Towards Scalable Spatial Probabilistic Graphical Modeling

Ibrahim Sabek
Department of Computer Science and Engineering
University of Minnesota, USA
Email: sabek@cs.umn.edu, Advisor: Mohamed F. Mokbel, ACM Member Number: 5700939, SRC: Graduate

KEYWORDS

Spatial Probabilistic Graphical Models, Spatial Analysis, Markov Logic Networks, Scalability

1 INTRODUCTION

There is a plethora of spatial data being generated at the moment. For example, space telescopes generate up to 150 gigabytes weekly spatial data, medical devices produce spatial images (X-rays) at a rate of 50 petabytes per year, and a NASA archive of satellite earth images has more than 500 terabytes. This raises the need for efficient spatial analysis solutions to extract insights and useful patterns from such data. *Spatial probabilistic graphical modeling* (SPGM) represents an essential class of spatial analysis techniques, which exploits probability distributions and graphical representations (e.g., spatial hidden Markov models [5]) to describe spatial phenomena and make predictions about them [12]. SPGM has revolutionized many scientific and engineering fields in the past two decades including health care, risk analysis, and environmental science (e.g., [1, 5]). However, existing SPGM techniques have a scalability issue. In particular, they were originally designed for running on a single machine and hence suffer from the limited computation resources (e.g., see [3, 6, 14]). Such techniques can not scale beyond implementing prototypes over small spatial datasets.

Meanwhile, Markov Logic Networks (MLN) [9] was introduced to efficiently build complex learning and inference models over big data in a declarative manner. Basically, MLN combines first-order logic rules with probabilistic graphical models to represent statistical learning and inference problems with few logical rules (e.g., rules with imply and bit-wise AND predicates) instead of thousands of lines of code. With MLN, data scientists and developers can focus their efforts only on developing the rules that represent their applications (e.g., knowledge base construction, data cleaning, genetic analysis). Although the recent advances in MLN frameworks [8] helped to scale up the performance of typical spatial analysis applications (e.g., spatial regression [10], and spatial-aware knowledge base construction [11]), MLN was never exploited to scale up the performance of SPGM techniques.

In this paper, we propose *Flash*; a framework for scalable spatial probabilistic graphical modeling (SPGM) using Markov Logic Networks (MLN). *Flash* has the following three main features: (1) *Declarativity*: *Flash* expresses any SPGM application with logical semantics, and allows developers to implement it using a set of logical rules. (2) *Efficiency*: *Flash* translates the equivalent MLN rules of any SPGM application into SQL queries using an efficient grounding technique [13], and then executes these queries inside scalable database engines. In addition, *Flash* provides spatial variations of the RDBMS-based learning and inference algorithms of MLN [8] to perform scalable SPGM predictions (e.g., predictions over models with millions of nodes). (3) *Abstraction*: *Flash* allows developers to build a myriad of spatial analysis applications as a set of user-defined functions (UDF) without the need to worry about the underlying SPGM computation. As a case study, we equipped *Flash* with the implementation of three fundamental SPGMs; spatial Markov random fields (SMRF) [2], spatial hidden Markov models (SHMM) [5], and spatial Bayesian networks (SBN) [3]. The following sections explain the architecture of *Flash*, the implementation details of these three supported SPGMs, and the preliminary evaluation results.

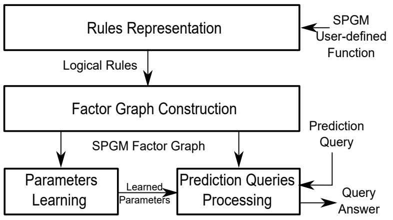
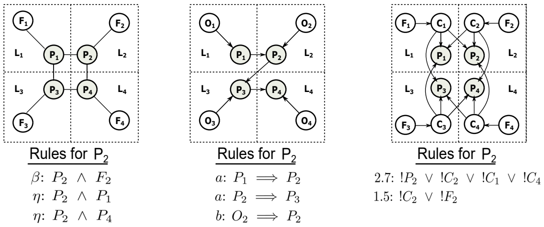


Figure 1: *Flash* System Architecture.

