Extracting Adverse Drug Events from Text using Human Advice

Phillip Odom[1], Vishal Bangera[1], Tushar Khot[2], David Page[3], and Sriraam Natarajan[1]

[1] Indiana University Bloomington
[2] Allen Institute of AI
[3] University of Madison-Wisconsin

Abstract. Adverse drug events (ADEs) are a major concern and point of emphasis for the medical profession, government, and society in general. When methods extract ADEs from observational data, there is a necessity to evaluate these methods. More precisely, it is important to know what is already known in the literature. Consequently, we employ a novel relation extraction technique based on a recently developed probabilistic logic learning algorithm that exploits human advice. We demonstrate on a standard adverse drug events data base that the proposed approach can successfully extract existing adverse drug events from limited amount of training data and compares favorably with state-of-the-art probabilistic logic learning methods.

1 Introduction

Adverse drug events (ADE) are one of the major causes of death in the world. For instance, nearly 11% of hospital admissions of older adults in US are attributed to ADEs [5]. Consequently there has been an increase in focus and application of statistical learning algorithms for detecting ADEs from data such as Electronic Health Records (EHRs) and clinical studies [16]. While there is a plethora of research on detecting them from clinical data, there are not many methods that can validate the output of these algorithms except for manually scanning through the ADEs. In this case, the burden is on the expert to evaluate these extracted ADEs by knowing all the ones published in the literature.

We explore the use of published medical abstracts to serve as ground truth for evaluation and present a method for effectively extracting the ADEs from published abstracts. To this effect, we adapt and apply a recently successful machine learning algorithm [15] that uses a human expert (say a physician) as more than a "mere labeler", i.e., the human expert in our system is not restricted to merely specify which of the drug event pairs are true ADEs. Instead, the human expert would "teach" the system much like a human student by specifying patterns that he/she would look for in the papers. These patterns are employed as advice by the learning system that seamlessly integrates this advice with training examples to learn a robust classifier.

More precisely, given a set of ADE pairs (drug-event pairs), we build upon an NLP pipeline [12] to rank the ADE pairs based on the proof found in the literature. Our system first searches for PubMed abstracts that are relevant to the current set of ADE pairs. For each ADE pair, these abstracts are then parsed through a standard NLP parser (we use Stanford NLP parser [3], [10]) and the linguistic features such as parse trees, dependency graphs, word lemmas and n-grams etc. are generated. These features are then used as input to a relational classifier for learning to detect ADEs from text. The specific relational classifier that we use for this purpose is called Relational Functional Gradient-Boosting (RFGB) classifier [14]. The advantage of employing this classifier over standard machine learning classifiers such as decision-trees [11], SVMs [2] and boosting [17] is that RFGB does not assume a flat-feature vector representation for learning. This is important in our current setting as it is unreasonable to expect the same number of abstracts for each ADE pair. More importantly, it is not correct to expect the same type of parse trees and dependency graphs for each article (as each set of authors can have a different style). The presence of this diverse set (and number) of features necessitates the use of a classifier that can leverage a richer representation that is more natural to model the underlying data. Needless to say, relational representations have been successful in modeling the true nature of the data and we adapt the state-of-the-art relational learning algorithm.

While powerful, standard learning will not suffice for the challenging task of extracting ADEs as we will show empirically. The key reason is that we do not have sufficient number of training examples to learn a robust classifier. Also, the number of linguistic features can be exponential in the number of examples and hence learning a classifier in this hugely imbalanced space can possibly yield sub-optimal results. To alleviate this imbalance and guide the learner to a robust prediction model, we explore the use of human guidance as advice to the algorithm. This advice could be in terms of specific patterns in text. For instance it is natural to say something like, "if the phrase *no evidence* is present between the drug and event in the sentence then it is more likely that the given ADE is not a true ADE". The learning algorithm can then identify the appropriate set of features (from the dependency graph) and make the ADE pair more likely to be a negative example. As we have shown in non-textual domains [15], this type of advice is robust both to noisy training examples as well as for a small number of training examples. We adapt and extend the previous work for textual data.

