)} &\sim \mathcal{N}(e_i^\top R_c e_j, \sigma_c^2) \qquad (7) \\ &= \mathcal{N}(e_i^\top \frac{1}{|c|}(R_{c(1)} + R_{c(2)} + \cdots + R_{c(|c|)})e_j, \sigma_c^2). \end{aligned}$$

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

$$p(e_i|E_{-i}, \mathcal{R}, \mathcal{X}^t, \mathcal{X}^{L(t)}) = \mathcal{N}(e_i|\mu_i, \Lambda_i^{-1}). \qquad (8)$$

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:

$$x_{(i,c,l)} \sim \mathcal{N}(e_i^\top (\frac{1}{|c|}R_k + \frac{|c|-1}{|c|}R_{c/k})e_j, \sigma_c^2). \qquad (9)$$

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

$$p(R_k|E, \mathcal{X}^t, \mathcal{X}^{L(t)}, \sigma_r, \sigma_x) = \mathcal{N}(\text{vec}(R_k)|\mu_k, \Lambda_k^{-1}). \qquad (10)$$

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.

4.2 Multiplicative Compositionality

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

$$x_{(i,c,j)} \sim \mathcal{N}(e_i^\top R_{c(1)}R_{c(2)} \ldots R_{c(|c|-1)}R_{c(|c|)}e_j, \sigma_c^2) \qquad (11)$$

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 forms 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., $e_i^\top R_{c(1)}R_{c(2)} \ldots R_{c(\delta_k)} \ldots R_{c(|c|-1)}R_{c(|c|)}e_j$, where δ_k is the index of relation k. For notational simplicity, we will denote the left side $e_i^\top R_{c(1)}R_{c(2)} \ldots R_{c(\delta_k - 1)}$ as $\bar{e}_{ic(:\delta_k)}^\top$, and the right side $R_{c(\delta_k+1)} \ldots R_{c(|c|-1)}R_{c(|c|)}e_j$ as $\bar{e}_{ic(\delta_k:)}$. With this notation, we can rewrite the mean parameter as $\bar{e}_{ic(:\delta_k)}^\top R_k \bar{e}_{ic(\delta_k:)}$, and the conditional of R_k as

$$p(R_k|E, \mathcal{X}, \sigma_r, \sigma_x) = \mathcal{N}(\text{vec}(R_k)|\mu_k, \Lambda_k^{-1}), \qquad (12)$$

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 8.