2 FRAMEWORK OVERVIEW

Flash adopts a modular system architecture as shown in Figure 1. It consists of four main modules, described briefly as follows:
Rules Representation. This module is responsible for generating an equivalent representation of logical MLN rules to any user-defined SPGM input. These rules have two main properties: (1) they contain first-order logical predicates (e.g., bitwise-AND, and imply) that capture the SPGM semantics; (2) they are associated with *weights* that represent the original SPGM parameters (Examples are in Section 3). The generated rules follow the syntax of a DBMS-friendly Datalog-like language, called DDlog [13], which can be efficiently processed with any relational DBMS (e.g., PostgreSQL) during the factor graph construction module.
Factor Graph Construction. This module takes the gerenated rules as input and uses them to build a *factor graph* [15] in a scalable way. The factor graph is the main data structure used to represent any MLN model, where the weights of the graph nodes correspond to the weights of rules (i.e., SPGM model parameters). To efficiently populate this factor graph, *Flash* adapts a scalable grounding technique from [13] that translates the generated rules into SQL queries, and then applies such queries on the input application data to obtain the final factor graph that is equivalent to the SPGM input.
Parameters Learning. This module learns the unknown weights of the constructed factor graph (i.e., weights of rules), which in turn specify the final SPGM parameters (e.g., spatial hidden Markov model [5] parameters). *Flash* proposes a pseudo-likelihood learning algorithm that adapts an efficient variation of a sampling-based



(a) SMRF (b) SHMM (c) SBN

Figure 2: SMRF, SHMM, and SBN Representations in *Flash*.

gradient descent optimization technique to compute the gradient of the SPGM pseudo-likelihood and then determine the weights.

Prediction Queries Processing. This module is responsible for answering prediction queries over the SPGM model (e.g., what is the probability of a specific event to happen?). Basically, it takes the prediction query along with the factor graph and its learned weights as inputs, and produces a prediction output associated with its confidence probability. Prediction queries can be answered using traditional Gibbs sampling-based inference algorithms over factor graphs [8]. However, such algorithms perform sequential sampling over the factor graph nodes which results in slow convergence to the inference answer in case these nodes have spatial dependencies as in SPGM applications [7]. Instead, *Flash* employs a variation of Gibbs Sampling that exploits a *concliques*-based traversal pattern [7] to efficiently sample spatially-dependent nodes in parallel while guaranteeing the rapid convergence.

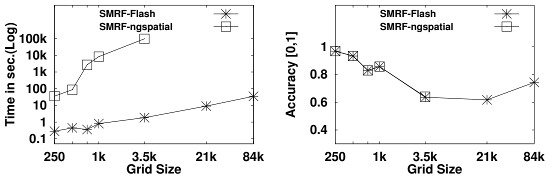
3 CASE STUDIES IN *FLASH*

Flash supports the implementation of three common spatial graphical models; spatial Markov random fields (SMRF) [2], spatial hidden Markov models (SHMM) [5], and spatial Bayesian networks (SBN) [3], as case studies. Figure 2 gives toy examples on the logical representation of these three models in *Flash*, where each model is defined over 4-cells grid, and the neighborhood of any cell l is assumed to be the cells that share edges with l only.

SMRF. Figure 2(a) shows a small SMRF model with a prediction P_l and feature F_l at each cell l. Each prediction P_l has undirected edges with feature F_l at this cell and each neighboring prediction variable. For example, P_2 is connected with feature F_2 and neighbors P_1 and P_4. *Flash* provides an equivalent weighted *bitwise-AND* predicate for each pair of connected variables, where these weights correspond to the SMRF parameters.

