Building Dynamic Knowledge Graphs

Jay Pujara
Department of Computer Science
University of Maryland
College Park, MD 20742
jay@cs.umd.edu

Lise Getoor
Department of Computer Science
University of California
Santa Cruz, CA 95064
getoor@soe.ucsc.edu

Abstract

One of the fundamental challenges in constructing knowledge bases is that the knowledge we seek to capture comes from an uncertain and changing world. Constructing a KB necessitates dealing with a continual flow of evidence that requires updating and revising the KB. A key question for the AKBC community is how to construct and update KBs using a steadily growing stream of noisy extractions from large text corpora, like the Web. In this vision paper, we (1) formalize the problem of building a dynamic knowledge graph, (2) define a probabilistic model for using new evidence to update the knowledge graph, and (3) present an efficient algorithm for knowledge graph construction on streaming extractions.

1 The Challenge of the Dynamic Knowledge Graph

Consider a knowledge base (KB) constructed in 2008: it might accurately capture that Barack Obama is a senator living in Illinois. However, a year later the correct KB must be revised: Barack Obama is no longer a senator, but a president and his residence is Washington, DC. As this tiny example motivates, a crucial challenge in updating the KB is that the facts it contains are not independent; updating one fact, such as employment, may require revising many others, such as residence.

Humans have grappled with updating beliefs in response to changing conditions for millennia and the problem has been studied in disciplines ranging from philosophy to computer science. Research over the last few decades has studied many aspects relevant to KBs, spanning areas such as belief revision[1], non-monotonic reasoning[2], and temporal knowledge bases[3, 4, 5]. However, this existing research does not address the practical problems facing AKBC, where information extraction systems are applied to an infinite stream of textual data (such as the Web) and produce millions of extractions with varying degrees of confidence[6, 7]. In this paradigm, we must identify the important statistical dependencies between facts and determine which facts are likely to require revision.

To capture statistical dependencies and uncertain information, we adopt a probabilistic approach and formulate the problem in terms of knowledge graph construction. Knowledge graphs represent entities as nodes and the relationships between them as edges, but the structure of the graph is often unknown - requiring entity resolution, node labeling and link prediction. This problem is known as knowledge graph identification (KGI)[8]. In KGI, a probabilistic model specifies a probability distribution over possible knowledge graphs, and inference finds the most probable knowledge graph.

Here we propose three of the fundamental challenges to applying KGI in a dynamic setting. (1) How do probability distributions change as new evidence is introduced? What characterizes a desirable model for updating the knowledge graph? (2) How can we specify probabilistic models for dynamic data? How can modeling languages easily express relationships between new extractions and existing knowledge? (3) What considerations are important for inferring dynamic knowledge graphs? How can we improve the speed of inference without sacrificing the quality of the inferred knowledge graph?

In this paper, we characterize the objective function for dynamic knowledge graph construction (Section 2), introduce expressive syntax for specifying probabilistic models for this task (Section 3), and propose an inference algorithm for dynamic graph construction that efficiently uses uncertainty and evidence (Section 4).

2 A Formal Objective for Dynamic Knowledge Graph Construction

Belief revision defines two basic operations to integrate new evidence into a set of beliefs. The first is updating an existing belief in response to new evidence. In probabilistic models, this can be thought of as updating the conditional probability of a random variable. The second operation is revision, the process of adding and reconciling beliefs. This can be interpreted as introducing new variables into the joint probability distribution. However, the complication we face in our problem setting is that as evidence is added, defining the probability distribution becomes intractable. We must use limited evidence to define a distribution over knowledge graphs.

The key question is how to use this limited evidence to define the distribution. This question can be answered differently for revisions and updates. For revisions, a crucial difficulty is finding the set of related variables necessary to update the probability distribution and assess the validity of the revision. We address this challenge through model specification (in Section 3). In the case of an update, a dynamic KGI system must decide if an update is necessary, which, in turn requires understanding how well the existing probability distribution captures the value of the variable. This problem is discussed in Section 4. However, to guide the discussion of these two essential components, in this section we introduce a formal objective for the dynamic knowledge graph construction task.

We begin by introducing the probability distribution that KGI defines over possible knowledge graphs (G). The distribution over knowledge graphs is induced by the particular model Π and conditions on the evidence (E) in the form of extractions from text. We use the notation $P^{\Pi}(G|E)$ to refer to this distribution. As new evidence, E_1 is added, applying the model results in updates and revisions. This produces a new distribution (P_1) over a new family of possible knowledge graphs (G_1): $P_1^{\Pi}(G_1|E \cup E_1)$. Unfortunately, as the evidence grows the size of the graphical model also increases, in terms of both nodes (random variables) and edges (conditional dependencies between variables). The result is that the distribution over knowledge graphs becomes intractable to define.