Algorithm 1 Particle Thompson sampling for probabilistic 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:
    $\forall h, w_h^t \propto p(x_{ikj}|E^h, \mathcal{R}^h)$                ▷ Reweighing
    if 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
```

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.

5. PARTICLE THOMPSON SAMPLING

We now introduce a particle Thompson sampling algorithm for the incremental knowledge population with the proposed PRESCAL models. In the incremental population task, a knowledge base starts with zero or initial observations, and at each time period, we select one triple to be queried and labelled by an external system, i.e. human experts. The queried triple is chosen selectively based on past observations. Each labelling requires a certain amount of expense, so the goal is to obtain as many valid triples as possible given a limited budget.

Thompson sampling provides a model based query selection process, and has been gaining an increasing attention because of a competitive empirical performance as well as its conceptual simplicity [3, 24]. 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 incremental knowledge population 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

Table 3: Description of datasets. Sparsity denotes the ratio 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

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 estimator, 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 [13] 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 [5, 4]:

$$w_h^t = \frac{p(\mathcal{X}^t|\Theta)}{p(\mathcal{X}^{t-1}|\Theta)} = p(x^t|\Theta, \mathcal{X}^{t-1}) \tag{13}$$

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 $N/2$ [8]. Resampling removes low weight particles with high probability, while keeping samples from the posterior. We summarise the particle Thompson sampling for PRESCAL with the Gaussian output variable in Algorithm 1.

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

We show that the Thompson sampling approach improves over passive PRESCAL in experiments with real and synthetic data. We also investigated the extension of the Rao-Blackwellisation approach as proposed in [13], but we did not observe any significant performance improvements. We describe our extension in the appendix.

6. EXPERIMENTS ON KNOWLEDGE COMPLETION

We first evaluate our model for the knowledge completion task to measure the performance of PRESCAL with all non compositional and compositional variants.

We evaluate the PRESCAL 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].

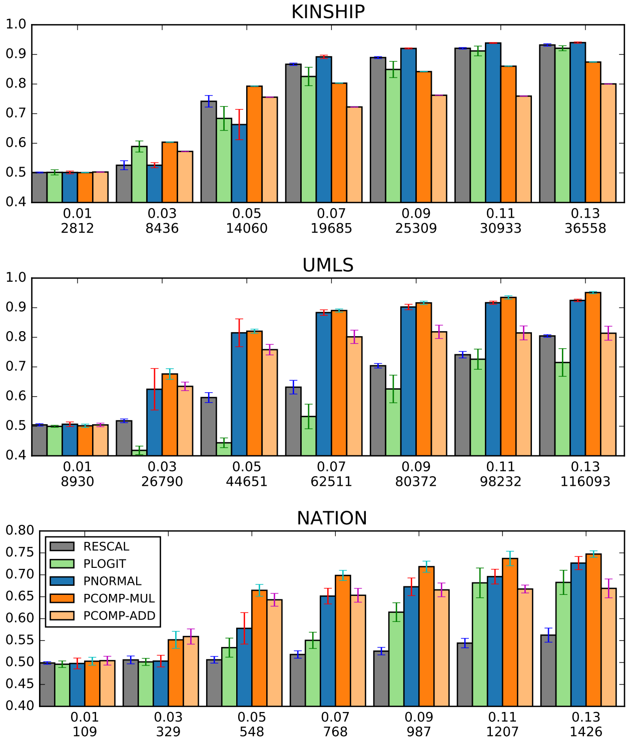


Figure 1: ROC-AUC scores of compositional models. The x-axis denotes the proportion and total number of triples used for training.

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[2], and measure performance over 10 runs with random initialisations. For PRESCAL and all the variants, we sample triples x_{ikj} from its posterior, and measure performance over 10 different samples. Given the test set, the performances of models are measured by ROC-AUC score:

$$\frac{1}{|\mathcal{X}_p||\mathcal{X}_n|} \sum_{\{i,k,j\} \in \mathcal{X}_p, \{i',k',j'\} \in \mathcal{X}_n} \mathbb{I}[\bar{x}_{ikj} > \bar{x}_{ikj}], \tag{14}$$

where $\mathcal{X}_p$ and $\mathcal{X}_n$ are the set of positive and negative triples in the test set, respectively, and $\bar{x}$ is a reconstructed triple.