SHMM. Figure 2(b) shows a small SHMM model with a hidden state P_l and observation O_l variables at each cell l. Each observation O_l has a directed edge to state P_l at this cell. In addition, SHMM imposes an ordered spatial dependence among neighboring locations, where it uses z-curve ordering technique to build a sequence that preserves the spatial dependence between prediction variables (e.g., P_1 has a directed edge to P_2, and P_2 has another one to P_3, etc). *Flash* provides an equivalent weighted *imply* predicate for any state/state or observation/state pair, where these weights correspond to the SHMM parameters.

SBN. Figure 2(c) shows a small SBN model with a prediction variable P_l at each cell l which is affected directly by a status variable C_l and indirectly by a feature variable F_l (i.e., F_l has a direct edge to C_l). In addition, each prediction P_l is affected by the status variables at the neighboring cells. *Flash* provides an equivalent weighted combination of *bitwise-OR* and *negation* predicates for each causality relation (i.e., directed edge). The weights of these predicates are calculated from the input prior probabilities of SBN.



(a) *Flash* Scalability (b) *Flash* Accuracy

Figure 3: Initial Experiments of *Flash*.

4 PRELIMINARY RESULTS

We conducted initial experiments to evaluate the performance of *Flash*'s SMRF by building an autologistic regression model for a real dataset of the daily distribution of bird species [4], and compared its scalability and accuracy to the SMRF built by a base method, namely ngspatial [6]. Figure 3 shows the running time and accuracy for both systems while building the SMRF-based autologistic regression over grid sizes ranging from 250 to 84k cells. *Flash* has at least two orders of magnitude reduction in the running time over ngspatial, while preserving the same accuracy. Note that the ngspatial curve is incomplete after a grid size of 3.5k cells because of the extremely long running times that cause killing the running processes.

REFERENCES

[1] S. Balbi et al. A Spatial Bayesian Network Model to Assess the Benefits of Early Warning for Urban Flood Risk to People. *Natural Hazards and Earth System Sciences*, 2016.

[2] J. Besag. Spatial Interaction and the Statistical Analysis of Lattice Systems. *Royal Statistical Society Journal*, 1974.

[3] bnspatial: Spatial Implementation of Bayesian Networks. cran.r-project.org/web/packages/bnspatial, 2019.

[4] EBird Data. ebird.org/science/download-ebird-data-products.

[5] P. J. Green and S. Richardson. Hidden Markov Models and Disease Mapping. *JASA*, pages 1055–1070, 2002.

[6] J. Hughes. ngspatial: A Package for Fitting the Centered Autologistic and Sparse Spatial Generalized Linear Mixed Models for Areal Data. *The R Journal*, 2014.

[7] M. Kaiser et al. Goodness of Fit Tests for a Class of Markov Random Field Models. *The Annals of Statistics*, pages 104–130, 2012.

[8] F. Niu et al. Tuffy: Scaling Up Statistical Inference in Markov Logic Networks Using an RDBMS. *PVLDB*, 4(6):373–384, 2011.

[9] M. Richardson and P. M. Domingos. Markov Logic Networks. *Machine Learning*, pages 107–136, 2006.

[10] I. Sabek, M. Musleh, and M. Mokbel. TurboReg: A Framework for Scaling Up Spatial Logistic Regression Models. In *SIGSPATIAL*, pages 129–138, 2018.

[11] I. Sabek, M. Musleh, and M. F. Mokbel. A Demonstration of Sya: A Spatial Probabilistic Knowledge Base Construction System. In *SIGMOD*, 2018.

[12] S. Shekhar et al. Identifying Patterns in Spatial Information: A Survey of Methods. *WIRES: Data Mining and Knowledge Discovery*, pages 193–214, 2011.

[13] J. Shin et al. Incremental Knowledge Base Construction Using DeepDive. *PVLDB*, 8(11):1310–1321, 2015.

[14] shmm: An R Implementation of Spatial Hidden Markov Models. github.com/mawp/shmm, 2019.

[15] M. Wick et al. Scalable Probabilistic Databases with Factor Graphs and MCMC. *PVLDB*, 3(1-2):794–804, 2010.