The goal of dynamic knowledge graph construction is to infer the best knowledge graph using a fixed amount of evidence (determined by the computational resources available). We formalize this problem as one of choosing a model (Δ) that includes the appropriate context for revisions (discussed in Section 3) and an algorithm (S^{Δ}) that identifies pertinent evidence using the uncertainty in existing inferences (discussed in Section 4). The model Δ specifies the interesting relationships as the knowledge graph grows, while S^{Δ} returns a set of interesting evidence to use during inference. These two components allow us to define a new probability distribution over knowledge graphs: $P_1^{\Delta}(G_1|S^{\Delta}(G, E \cup E_1))$

To determine the quality of the new distribution, P_1^{Δ}, we compare the dynamic knowledge graph to the knowledge graph inferred by KGI using all evidence. We refer to these knowledge graphs, respectively as G_1^{Δ} and G_1^{Π}. We define the quality of the dynamic model, $Q(\Delta)$, as the reciprocal of the L2-norm of the difference between inferred values of variables, using notation: $||G_1^{\Delta} - G_1^{\Pi}||_2^{-1}$. As the distance between the inferences in the dynamic and full knowledge graph models decreases, the quality score grows. While this quality score is not useful in the true problem setting, where the full knowledge graph is intractable to compute, it provides a mechanism for evaluating the effectiveness of the dynamic model and algorithm during training.

Having defined this quality score, what remains is to define an objective to optimize the dynamic model and algorithm. Using the quality score, we can define the formal objective of dynamic knowledge graph identification:

$$\max_{\Delta; S^{\Delta}} \quad Q(G_1^{\Delta}) \quad \text{where} \quad G_1^{\Delta} = \text{argmax}_{G_1} P_1^{\Delta}(G_1|S^{\Delta}(G, E \cup E_1)) \ \text{ s.t. } |G_1 \cup S^{\Delta}(G, E \cup E_1)| < t$$

Although optimizing over all possible dynamic models and algorithms is impossible, we show in the next section how to parameterize Δ and S^{Δ} to allow maximization over models and algorithms in the parameterized space.

3 Specifying Models for Dynamic Knowledge Graphs

Thus far, we have defined the problem in terms of probability distributions over knowledge graphs. However, these probability distributions are defined by variable dependencies in a probabilistic graphical model. Specifying graphical models that can selectively encode dependencies between variables is a necessary step for dynamic graph construction. Previous work[4, 5] has considered a variety of temporal aspects of knowledge necessary for cleaning KBs. In this section we describe desirable properties for dynamic model specification (Δ) and introduce syntax that easily allows models to specify when to use new evidence and when to condition on existing inferences.

For example, we may want to encode a dependency between a person's employment, the location of their employer, and their residence. While we expect a person's employment and residence to change frequently, the location of a company may remain relatively stable. Thus a dynamic model may want to frequently infer new values for employment and residence while rarely inferring new values for company location.

We consider extending the probabilistic soft logic (PSL) framework[9] to support specifying dynamic models. Like many statistical relational learning approaches, PSL uses first-order logic syntax to template graphical models. The model, specified by a set of universally quantified rules, is ground using evidence and the resulting ground rules (R) are mapped to factors in a graphical model. For example the rule $\textsc{Rel}(\texttt{worksFor}, P, C) \wedge \textsc{Rel}(\texttt{headquarters}, C, L) \Rightarrow \textsc{Rel}(\texttt{residentOf}, P, L)$ captures a factor relating three types of facts: employment (`worksFor`), company location (`headquarters`) and residence (`residentOf`).

Our goal is to improve the tractability of inference by selectively grounding rules using a subset of the evidence. The main obstacle is that the model specification, in terms of rules, does not allow us to control the grounding of these rules. We extend the logical syntax used by PSL to allow each logical predicate in a rule to specify the set of atoms used to ground that rule. As a concrete example, consider the modified rule where the `worksFor` and `residentOf` predicates ground against the set of atoms E_1 while `headquarters` is grounded against all evidence:

$$\textsc{Rel}(\texttt{worksFor}, P, C, \mathbf{E}_1) \wedge \textsc{Rel}(\texttt{headquarters}, C, L) \Rightarrow \textsc{Rel}(\texttt{residentOf}, P, L, \mathbf{E}_1)$$

This extension of the model syntax allows a range of grounding specifications, including grounding each predicate against a different set of evidence, or lazily instantiating atoms outside the specified evidence sets when using an open-world model. Experimental code for this extension is available at `https://github.com/linqs/psl/tree/online_psl`

4 An Algorithm for Dynamic Model Optimization

The final component in our approach to dynamic knowledge graph construction is an algorithm for selecting limited evidence used to defined the distribution over knowledge graphs. We approach this problem by defining a ranking function over evidence, allowing the algorithm to easily fulfill requests for varying amounts of evidence. We assess the importance of each variable in the optimization using two criteria: (1) how important is this evidence to our model? and (2) how uncertain is our estimate of this variable? We measure these criteria using features from the optimization algorithm used to find the most probable knowledge graph. In this section we provide a brief background of this optimization algorithm, then discuss how features from this algorithm can be used to determine the importance of evidence.

