

Dynamic Domain Sizes in Temporal Probabilistic Relational Models

Nils Finke and Tanya Braun and Marcel Gehrke and Ralf Möller

Institute of Information Systems, University of Lübeck, 23562 Lübeck, Germany
{finke, braun, gehrke, moeller}@ifis.uni-luebeck.de

Abstract

Probabilistic dynamic relational models (PDRMs) allow for an expressive, yet sparse and efficient representation of *uncertain* temporal (*dynamic*) and *relational* information with a fixed (*static*) set of domain objects (entities). While for different points in time, information about objects may differ, the set of objects under consideration is the same for all time points in standard PDRMs. Motivated by examples from a logistics application, in this paper we extend the theory of PDRMs with *dynamically changing sets of domain objects*. The paper introduces the semantics of so-called PD²RM and analyses model management as well as query answering problems and algorithms.

1 Introduction

In order to cope with uncertainty and relational information of numerous objects over time, in many real-world applications, probabilistic temporal (also called dynamic) relational models (PDRMs) need often be employed (Finke et al. 2020). The sizes of sets of domain objects determine the overall joint distribution behind a PDRM unrolled over time. While for different points in time, information about objects may differ, the set of objects under consideration is the same for all time points in standard PDRMs. Providing semantics for dynamically changing sets of domain objects is important for modeling real-world applications, as it is very hard to foresee in advance about which objects an intelligent system needs to reason. The problem we face in this paper is to define the semantics of a PDRM formalism with dynamically changing sets of domain objects to be substituted for variables used in the model, such that sparse domain models can be specified and efficient query answering algorithms can be provided for practically relevant application domains.

Research about the effect of changing sets of domain objects on overall probability distributions of static relational models has provided partial solutions. If the model’s full joint distribution keeps being valid with a change in the domain, models are called *projective*. Based on the work of Shalizi and Rinaldo (2013), Jaeger and Schulte (2018) define conditions, when predictions are robust to changes in the domain size. For non-projective models, various research about

how to scale probability distributions with changes in the domain size have been performed (Jain, Barthels, and Beetz 2010; Weitkämper 2020; Poole et al. 2014). The notion of non-stationarity has indeed been discussed in the literature very often. However, non-stationarity as found in the literature refers to changes in distributions (and respective factors in a model). We extend the theory of probabilistic dynamic relational modeling with *dynamically changing sets of domain objects over time*. The paper introduces the semantics of so-called PD²RM and analyses model management as well as query answering problems and algorithms.

2 Preliminaries

We recapitulate PDRMs (Gehrke, Braun, and Möller 2018) in context of an example in logistics, specifically, shipping.

2.1 Parameterized Dynamic Models

PDRMs combine first-order logic with a factor graph, representing first-order constructs using logical variables (logvars) as parameters. Random variables (randvars) are parameterized with logvars (PRV) to compactly represent sets of randvars considered indistinguishable without evidence.

Definition 1 Let \mathbf{R} be a set of randvar names, \mathbf{L} a set of logvar names, Φ a set of factor names, and \mathbf{D} a set of entities. All sets are finite. Each logvar L has a domain $\mathcal{D}(L) \subseteq \mathbf{D}$. A constraint is a tuple $(\mathcal{X}, C_{\mathbf{X}})$ of a sequence of logvars $\mathcal{X} = (X_1, \dots, X_n)$ and a set $C_{\mathcal{X}} \subseteq \times_{i=1}^n \mathcal{D}(X_i)$. A PRV $R(L_1, \dots, L_n), n \geq 0$ is a construct of a randvar $R \in \mathbf{R}$ possibly combined with logvars $L_1, \dots, L_n \in \mathbf{L}$. The term $\mathcal{R}(A)$ denotes the possible values (range) of a PRV A .

PRVs are linked through parametric factors (parfactors), which are functions that take PRVs as arguments to represent their relation, and return a real number, called potential.

Definition 2 We denote a parfactor g by $\phi(\mathcal{A})|_C$ with $\mathcal{A} = (A^1, \dots, A^n)$ a sequence of PRVs, $\phi : \times_{i=1}^n \mathcal{R}(A^i) \mapsto \mathbb{R}^+$ a function with name $\phi \in \Phi$, and C a constraint on the logvars of \mathcal{A} . A PRV A or logvar L under constraint C is given by $A|_C$ or $L|_C$, respectively. The term $gr(P)$ denotes the set of all instances of P . An instance is a grounding of P , substituting the logvars in P with a set of entities from the constraints in P . The term $lv(P)$ refers to logvars in P .

