A simple PSL rule for geolocation of social-media posts is posted from or referring to a particular geographic location. To have to be accurately geolocated structures induced by rich, natural networks. For example, plates over entities and relationships in data, enabling inter-a common dependency in the data. These rules form tem-
are defined as a set of logical rules, each of which describes
of discrete and continuous information.

The rule relates sets of three first-order atoms, such as \( \text{HasLocation}(P, E) \) via probabilistic dependencies for each post \( P \) and named entity \( E \). Although the rule certainly does not always hold, PSL will combine it with other rules probabilistically to make predictions. Each rule is annotated with a non-negative weight, indicating how strongly the rule should hold in the data. To construct a ground model for specific data, each rule is grounded out by replacing the logical variables in the first-order atoms with constants from the data to construct a set of rules containing only ground atoms.

PSL treats the truth values of ground atoms as continuous variables in the \([0,1]\) interval, using soft logic relaxations of Boolean logic. This continuous representation easily incorporates naturally continuous quantities into its logic-based dependencies, allowing PSL to model and predict both continuous and discrete information by treating the soft truth values as either truly continuous quantities or confidences in discrete predictions. The resulting continuous-variable models are probability densities over the possible continuous truth-value assignments to the ground atoms. Each ground rule induced by a data set contributes a hinge-loss potential function to the graphical model, measuring how far the rule is from being logically satisfied for different assignments of truth values to the ground atoms. These models are members of a powerful class of graphical models known as hinge-loss Markov random fields \([2]\), which admit efficient inference and learning in fully-labeled settings as well as partially labeled settings with latent variables.

This intuitive modeling language, backed by scalable machine learning algorithms, makes PSL a flexible tool for social-data applications with the potential for positive impacts. Figure 1 shows an example PSL program for detecting disease outbreaks from social-media posts. The goal is to infer the prevalence of diseases in different locations for a given dictionary of locations and diseases from a corpus of social media posts. The first two rules geolocate posts. The atom \( \text{PostMentionsEntity}(P, E) \) can be grounded with substi-
tutions via entity recognition on the collection of posts, and \( \text{IsLocation}(E) \) can be grounded by substituting from the dictionary of locations. \( \text{PostIsGeotagged}(P, GT) \) can be granted by extracting any available latitude and longitude geotags from posts, and \( \text{GeotagInLocation}(GT, L) \) can be grounded via a mapping from geotags to the dictionary of loc-
ations. \( \text{HasLocation}(P, L) \) is unobserved, so the first two rules will be used to infer locations for posts. The locations of posts will then be combined with mentions of diseases in the third rule to infer disease prevalence based on disease mentions. The fourth rule propagates disease prevalence to nearby locations, where \( \text{Close}(L1, L2) \) is a continuous-valued measure of how geographically close locations are, making this rule propagate disease prevalence more strongly to nearby locations. Finally, the fifth rule acts as prior information, indicating that lower disease prevalence should be preferred in the absence of additional evidence. As additional evidence accumulates, the prior will have less influence on the prediction, and higher prevalence will be predicted.

Previous applications of PSL include detection of events such as disease outbreaks and civil unrest from social me-
rules from data, i.e.,
opposing new learning algorithms that are able to learn PSL inference via vertex programming. We are also developing our preliminary work on distributed computation for the efficient and convex form of PSL inference. We seek to extend our efforts to training from large-scale data that exploit making PSL more powerful. We are investigating fast learning algorithms are included for supervised and semi-supervised learning. We also have implemented an especially scalable backend for fast grounding of models. A variety of learning algorithms are included for supervised and semi-supervised learning. We also have implemented an especially scalable algorithm for inference that lazily constructs ground PSL models as it becomes necessary to actually reason about non-zero truth assignments to ground atoms. In relational domains, in which many possible relations do not actually exist, this can greatly improve scalability. The entire package is licensed under the Apache 2.0 license.

We have implemented an open-source software package for the PSL framework. The code is written in Java, so it is portable to a variety of platforms. A Groovy front-end layer is also implemented, allowing users to mix Java and a domain-specific language for easily defining PSL rules and constraints. The PSL package uses a relational-database backend for fast grounding of models. A variety of learning algorithms are included for supervised and semi-supervised learning. We also have implemented an especially scalable algorithm for inference that lazily constructs ground PSL models as it becomes necessary to actually reason about non-zero truth assignments to ground atoms. In relational domains, in which many possible relations do not actually exist, this can greatly improve scalability. The entire package is licensed under the Apache 2.0 license.

In addition to exploring new applications, we are actively researching a number of methodological directions toward making PSL more powerful. We are investigating fast learning algorithms for training from large-scale data that exploit the efficient and convex form of PSL inference. We seek to extend our preliminary work on distributed computation for PSL inference via vertex programming. We are also developing new learning algorithms that are able to learn PSL rules from data, i.e., structure learning. Another area we are investigating is using PSL as a decision-support system, helping to target positive interventions by modeling causal relationships. Finally, we are designing methods for learning and inference in streaming settings, in which the model and predictions must be continually—and efficiently—updated as new data arrives.

We have described PSL and some of its qualities that make it well suited to innovative applications of social data with the potential for positive social impacts. PSL’s intuitive modeling language and scalable algorithms enable it to work with large-scale social data. With our ongoing research in both algorithms and applications, we are eager to apply PSL to new social-good problems.

![Figure 1: A sample PSL program for disease-outbreak detection using social media.](http://psl.cs.umd.edu)

**Acknowledgments**

This work was supported by NSF grants CCF0937094 and IIS1218488, and IARPA via DoI/NBC contract number D12PC00337. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

**References**


