1. Strict Bayesian approach doesn't seem to work: no data, no posterior. But, posterior over latent variables. Latent variables are embeddings and tranformations, *observed are the memberships/responsibilities* . So we *do* have observed variables?
1.1. Optimize for the posterior over the connections... This would tie in better with the link prediction use case.
1.2. If the structure of the grpah is the latent variable, we should be bale to sample a structure from the model.
2. Take inspiration from the fractal model start with X_0 $N(0, I)$ as embedding for all nodes and iterate X_1 is X_0 transformed by all translations, for all points.
3. It doesn't look like the constraints of just a probability model are sufficient to determine the problem. It seems most elegant to assume that the embeddings are (isometric) Gaussians, and to optimize for minimal KL divergence to the transformed ones.

A Bayesian Approach to Knowledge Graph Node Embeddings

August 16, 2017

Abstract

We present a variational approach to to the problem of embedding knowledge graph nodes.

1 Introduction

Knowledge graph node embeddings are vector representations of the nodes in a knowledge graph. For a knowledge graph with nodeset N, an embedding is a function $v : N \to \mathbb{R}^k$ such that v organizes the nodes meaningfully in the space $\mathbb{R}^k$ according to some criterion.

Effective embeddings are an important first step in many pipelines for knowledge graph visualisation, completion, classification and other tasks. Many embedding algorithms model the relations in the knowledge graph as maps in the embedding space: either translations, or more complex transformations. If for instance, we have the triple $(\texttt{Mary}, \texttt{childof}, \texttt{John})$, then we learn embeddings $v_{\texttt{Mary}}, v_{\texttt{John}} \in \mathbb{R}^k$ (for some chosen embedding dimension k) and a map $f_{\texttt{childof}} : \mathbb{R}^k \to \mathbb{R}^k$ for the relation `childof` such that $f_{\texttt{childof}}(v_{\texttt{John}})$ should be as close to $v_{\texttt{Mary}}$ as possible (for some chosen notion of distance).

This scheme works well, but has trouble representing multiple relations. If John has more than one child, what should $f_{\texttt{childof}}(v_{\texttt{John}})$ be? Many approaches have been introduced, often requiring an advanced mathematical structure for the embedding space, such as complex vectors [**?**], or holographic projections [**?**].

Our approach is based on a simple intuition. For each source node, we place a spherical Gaussian distribution over the embedding. Transforming this Gaussian by the relational map should then result in a new Gaussian

which maximizes the likelihood of the nodes connected to the original by the given relation.[1] In other words, transforming a spherical Gaussian centered on $v_{\texttt{John}}$ by $f_{\texttt{childof}}$ should result in a distribution that maximizes the probability of John's children.

The task, then, given a knowledge graph, is to find an embedding v_i for each node i, and a function f_r for each relations, such that the resultant Gaussians optimally cover the required nodes. Intuitively, this problem can be solved by an alternating optimization algorithm in the vein of expectation-maximization: given a choice for the node embeddings, we can find the optimal functions to represent the relations, and given a choice for the relation functions, we can compute the optimal embeddings. Alternating these two optimizations is likely to converge to a good solution. To formalize this intuition in an elegant manner, we take a Bayesian approach, and find a solution using variational inference. The precise model is detailed in Section 2.

Our approach has several advantages over the alternatives:

- The nodes are embedded in a simple Euclidean space, making no special demands on downstream architectures.
- The multiple-relation issue is solved in an intuitive manner: mapping a parent node to its children creates a probability distribution that covers the children.
- The method is agnostic to the family of maps used to represent relations. We choose affine maps as a good tradeoff between ease of optimization, rigidity to protect against overfitting, and expressiveness. The method easily generalizes to other function classes.
- For linear maps, we can decompose the loss function into factors which can be solved analytically. Thus, the problem can be solved using an iterative algorithm which is guaranteed to converge to a local optimum.
- Negative sampling is not required. The basic framework constrains the problem sufficiently to lead to models that generalize well.
- The variational framework protects against convergence to solutions with singularities, and allows us to incorporate hyperparameters like the dimension of the embedding space into the optimization process.