Figure 1 shows the ROC-AUC scores of the compositional models with the various baseline models. We can see that the PRESCAL with the normal output (PNORMAL) or logistic output (PLOGIT) generally outperform RESCAL. We compare the compositional model with original RESCAL, PNORMAL, and PLOGIT. In general, the multiplicative compositional model (PCOMP-MUL) outperforms the additive compositional model (PCOMP-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

[1] https://alchemy.cs.washington.edu/papers/kok07/
[2] https://github.com/mnick/rescal.py

Table 4: Example of path prediction from UMLS data. We predict top 5 entities in compositional triples starting from entity `Mental-or-Behavioral (MB) Dysfunction` followed by two relations `Affects` and `Produces`. Correct entities are bolded.

(a) Triple prediction: `(MB Dysfunction, Affects, -)`

Model	Top 1	Top 2	Top 3	Top 4	Top 5
PNORMAL	**Invertebrate**	**Reptile**	**Archaeon**	**Bird**	**Phy.-Function**
PLOGIT	**Cell-Function**	**Disease-or-Syndrome**	**Cell-or-Molecular-Dysf.**	**Exp.-Model-of-Disease**	**Mental-Process**
PCOMP-MUL	**Archaeon**	**Fish**	**Fungus**	**Invertebrate**	**Human**
PCOMP-ADD	**Path.-Function**	**Bird**	**Cell-or-Molecular-Dysf.**	Drug-Delivery-Device	Congenital-Abnormality

(b) Length-2 path prediction: `(MB Dysfunction, Affects, Produces, -)`

Model	Top 1	Top 2	Top 3	Top 4	Top 5
PNORMAL	Clinical-Drug	Sign-or-Symptom	Org.-Attribute	Drug-Delivery-Device	Clinical-Attr.
PLOGIT	Amphibian	Gov.-or-Reg.-Activity	Food	Biologic-Func.	Classification
PCOMP-MUL	**Enzyme**	**Body-Substance**	**Biogenic-Amine**	Carbohydrate	**Immunologic-Factor**
PCOMP-ADD	**Immunologic-Factor**	**Body-Substance**	Molecular-Biology-Research-Technique	Clinical-Drug	Chemical-Viewed-Structurally

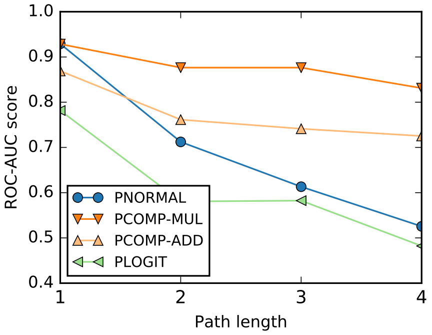


Figure 2: Path prediction result with UMLS. The performances of both compositional models remain consistent whereas those of the non-compositional models drop sharply as the length increases.

KINSHIP, however, the model performs better when the training proportion is less than 7%.

The goal of the compositional models is to factorise triples along with the graph structure as a whole. The triple prediction task tells us the trained model is capable of triple prediction, but does not tell whether the features containing the graph structure. If the model factorise the graph structure properly, then the trained model can predict not only triples but the graph structure as well. To validate the model assumption, we evaluate a path prediction task. For this task, we use 10% of UMLS dataset for training. We compute the expected value of unobserved path given a trained model. The non-compositional models are not capable to compute the expected value. In such case, we approximate paths with multiplicative model assumption in Equation 11. We vary the path length from 1 (triple) to 4, and measure ROC-AUC scores on the reconstructed compositional triples. Figure 2 shows the result. Both compositional models show consistent performance regardless of the path length. However, the performance of the non-compositional models drops sharply as the length increases. The results show the compositional models preserve a graph structure in the embedded space. It is worth emphasising that although the compositional length for training is 2, the compositional models show consistent results on predicting path of length 3 and 4.