To summarize, we make several key contributions: first is that we develop a robust method that can automatically learn a classifier for detecting ADEs from text. This goes beyond current state-of-the-art methods that employ a hand-crafted classifier such as conditional random fields (CRF) [6]. Second, we lessen the burden on human experts by allowing them to provide some generalized advice instead of the mundane task of manually labeling a huge number of learning examples. Also, it removes the burden of designing a specific classifier such as CRF or a SVM for the task. Effectively, our expert is required to be a domain expert (who understands medical texts) instead of machine learning expert who needs to carefully design the underlying model and set the parameters. Finally,

we evaluate the learning method on a corpus of available ADEs and empirically demonstrate the superiority of the proposed approach over the alternatives.

The rest of the paper is organized as follows: we present the background on the learning algorithms (with advice) next. We follow this with a discussion on how these algorithms are adapted to our specific task. We then present the empirical evaluations before concluding by outlining areas for future research.

2 Prior Work on Learning Relational Models

We now present our prior work on relational classifiers that we build upon in this work. We first present Relational Functional Gradient Boosting (RFGB) [14] and its extension to handle expert knowledge [15].

2.1 RFGB

Before outlining the algorithms that we employ, we will present them in the standard machine learning setting. Gradient ascent is the standard technique for learning the parameters of a model and typically starts with initial parameters θ_0 and computes the gradient (Δ_1) of an objective function w.r.t. θ_0. The gradient term is then added to the parameters θ_0 and the gradient ascent is performed for the new parameter value $\theta_1 = \theta_0 + \Delta_1$ and repeated till convergence. Friedman [4] proposed an alternate approach where the objective function is represented using a regression function ψ over the examples $\mathbf{x}$ and the gradients are performed with respect to $\psi(x)$. Similar to parametric gradient descent, the final function after n iterations of functional gradient-descent is the sum of the gradients, i.e., $\psi_n(x) = \psi_0(x) + \Delta_1(x) + \cdots + \Delta_n(x)$. Each gradient term ($\Delta_m$) is a regression function over the training examples (E) and the gradients at the m^{th} iteration can be represented as $\langle x_i, \Delta_m(x_i) \rangle$ where $x_i \in E$.

Rather than directly using $\langle x_i, \Delta_m(x_i) \rangle$ as the gradient function (memorization), functional gradient descent *generalizes* by fitting a regression function $\hat{\psi}_m$ (generally regression trees) to the gradients Δ_m. The $\hat{\psi}_m$ function uses the features of the example x to fit a regression function to $\Delta_m(x)$. For example, to predict the relationship between an example drug-effect pair in a sentence, the dependency paths and the words connecting the drug-effect pair would be the features used to learn the regression function. The final model $\psi_m = \psi_0 + \hat{\psi}_1 + \cdots + \hat{\psi}_m$ is a sum over these regression trees. Functional-gradient ascent is also known as functional-gradient boosting (FGB) due to this sequential nature of learning models based on the previous iteration.

But standard FGB assumes the examples have a flat feature representation. However, as mentioned earlier, each sentence can have structured features such as dependency path structure and parse trees leading to different number of features for every example in a flat representation. Relational models can handle data by using first-order logic representation. E.g., "prep_of" dependency between words "cause" and "MI" can be represented as prep_of(cause, MI).

FGB has been extended to relational models [14], [8], [9], [13] to simultaneously learn the structure and parameters of these models. Relational examples

are groundings/instantiations (e.g. drug-event(aspirin, headache)) of the predicates/relations (e.g. drug-event) to be learned. The ψ function is represented by relational regression trees (RRT)[1] which uses the structured data as input in the trees. A standard objective function used in RFGB is the log-likelihood and the probability of an example is represented as a sigmoid over the ψ function [14]. They showed that the functional gradient of likelihood w.r.t. ψ is

$$\frac{\partial log P(\mathbf{X} = \mathbf{x})}{\partial \psi(x_i)} = I(y_i = 1) - P(y_i = 1; x_i, Pa(x_i)) \tag{1}$$

which is the difference between the true distribution (I is the indicator function) and the current predicted distribution. A sample relational regression tree for $target(X)$ is shown in Figure 1.