[1] This approach is inspired by the Coherent Point Drift algorithm [], which uses a similar approach for the problem of point set registration.

We test our embeddings on the tasks of knowledge base completion and node classification. We outperform the state-of the art in each case. We also compare a general-purpose stochastic gradient descent approach for our model, to the variational search, showing that the latter converges faster, in fewer iterations, and to better solutions.

2 Model

Define a knowledge graph G as a triple $G = (V_G, R_G, E_G)$ with V_G representing a set of labeled nodes, R_G representing a set of available relations, and $E_G \subset V_G \times R_G \times V_G$ the set of *triples*—edges labeled with relations. Since we are not interested in modeling the internal structure of labels, we will assume that V is always the set of the first $|V|$ natural numbers, and likewise for R.

To simplify notation, we will assume that the graph being modeled is clear from context, and drop the subscript G.

Let T_{spo} be a random bolean variable indicating whether the triple $\langle s, p, o\rangle$ is true. Let T^u be the set of all such variables (for all combination of s, p, o) and let T be the set of such variables for the observed triples. We will assume that the observed triples we samples from the set of all true triples by some unspecified distribution $p(G|G^u)$. This graph is fully specified by a set of node-embeddings x_i and relation-maps f_p: we have $f_p(x_s) = x_o$ is and only if the triple s, p, o is true. Thus, the embeddings specify the graph. We assume that the observed triples T constitute a sequence of i.i.d. draws from the set of all triples satisfying the embeddings.

To develop our model, we will first assume we know a value $x_i \in \mathbb{R}^k$ for each node i in the knowledge graph, and a function f_p for each predicate p. We will choose affine functions as our family, so that we can say (with some abuse of notation) $f_p \in \mathbb{R}^{k\times k} \times \mathbb{R}^k$. For such a fully known model, we will say that a triple is true if and only if f_p is mapped exactly onto x_s:

$$p(T_{spo} = \text{true} \mid f_p, x_s, x_o) = \begin{cases} 1 & \text{if} f_p(x_s) = x_o \\ 0 & \text{otherwise} \end{cases}.$$

This model suffers from the same problem as TransE: we cannot represent multiple relations. Interestingly, the fact that we do not observer x_s, x_o and f_p directly allows us to solve this problem. We express our uncertainty about these values in probability. If s is mapped to several points by p, we are thus *uncertain* about which point f_p should map x_s to. Expressing this

uncertainty as probability as well allows as to retrieve a probability distribution over the model parameters that let x_s map to deveral different x_o's under f_p.

We start dy removing the assumption that we know the model parameters. We express our uncertainty as probabilities: $p(F_p = f_p)$, $p(X_s = x_s)$, $p(X_o = x_o)$ and marginalize them out. The probability of a triple becomes:

$$p(T_{spo} = \text{true}) = \int_{x_s, f_p, x_o \in S_{spo}} p(X_s = x_s) p(F_p = f_p) p(X_s = x_s)\, dx_s df_p dx_o$$
$$\text{with } S_{spo} = \{(x_s, f_p, x_o) \mid f_p(x_s) = x_o\}$$

$$p(\{X_i\}, \{F_p\} \mid T) = \frac{p(T \mid \{X_i\}, \{F_p\}) \prod_i p(X_i) \prod_p p(F_p)}{P(T)}$$

3 Experiments

4 Conclusion

It is worth noting that we started from an impossible definition of the structure of our graph. The presumed unobserved true embeddings cannot exist for any graph contain multiple children or parents. Moreover, the system may be overdetermined: even with unique relations for every triple, there may still be no way to satisfy the whole knowledge graph in a single set of embeddings. Nevertheless, the Bayesian approach allows us to satisfy the resulting contradictions, by not choosing a single truth, but but providing a probability distribution over configurations that are each close to a correct one. In other words, the uncertainty about how the contradictions should be resolved is simply modeled as another probability, and folded into the model, combined with our other uncertainties.