Table 4 shows an example of the path prediction result starting from entity `Mental-or-Behavioral (MB) Dysfunction` followed by two relations `Affects` and `Produces`. Both compositional and non-compositional models predict triples well. For length-2 path prediction, only the compositional models can capture correct entities on top 5. We also visualise the multi-dimensional entities inferred by PNORMAL and PCOMP-MUL into a two-dimensional space through the spectral clustering [26] in Figure 3. A circle represents an entity, and the size of the circle is proportional to the uncertainty of the entity in the latent space. In the UMLS dataset, the entities are categorised into 15 types, e.g. `Disorders`, `Living-Beings`, `Phenomena`, etc. We use the same color to represent the entities with the same type. The entities with the same type are located closer to each other with PCOMP-MUL than PNORMAL.

7. EXPERIMENTS ON INCREMENTAL KNOWLEDGE POPULATION

In this section, we show results for Thomson sampling of the PRESCAL and its compositional variants, first, on two synthetic datasets, and then on three common benchmarks used in the knowledge graph completion task.

7.1 Thompson Sampling on synthetic data

We first synthesise two datasets following the model assumptions in Equation 1 to 3. First, entities and relations

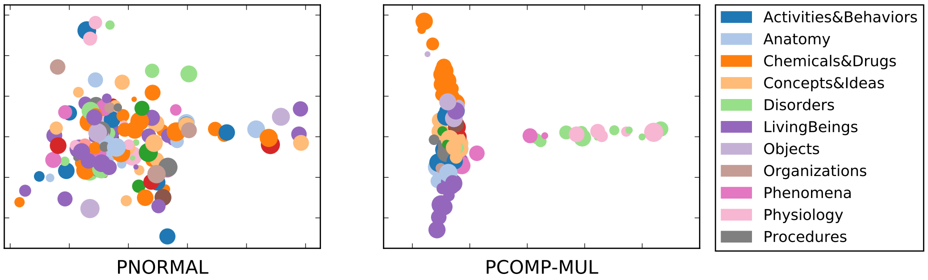
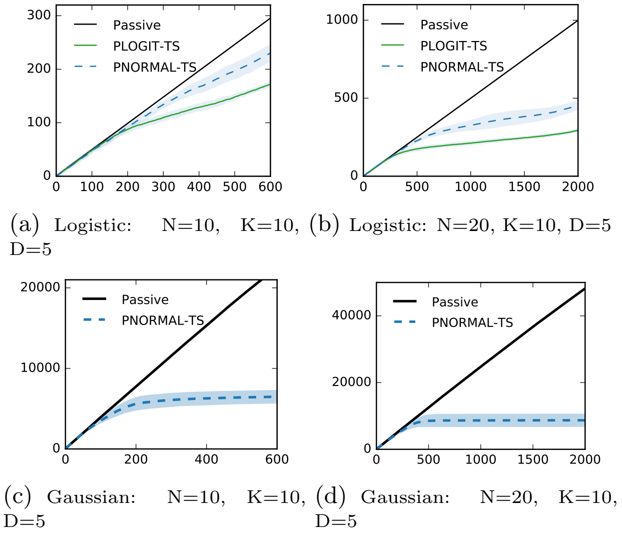


Figure 3: Embedding learned entities of the UMLS dataset into a two-dimensional space through the spectral clustering. Entities with the same type are represented by the same color.



(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 4: Cumulative regret of particle Thompson sampling with Gaussian and logistic output (PNORMAL-TS, PLOGIT-TS) 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.

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 formulation, 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 4 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 logistic-output PRESCAL (PLOGIT) and Gaussian-output PRESCAL (PNORMAL) whereas the dataset with the Gaussian output can only be trained by PNORMAL. Figure 4(a) and 4(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 4(c) and 4(d) show that the Thompson sampling for PNORMAL (PNORMAL-TS) outperform passive learning in the real valued dataset.

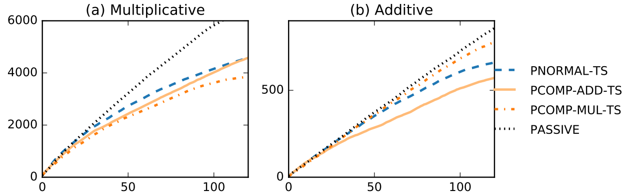
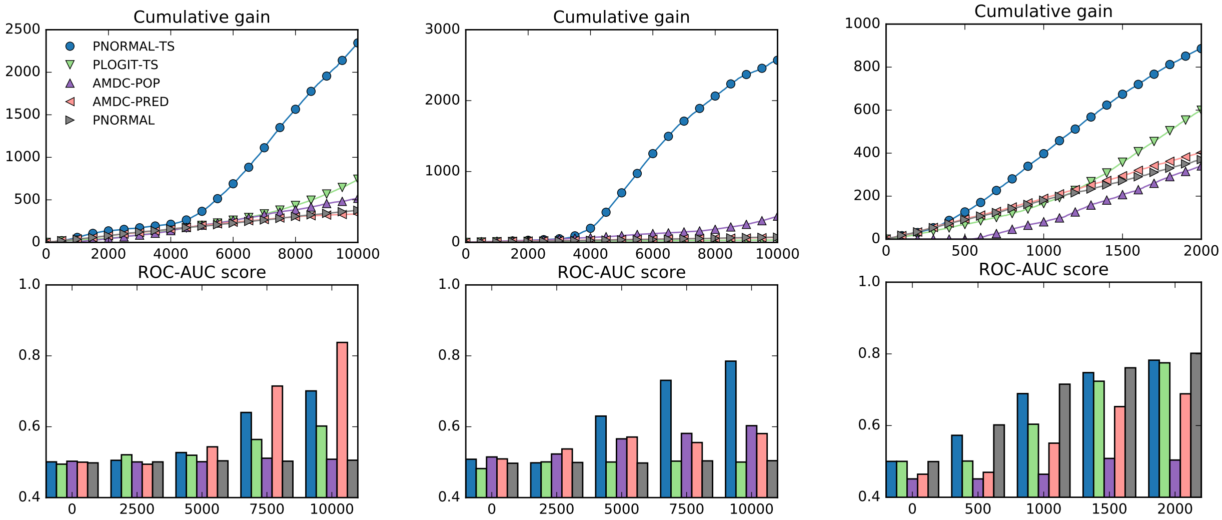


Figure 5: Cumulative regret of particle Thompson sampling of the compositional models on synthetic dataset with N=5, D=5. The synthetic dataset has three relations (K=3); the first two are independently generated, and the third relation is composed by the first two relations. The dataset used in (a)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

7.2 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 a set of triples with Gaussian output as in Equation 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 4, 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 a real world situation, the



(a) KINSHIP

(b) UMLS

(c) NATION

Figure 6: The cumulative gain and ROC-AUC score of the Thompson sampling with passive learning and AMDC models. Thompson sampling with PRESCAL (PNORMAL-TS) model achieves the highest cumulative gain to compare with AMDC and passive learning algorithms and shows comparable performance on ROC-AUC scores.

expanded tensor can only be 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 PNORMAL 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 5 shows the cumulative regrets on synthetic datasets. The multiplicative and additive compositionality are used to generate the dataset for Figure 5(a) and 5(b), respectively. The results correspond to our assumption: the Thompson sampling for multiplicative compositional model (BCOMP-MUL-TS) shows lower regrets on the multiplicative data in Figure 5(a), and the Thompson sampling for additive compositional model (BCOMP-ADD-TS) shows lower regrets on the additive compositional data in Figure 5(b), and both have lower regrets than passive learning or PNORMAL-TS without compositions.