MPE inference of models in PSL is implemented using alternating dual method of multipliers (ADMM) for consensus optimization[10]. Bach et al.[11] demonstrated that consensus optimization can vastly improved scalability for inference in constrained, continuous MRFs, with empirical results suggesting inference time scaled with the number of potential functions. We analyze the ADMM objective and show how an optimization-aware algorithm can be used to determine which variables are important to retain in the optimization. PSL defines the probability of a given configuration of variables (I) using an energy function where each ground rule is captured by a potential ϕ_r:

$$P_{\Pi}(I) = \frac{1}{Z}\exp\left[-\sum_{r \in R} w_r \phi_r(I)\right]$$

At a high level, consensus optimization decomposes a problem into subproblems and iteratively optimizes the model until convergence. In each iteration, the algorithm introduces a separate optimization subproblem for each potential function in the graphical model, which is optimized independently over local copies of the variables (I_r). At the end of each iteration, the local values of each variable are averaged across potentials to provide a consensus estimate (I_r^c). Deviations from the consensus estimate are penalized in each potential through the introduction of a Lagrange multiplier.

More formally, ADMM decomposes the optimization of the term $\sum_{r \in R} w_r \phi_r(I)$ into separate optimizations for each ϕ_r, using the augmented Lagrangian to enforce consistency in values for the same variables. This optimization is written in the scaled form as: $\min w_r \phi_r(I) + \frac{\rho}{2} ||I_r - I_r^c + \frac{1}{\rho} y_r||_2^2$ where I_r are values assigned to the variables in potential ϕ_r and I_r^c are consensus estimates for those variables found by averaging the previous round values of each variable. Here y_r is the set of Lagrange multipliers for the variables, derived from the constraint $I = I^c$.

Given the optimization objective, we consider two factors that affect how important the variable $x_i \in I$ is to the optimization. One factor is the weights w_r for the potentials that take x_i as an argument, since intuitively potentials with higher weights have the largest impact on the objective value. The second factor that captures the influence of x_i are the Lagrange multipliers for variable $y_r[i]$, which, intuitively, measure the disagreement of the potential function and the consensus estimates. Thus, we can define the importance of variable x_i as a combination of these two factors. We introduce a simple, scaled combination of these two features to generate ranking scores: $\max_R(w_r[x_i]) + \beta y_r[i]$, with scaling parameter β.

In addition to variables in the initial inference optimization, we also consider how to score variables from new evidence added to the knowledge graph. For variables in the newly introduced evidence determining the weights of associated rules is straightforward. Since the Lagrange multipliers were calculated without using the evidence, their value may incorrectly capture the disagreement of the value of the variable. In the case of updates to existing variables, we can compare the new and existing values assigned to the variable. For new variables that were not previously part of the graphical model, we use the maximum value of Lagrange multipliers in the optimization.

5 Discussion

In this paper, we motivate a fundamental problem confronting the AKBC community: constructing knowledge graphs from an ever-growing stream of evidence. Broadly we argue that given boundless evidence and bounded computational resources, collective methods for knowledge graph construction must be capable of prioritizing the inferences and evidence available. In this process, we confront two critical questions: What must we learn? What existing knowledge will best help us learn it?

We identify three basic challenges:

1. Formalizing an objective function for dynamic knowledge graph construction
2. Specifying models that selectively leverage evidence
3. Selecting the most important evidence for updating beliefs

For each of these challenges, we provide analysis and an initial solution. We believe that dynamic knowledge graph construction is a new frontier where many interesting open questions remain. Our initial exploration suggests that our approach holds promise for addressing some of these questions, and we are eager to apply these ideas in our future work.

Acknowledgments This work was partially supported by NSF grant IIS1218488. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Peter Gardenfors, editor. *Belief Revision*. Cambridge University Press, New York, NY, USA, 1992.

[2] Laura Giordano and Francesca Toni. Knowledge representation and non-monotonic reasoning. In Agostino Dovier and Enrico Pontelli, editors, *A 25-year Perspective on Logic Programming*, pages 87–111. Springer-Verlag, Berlin, Heidelberg, 2010.

[3] James F. Allen. An interval-based representation of temporal knowledge. In *IJCAI*. Morgan Kaufmann Publishers Inc., 1981.

[4] Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. Coupled temporal scoping of relational facts. In *WSDM*, New York, NY, USA, 2012. ACM.

[5] Yafang Wang, Maximilian Dylla, Marc Spaniol, and Gerhard Weikum. Coupling label propagation and constraints for temporal fact extraction. In *ACL*, Stroudsburg, PA, USA, 2012. ACL.

[6] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, 2010.

[7] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open Information Extraction from the Web. *Communications of the ACM*, 51(12), 2008.

[8] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. Knowledge Graph Identification. In *ISWC*, 2013.

[9] Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. A Short Introduction to Probabilistic Soft Logic. In *NIPS Workshop on Probabilistic Programming*, 2012.

[10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011.

[11] Stephen H. Bach, Matthias Broecheler, Lise Getoor, and Dianne P. O'Leary. Scaling MPE Inference for Constrained Continuous Markov Random Fields with Consensus Optimization. In *NIPS*, 2012.