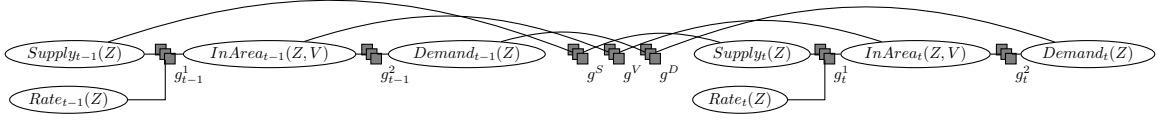


Figure 1: Two-slice parameterized probabilistic model G_{\rightarrow} (and G_0)

Like most dynamic model formalisms, PDRMs use two static parameterized models (PRMs) to describe how a model changes from one time step to the next. Changes from one time step to the next follow the same distribution. A PDRM encodes a sequential dimension by a pair of PRMs, one representing an initial time step and the other representing how the model transitions from one time step to the next. Semantics of a PDRM are given by instantiating a PDRM for a given number of time steps using G_0 for the initial time step and appending G_{\rightarrow} for the other time steps, followed by grounding and building a full joint distribution.

Figure 1 shows a PDRM illustrating seaborne transportation. Variable nodes (ellipses) correspond to PRVs, factor nodes (boxes) to parafactors. Parafactors g^S , g^V , and g^D are so called inter-slice parafactors. The submodel to the left and to the right of these inter-slice parafactors are duplicates of each other, with the left referencing time step $t - 1$ and the right referencing time step t . Parafactors reference time-indexed PRVs, namely, a Boolean PRV $InArea(Z, V)$ and PRVs $Supply(Z)$, $Rate(Z)$, $Demand(Z)$ with range values $\{high, medium, low\}$, built from randvar names $\mathbf{R} = \{Supply, Rate, InArea, Demand\}$ and logvar names $\mathbf{L} = \{Z, V\}$. Seaborne transportation using vessels is mainly driven by supply ($Supply(Z)$) and demand ($Demand(Z)$) of commodities across various locations (zones Z). Vessels V move between these zones, captured by $InArea(Z, V)$, representing trade flows: Vessels are in zones with high supply (to load cargo), in zones with high demand (to discharge cargo), and in between while traveling. For transportation, a fee per ton, called freight rate ($Rate(Z)$), is charged.

Joining all parafactors within the model using a join over the arguments and multiplication for the potentials leads to a full joint probability distribution after normalising the join result (Sato 1995). Given a PDRM, one can ask queries for probability distributions or the probability of an event, possibly given a set of observations as evidence, like $P(Rate(z_1) \mid Supply(z_2) = high, Supply(z_3) = high)$. Here, $Supply(z_2) = high$ and $Supply(z_3) = high$ denote evidence. Sets of parafactors can encode evidence, one parafactor for each subset of evidence that concern one PRV with the same observation.

Definition 3 A parfactor $g_e = \phi_e(E(X))|_{C_e}$ specifies evidence for a set of events $\{E(x_i) = o\}_{i=1}^n$ of a PRV $E(X)$. The function ϕ_e maps the value o to 1 and the remaining range values of $E(X)$ to 0. Constraint C_e encodes the observed groundings x_i of $E(X)$, i.e., $C_e = (X, \{x_i\}_{i=1}^n)$.

3 Dynamic Domains

As follows, we characterize and discuss, how entity changes can be incorporated in PDRMs, yielding PD²DRMs.

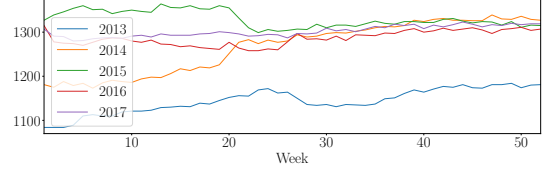


Figure 2: Changing vessel counts in dry-bulk shipping.

3.1 Characteristics of Domain Size Changes

In shipping, vessels are continuously joining or leaving the market. Figure 2 shows the number of vessels over time of a specific ship class (*Capesize* vessels) available on the market. The lines indicate continuous changes in the number of vessels due to a variety of different reasons, such as vessels are newly build, dumped, going into maintenance, or are laid up in case freight rates do not cover running costs any more. We assume that these domain changes are directly observable and are not affected by uncertainty. In general, we differentiate between the following cases: An entity (here, a vessel) is (a) fully removed from the model (*Removal*), (b) either completely new to the model (*Addition*), or (c) is temporally removed and at a later stage again added to the model (*Restoration*).