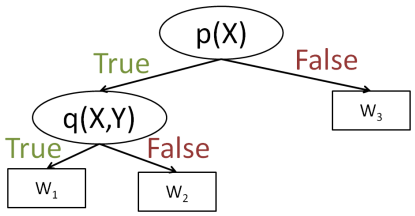

Fig. 1. Relational regression tree for a target predicate of interest, such as target(X) where p(X) and q(X, Y) are the features used. w_1 is the weight returned for target(x), if p(x) is true and q(x, Y) is true for some value of Y. X and Y are variables and can be instantiated with values such as "aspirin","headache" etc.

2.2 Relational Advice

While effective, the above method requires a large number of manually annotated examples. This translates to requiring a human to manually annotate every mention of a positive ADE pair and possibly several negatives. This is unreasonable and limits the human expert to be a *mere labeler*. It would be more practical for the human to provide some sort of advice. An example could be to "extract all positive ADEs even at the cost of some false positives". This is a cost-sensitive advice and we have explored this in the context of RFGB [18]. While effective, this advice is restricted to a trade-off between false positives and false negatives.

Human experts are capable of specifying richer advice. For instance, it is more reasonable to specify that *if the same sentence has an event word and a drug with a word cause somewhere in their path, then it is more likely that it is an adverse event*. We have recently developed a formulation based on RFGB that can handle such advice [15].

Our gradients contain an extra term compared to RFGB.

$$\Delta(x_i) = \alpha \cdot (I(y_i = 1) - P(y_i = 1; \psi)) + (1 - \alpha) \cdot [n_t(x_i) - n_f(x_i)]$$

where n_t is number of advice rules that prefer the example x_i to be true and n_f that prefer it to be false. Hence, the gradient consists of two parts: $(I - P)$ which is the gradient from the data and $(n_t - n_f)$ which is the gradient with respect to the advice.

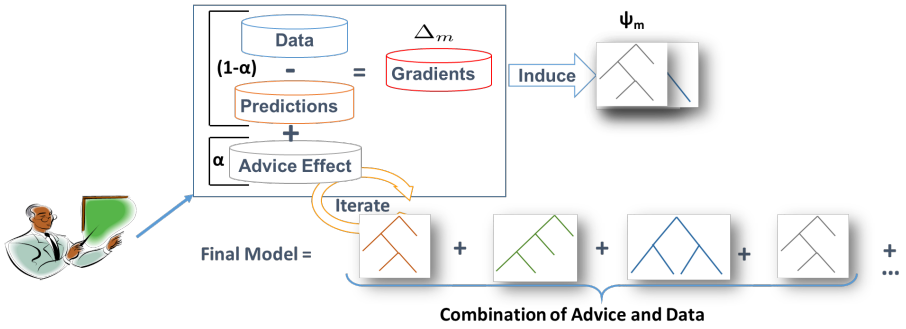

Fig. 2. Advice-based RFGB.

Figure 2 presents the advice-based RFGB approach. Intuitively when the example label is preferred in more advice models than the avoided target, $n_t(x_i) - n_f(x_i)$ is set to be positive. This will result in pushing the gradient of these examples in the positive direction (towards $+\infty$). Conversely when the example label should be avoided in more advice models, $n_t(x_i) - n_f(x_i)$ is set to be negative which will result in pushing the gradient of this example in the negative direction (towards $-\infty$). Examples where the advice does not apply or has equally contradictory advice, $n_t(x_i) - n_f(x_i)$ is 0. Hence, this approach can also handle *conflicting advice for the same example.*

Consider the adverse drug event prediction task using the dependency paths from sentences. A sample advice in our formalism is:

$$object(\text{"}cause\text{"}, event) \wedge agent(\text{"}cause\text{"}, drug) \rightarrow adverse(drug, event)$$

where *adverse* is the preferred label for this advice[4].

3 Proposed Approach

Our approach aims to predict whether there is evidence in the medical literature that a drug is known to cause a particular event. As there are very few examples that we are provided with compared to the number of features, our system incorporates domain knowledge that could be employed to identify text patterns in sentences that suggest an event is caused by a given drug. As mentioned previously, such a system can be used by other ADE predictors to evaluate