7.3 Thompson sampling on real datasets

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

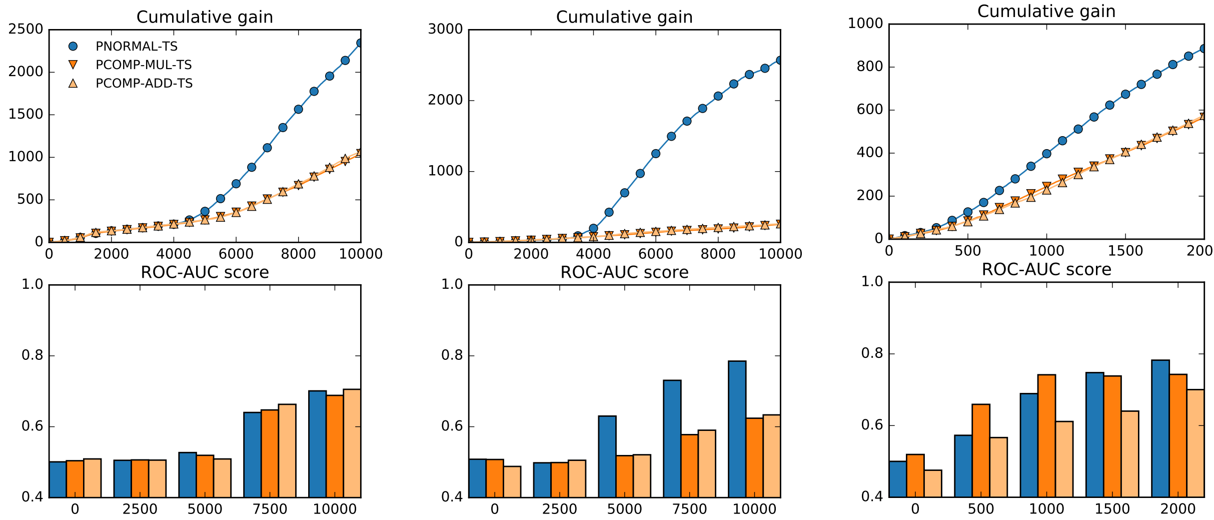
Experimental settings: We compare the Thompson sampling models with AMDC models, and PRESCAL for passive learning. AMDC model has been proposed to achieve two different active learning goals, constructing a predictive model and maximising the valid triples in a knowledge base, with two different querying strategies [12]. 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 PRESCAL, 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 triple 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, a model obtains a label of the queried triple from an oracle, then the model updates the parameters.

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

Exploitation and exploration: Figure 6 and 7 show the cumulative gains and ROC-AUC scores of the Thompson sampling on three real datasets. PNORMAL-TS 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 PNORMAL-TS across all datasets.

In the original AMDC work [12], AMDC-POP model obtains more valid triples than AMDC-PRED, and AMDC-PRED shows high ROC-AUC scores than AMDC-POP. In



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

Figure 7: The cumulative gain and ROC-AUC score of the Thompson sampling with the non-compositional model and compositional models. Unlike the knowledge completion task in Section 6, both compositional models perform worse than PNORMAL-TS in the incremental knowledge population.

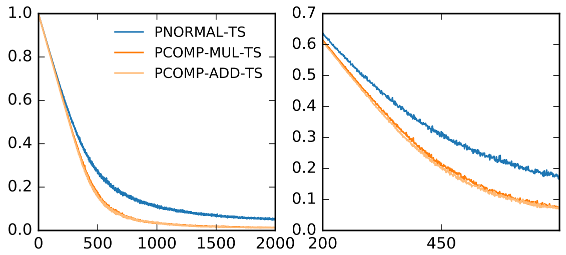


Figure 8: Trace plot of mean posterior variance of the non-compositional model and compositional models. Y-axis denotes the average posterior covariance, and X-axis denotes the number of queries. The second plot magnifies the first plot.

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 and query size results in the different performances: in the original experiment, the model starts from a small set of training data, and the query size was 1,000 for KINSHIP and UMLS. So this gives the model focusing on exploit and advantage, whereas in our experiment, we start from zero observation and query one triple at each time, 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 non-compositional models, especially than PNORMAL-TS, as shown in Figure 7. This is counter-intuitive to our general understanding where the model that performs well in the predictive task also shows a better performance in the active learning. Of course, we also emphasis the difference between two experiments; the goal of incremental population is to maximise the number of triples whereas the goal of knowledge completion in Section 6 is to maximise the predictive performance. Nevertheless, the compositional models do not outperform PNORMAL-TS in the active learning. This result can be partially understood in terms of the balance between exploration-exploitation. Figure 8 shows the average posterior variance of the entity vectors. We compute the eigenvalues of posterior covariance matrix Λ_i^{-1} and trace the average eigenvalues over the iterations. As shown in the figure, the average variance of the compositional model shrinks much faster than the PNORMAL-TS. Because the exploration-exploitation of the Thompson sampling depends on the posterior uncertainty, the fast shrinkage in the posterior variance may indicate the under exploration of the model. This is predictable to a certain extent in the sense that one new triple with the compositional models induces multiple new observations in the compositional triples, so the uncertainties of entities and relations are measured less than those with non-compositional model. The most of active learning algorithm utilise an uncertainty of a model, and therefore, a model with augmented structures such as the relation compositions should be more careful about reflecting its uncertainty correctly.