3.2 Challenges with Domain Size Changes

In general, domain size changes can be encoded within PDRMs without changing the model structure itself as PRVs, in simple terms, use a single representative randvar for all entities within the model. In contrast, in the propositional case, randvars are duplicated for entities of the same type. Thus, encoding changing domain sizes within PDRMs relates to changes in the domain represented through the logvars of a model. Each logvar L traditionally represents a *fix* set of entities $D(L)$ (its domain). To allow for a *dynamic* domain, thus for changes (in- or decrease) in the set of entities $D(L)$, it is required to adopt the model’s parfactor constraints whenever a change in the domain occurs. Without any splits in the model (due to no new evidence after the creation of the model), new entities easily integrate as no further changes are necessary. The challenge of incorporating entity changes in relational models comes with evidence introducing splits in PDRMs. With splits, constraints ($\mathcal{X}, C_{\mathcal{X}}$) on parafactors have to be adjusted as they limit the domain of logvars $\mathcal{X} = (X_1, \dots, X_n)$ to a certain set of entities $C_{\mathcal{X}} \subseteq \times_{i=1}^n D(X_i)$. We define models splits as follows.

Model Splits Suppose we observe a vessel at a certain position ($InArea_t(z_1, v_1) = true$). Then evidence is encoded in the parafactors g_t^1 and g_t^2 by duplicating each of

the parfactors and use one to encode evidence and one to represent all sets of entities, which are still considered indistinguishable. Each parfactor represents a different set of entities limited by the use of constraints, e.g., limiting the domain for the evidence parfactor to z_1 and v_1 . The parfactor that encodes evidence is adjusted so that all lines of range value combinations in the parfactored distribution ϕ are dropped for $InArea(z_1, v_1) \neq true$. Formally, for a model $G_t = \{g_t^i\}_{i=1}^n$ at timestep t and with evidence, the model G_t is split w.r.t. its parfactored such that its structure remains:

$$G_t = \{g_t^{i,1}, \dots, g_t^{i,k}\}_{i=1}^n, \quad k \in \mathbb{N}^+ \quad (1)$$

Every parfactor g_t^i can have up to $k \in \mathbb{N}^+$ splits

$$g_t^{i,j} = \phi_t^{i,j}(\mathcal{A}^i)_{C^{i,j}}, \quad 1 \leq j \leq k \quad (2)$$

with \mathcal{A}^i as a sequence of the same PRVs but with different constraint $C^{i,j}$ and varying functions $\phi_t^{i,j}$ due to evidence.

A challenge that arises from a model as given in Eq. (1) is to *add* a new entity $x_z \in D(X \cup \{x_z\})$ to the model, considering additional knowledge encoded in the model as given in Eq. (2). In simple terms, it is to decide and identify if and in case to which *group*, denoted by a parfactor split, the entity can be added. In general, a new entity can either be added to one of the existing parfactor groups (Eq. (2)) or can be separated by introducing further parfactor groups. Adding new entities to a parfactor group comes done to adjust parfactor constraints, which is defined as follows.

Adding Entities Adding an entity x_z to a parfactor group $g_t^{i,j} = \phi_t^{i,j}(\mathcal{A}^i)_{C^{i,j}}$, requires to change the constraint $C^{i,j}$ limiting all logvars $L \in lv(\mathcal{A})$. Thus, for all $L \in lv(\mathcal{A})$ with $C^{i,j} = (L, \{x_i\}_{i=1}^n)$ include x_z such that $C^{i,j} = (L, \{x_i\}_{i=1}^n \cup \{x_z\})$. In further course of this paper, we are using the term *adding to a parfactor group* as an equivalent for adjusting the constraint of a parfactor group.

When observing an entity change, e.g., a new entity x_z , further *additional evidence* for that specific entity x_z can be present. If such additional evidence $E_t(x_z) = o$ for any PRV $E_t(X)$ in G_t is available, it can be used to identify one of the parfactor groups, representing entities $x_i \in X$ behaving similar to x_z . In the following, we define an auxiliary construct, called parfactor partitions, over sets of parfactored \mathbf{P} in G_t , denoting groups with similarities to x_z .