[4] $\wedge$ is used to represent AND in logic

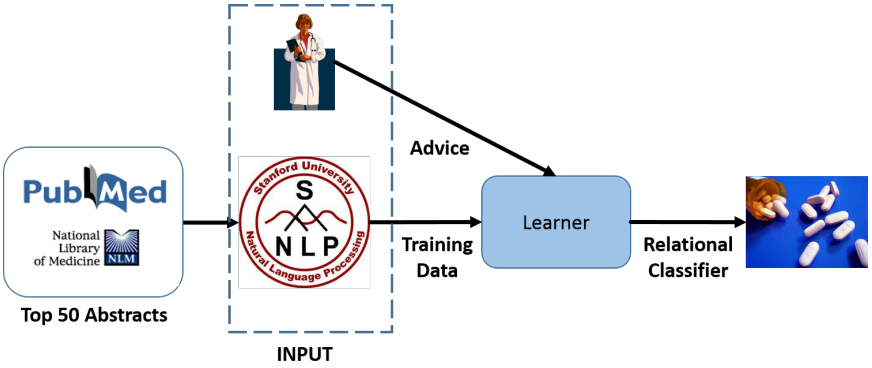

Fig. 3. Our proposed approach first finds medical abstracts that contain the drug and effect. Features are constructed by running them through the Stanford NLP parser. This data, along with the expert advice, is input to the learning algorithm.

whether they have identified a previously known drug and event. This would allow a knowledge-base of known ADEs that is constantly updated with the latest medical knowledge.

Figure 3 shows the process of training a model to predict ADEs. The first step of this process starts with searching PubMed, the standard database of medical publications, for abstracts that contain names for both the drug as well as the event. A sample query that we use is "Angioedema Renal Failure". We collect the first 50 abstracts for our deeper analysis. Previous empirical analysis [12] showed that 50 publications were sufficient and going beyond 50 to 75 or 100 did not statistically improve the results. While PubMed contains many articles of varying degrees of quality, we restrict the search to only articles that have been verified by PubMed (MEDLINE). If a drug and event have more than 50 results, then only the top 50 are extracted.

The abstracts for these drug and events are then passed to the Stanford NLP parser that generates facts (relational features) that represent the known medical knowledge for these drug and events. The specific features that we extract are parse trees, dependency paths, word lemmas and bag-of-word features. These are the standard features used in NLP literature and hence we employ them as well. The key reason for considering a relational representation is that the chances of two parse trees and/or dependency paths to look similar is minimal. Instead of carefully standardizing the features, relational models allow for learning using their natural representations. To summarize, for every ADE pair, the top 50 abstracts are parsed through the NLP parser and the corresponding features are then given as training data to the next step – the relational learning algorithm.

The learning algorithm has two sources of input: the training data and the expert domain knowledge. The training data is generated from a database (described in Section 4) of drug and event pairs that are either known to or known not to be ADEs. The second source of input is the expert domain knowledge.

This knowledge should capture the terminology by which medical experts express whether or not a drug and effect are related. For instance, "drug A causes event B" or "drug A is caused by event B" are two sample sentences that could have been used in abstracts. These are then used as advice to express that a drug causes a particular event. This knowledge is key to overcome the few training examples from which to learn. Note that soliciting advice is less costly than labeling more examples. We use 10 similar statements. For the purposes of this work, we as English speakers, served as the domain expert and wrote these rules. These rules were then used as advice by the learning algorithm for learning a set of relational regression trees that will serve as the model.

Once the learning phase is complete, the model can then be queried for inferring unseen ADEs from published, medical literature. This will become the test phase of our approach. Given, a new set of ADEs, the method automatically searches PubMed, constructs the NLP features and queries the model. The model in turn returns $P(ade(drug, event)|evidence)$ i.e., it returns the posterior probability of the drug-event pair being an ADE given the scientific evidence. Since all the evidence is observed, performing inference requires simply querying all the relational regression trees, summing up their regression values and returning the posterior estimates.

We must mention a few salient features of the proposed system (1) As more medical papers are published, the evidence of a drug causing an event can change and the system can automatically update its prediction resulting in an efficient refinement of medically known ADEs. (2) The nature of the formulation allows for contradicting and imprecise advice from domain experts. This allows for multiple experts to provide their inputs and our algorithm can automatically learn which of these are valid and which are not. (3) The use of richer advice enables for potentially weighing the different medical literature as well. For instance, it is possible to specify that "Journal X is more prestigious than Journal Y and hence trust it more than Y". This type of advice can also allow for specifying that more recent findings can potentially be more correct than older ones.