8. 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 both knowledge completion and active knowledge population problems. 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, probabilistic 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 [12] views knowledge population and knowledge completion 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 maximising 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 based construction problem [7].

References

[1] C. M. Bishop. Pattern recognition. *Machine Learning*, 2006.

[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250. ACM, 2008.

[3] O. Chapelle and L. Li. An empirical evaluation of thompson sampling. In *NIPS*, 2011.

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

[5] P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[6] L. Dong, F. Wei, M. Zhou, and K. Xu. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. In *ACL*, pages 260–269, 2015.

[7] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610. ACM, 2014.

[8] A. Doucet and A. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. *Handbook of Nonlinear Filtering*, (December):656–704, 2011.

[9] K. Guu, J. Miller, and P. Liang. Traversing knowledge graphs in vector space. In *EMNLP*, 2015.

[10] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[11] J.-Y. Jiang, J. Liu, C.-Y. Lin, and P.-J. Cheng. Improving ranking consistency for web search by leveraging a knowledge base and search logs. In *CIKM*, pages 1441–1450. ACM, 2015.

[12] H. Kajino, A. Kishimoto, A. Botea, E. Daly, and S. Kotoulas. Active learning for multi-relational data construction. In *WWW*, pages 560–569, 2015.

[13] J. Kawale, H. H. Bui, B. Kveton, L. Tran-Thanh, and S. Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *NIPS*, pages 1297–1305, 2015.

[14] D. Kim, H. Wang, and A. Oh. Context-dependent conceptualization. *IJCAI*, 2013.

[15] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

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

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

[18] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[19] A. Neelakantan, B. Roth, and A. McCallum. Compositional Vector Space Models for Knowledge Base Completion. In *ACL*, 2015.

[20] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. 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.

[21] M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816, 2011.

[22] N. Ruchansky, M. Crovella, and E. Terzi. Matrix completion with queries. In *KDD*, pages 1025–1034. ACM, 2015.

[23] M. N. Schmidt and S. Mohamed. Probabilistic non-negative tensor factorization using markov chain monte carlo. In *Signal Processing*, pages 1918–1922. IEEE, 2009.

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

[25] D. J. Sutherland, B. Póczos, and J. Schneider. Active learning and search on low-rank matrices. In *KDD*, pages 212–220. ACM, 2013.

[26] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[27] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *ICDM*, volume 10, pages 211–222. SIAM, 2010.