In summary, we have outlined a powerful system that allows for seamless human advice taking learning system that can automatically infer if a given drug-event pair has evidence in the literature to be an ADE.

4 Experiments

Our experimental results focus on three key questions:

Q1: How effective is the ADE extraction from text?
Q2: Can domain experts provide useful knowledge to extract evidence about ADEs from medical abstracts?
Q3: How effective is our method in incorporating advice into learned model?

Methods considered: We compare our method (called *Adv-RFGB* in the results) to three different baseline approaches. Both approaches, *MLN-Boost* and *RDN-Boost*, learn only from the data without considering the expert knowledge.

The third baseline that we considered is Alchemy[5], the state-of-the-art structure learning package for learning relational probabilistic models. The goal of this comparison is to establish the value of the expert knowledge (i.e., answer **Q2**).

Experimental Setup: The drug and event pairs come from Observational Medical Outcomes Partnership [6] 2010 ground truth, a manually curated database. To facilitate evaluation and comparison of methods and databases, OMOP established a common data model so that disparate databases could be represented uniformly. This included definitions for ten ADE-associated health outcomes of interest (HOIs) and drug exposure eras for ten widely-used classes of drugs.

Since this OMOP data includes very few positive examples (10 to be precise), we investigated other positive examples found in the literature to increase the training set. Our final dataset that we built contains 39 positive and 1482 negative examples (i.e.,39×38, the cross-product of all drug-effect pairs and obtained the ones that are not true ADE). The abstracts that we collected for the drug and event pairs contained 5198 sentences. Note that some drug and event pairs were not mentioned in any abstracts. In all experiments, we performed 4-fold cross validation. We compare both area under the curve for ROC and PR curves.

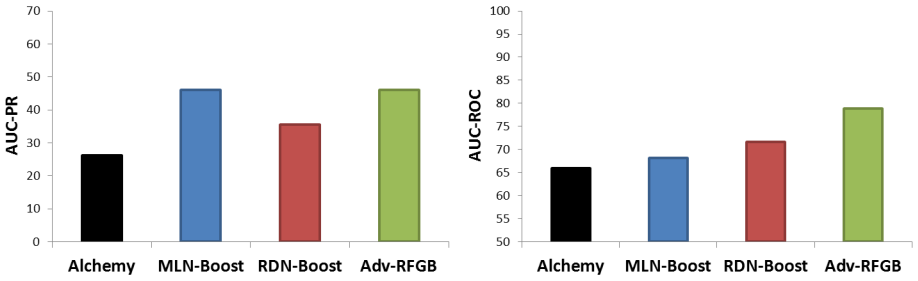

Fig. 4. Experimental results for predicting ADEs

Results: The results are presented in figure 4. The first three graphs present the results of using only data and employing standard relational learning methods. As can be seen, our proposed method that also employs "human advice" outperforms the three baselines that do not incorporate advice - (*RDN-Boost*, *MLN-Boost*, and *Alchemy*). This highlights the high value that the expert knowledge can have when learning with few training examples and thus answers **Q2**. **Q1** and **Q3** can also be answered affirmatively in that our proposed method is effectively learning with a high degree of accuracy to predict from the text abstracts. It is also clear that the advice is effectively incorporated when compared to merely using the data for learning and inference.

We investigated the differences between our predictions and the OMOP ground truth to understand whether our method was truly effective. One key

[5] alchemy.cs.washington.edu
[6] http://omop.org/

example where our method predicted an ADE pair to be positive, but OMOP labeled it as a negative ADE pair was: **Bisphosphonate** causes **Acute Renal Failure**. Our method predicted it as an ADE with a high (98.5%) probability. We attempted to validate our prediction and were able to find evidence in the literature to support our prediction. As an example, PubMed article (PMID 11887832) contains the sentence:

> **Bisphosphonates** have several important toxicities: **acute renal failure**, worsening renal function, reduced bone mineralization, and osteomalacia.

This suggests that our method (1) is able to find some evidence to support its prediction and (2) is capable of incorporating novel medical findings.

5 Discussion

Extracting ADEs from medical text has been an active area for recent research [7], [12]. Kang et al.'s method relied on a dictionary system to identify the drugs and effects in the sentence and a knowledge graph to semantically identify if any relationship was present between drug and effect. We allow human advice to guide our learning algorithm as opposed to using previously defined knowledge-bases. Natarajan et al. first use a human expert to define a full model that can just be queried and not learned. They use the expert advice as a prior and then refine that model according to the data. In comparison, we learn from human advice and training data jointly to learn a more robust model in the presence of noisy evidence. Our proposed approach builds upon a recently successful probabilistic learning algorithm that exploits domain knowledge. We adapted an NLP pipeline that allows for this learning method to search for PubMed abstracts, construct appropriate NLP features and learn a model by seamlessly taking human advice. Our experimental evaluation on the standard OMOP data set showed that this approach effectively and efficiently exploits human advice.

There are several possible directions for future work. In this work, we assume ADE pairs but extending this to multiple drugs and multiple events is not difficult and we plan to pursue this next. Also, we only consider abstracts but considering the full text of articles remains an interesting direction. As we have shown even with only abstracts, there is an imbalance in the number of examples vs the number of features. This dimensional disparity can potentially grow exponentially with full text. Extending our learning algorithms to handle this huge dimension is another direction. Finally, understanding if it is possible to unearth novel ADEs by "reading between lines" of text articles remains an exciting and potentially game-changing future direction of research.

Acknowledgements: SN and PO thank Army Research Office (ARO) grant number W911NF-13-1-0432 under the Young Investigator Program. SN gratefully acknowledges the support of the DARPA DEFT Program under the Air Force Research Laboratory (AFRL) prime contract no. FA8750-13-2-0039. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, ARO, AFRL, or the US government.

References

1. Blockeel, H.: Top-down induction of first order logical decision trees. AI Communications 12(1-2) (1999)
2. Cristianini, N., Shawe-Taylor: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press (2000)
3. Finkel, J., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. pp. 363–370. ACL '05, Association for Computational Linguistics (2005)
4. Friedman, J.: Greedy function approximation: A gradient boosting machine. In: Annals of Statistics (2001)
5. Gurwitz, J., Field, T., Harrold, L., J, R., Kebellis, K., Seger, A.: Incidence and preventability of adverse drug events among older persons in the ambulatory setting. JAMA
6. Gutmann, B., Kersting, K.: Tildecrf: Conditional random fields for logical sequences. In: ECML (2006)
7. Kang, N., Singh, B., Bui, C., Afzal, Z., van Mulligen, E.M., Kors, J.: Knowledge-based extraction of adverse drug events from biomedical text. BMC Bioinformatics 15 (2014)
8. Karwath, A., Kersting, K., Landwehr, N.: Boosting relational sequence alignments. In: ICDM (2008)
9. Kersting, K., Driessens, K.: Non-parametric policy gradients: A unified treatment of propositional and relational domains. In: ICML (2008)
10. Klein, D., Manning, C.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1. pp. 423–430. Association for Computational Linguistics (2003)
11. Mitchell, T.: Machine Learning. Mcgraw Hill (1997)
12. Natarajan, S., Bangera, V., Khot, T., Picado, J., et al.: A novel text-based method for evaluation of adverse drug event discovery. Journal of Biomedical Informatics (Under Review) (2015)
13. Natarajan, S., Joshi, S., Tadepalli, P., Kersting, K., Shavlik, J.: Imitation learning in relational domains: A functional-gradient boosting approach. In: IJCAI (2011)
14. Natarajan, S., Khot, T., Kersting, K., Gutmann, B., Shavlik, J.: Gradient-based boosting for statistical relational learning: The relational dependency network case. Machine Learning 86(1) (2012,)
15. Odom, P., Khot, T., Porter, R., Natarajan, S.: Knowledge-based probabilistic logic learning. In: AAAI (2015)
16. Ryan, P., Welebob, E., Hartzema, A.G., Stang, P., Overhage, J.M.: Surveying us observational data sources and characteristics for drug safety needs. Pharmaceutical Medicine pp. 231–238 (2010)
17. Schapire, R., Freund, Y.: Boosting: Foundations and Algorithms. MIT Press (2012)
18. Yang, S., Khot, T., Kersting, K., Kunapuli, G., Hauser, K., Natarajan, S.: Learning from imbalanced data in relational domains: A soft margin approach. In: ICDM (2014)