Model Partitioning Every parfactor $g_t^i \in G_t$ can have up to $k \in \mathbb{N}^+$ splits as given in Eq. (1). Each parfactor g_t^i contains a sequence of PRVs $\mathcal{A} = (A^1, \dots, A^n)$, which can be affected by evidence by having observations $A(x_i) = o$ for any entity $x_i \in D(X) \setminus \{x_z\}$ over PRVs $A \in \mathcal{A}$, leading to those splits. A parfactor partition $P_t^{E(x_z)}$ under evidence E for x_z is defined as a subset of those parfactor groups

$$P_t^{E(x_z)} = \{g_t^{i,1}, \dots, g_t^{i,l}\}_{i=1}^n \quad (3)$$

with $l \leq k$ having matching evidence between x_z and $x_i \in D(X) \setminus \{x_z\}$ w.r.t. PRVs in \mathcal{A} . More specifically, if an additional observation o (evidence) for x_z over any PRV E is available ($E(x_z) = o$), and similarly the same observation o has already been made for any $x \in D(X) \setminus \{x_z\}$ (which

has lead to a parfactor split), $P_t^{E(x_z)}$ denotes all matching parfactored. Formally, for at least one PRV $E \in \mathcal{A}$ for all $g_t^{i,l} \in P_t^{E(x_z)}$, it is $E(x_i) = E(x_z)$ with $x_z, x_i \in D(X)$ and $X \in lv(E)$. Note, that the initial parfactor distribution (as if no evidence have been observed yet), is as well part of the partition $P_t^{E(x_z)}$ as this parfactor group will as well match evidence. Let $g_t^{i,1}$ here always denote the initial parfactor distribution of the model. Further, it is to be noted that when adding a new entity x_z to the model, additional evidence for x_z might not be limited to just one observation o . In case of having multiple observations, e.g., $E(x_z) = o$ and $F(x_z) = p$, an additional parfactor partition is created, such as $P_t^{E(x_z)}$ and $P_t^{F(x_z)}$. Let \mathbf{P}_t denote the set of all partitions. Next, we introduce adoption to domain size changes utilizing parfactor partitions.

3.3 Adapting to Domain Size Changes

We handle cases as defined in Section 3.1 and give advice and highlight limitations based on options for new entities.

Removal Removing an entity x_z from the model denotes an easy case as it comes down to adjusting all constraints, limiting parfactored groups to the entity x_z . Formally, finding all constraints $(\mathcal{X}, C_{\mathbf{X}})$ on logvars \mathcal{X} , such that for all $X \in \mathcal{X}$ it is $x_z \in D(X)$, and adjusting them to remove the entity accordingly. After removing an entity from a parfactor group by adjusting constraints and one of those adjusted constraints *becomes empty*, the parfactor group is similarly removed. A constraint becomes empty in case a constraint only limits a parfactor group to x_z before its removal.

Entities might only be temporarily removed from the model, thus, we do *bookkeeping*, to keep the state of the entity, before its removal, in memory. Therefore, we extract all observations made for x_z after the initial model setup, and store them for later use. Extracting evidence for x_z refers to extracting evidence encoded in the model's parfactor groups.

Addition Adding a new entity x_z splits into the two cases, whether x_z comes (a) *with additional evidence*, or (b) *without additional evidence*. Following, we look at both cases.

In case of (a) having *additional evidence* $E_t(x_z) = o$ for a new entity $x_z \in D(X \cup \{x_z\})$ for a PRV $E_t(X)$ at timestep t , we can make use of partitions P as per Eq. (3). With including x_z in the model, one has to ensure to group x_z into all parfactored $g_t^i \in G_t$, containing the logvar $X \in lv(g_t^i)$ as $x_z \in D(X)$. For adding under additional evidence, we discuss the following three approaches: (i) Group x_z into the evidence parfactor group $g_t^{i,l} \in P_t^{E(x_z)}$, in which most of the other entities are organized (denoted as *Max Evidence*), or (ii) create a new parfactor group $g_t^{i,l+1}$ by weighting the initial parfactor distribution $g_t^{i,1}$ based on other evidence parfactor groups in $P_t^{E(x_z)}$ (denoted as *Weighted Initial*), or (iii) add x_z into the initial parfactor distribution $g_t^{i,1}$ (denoted as *Initial*).

For (i), let $|g_t^{i,l}|$ define the number of entities w.r.t. the same logvar type $x_z \in X$ represented by $g_t^{i,l} \in P_t^{E(x_z)}$. As a parfactor may contain more than just one logvar, $|g_t^{i,l}|$ de-

finer the number of entities of logvar X in $g_t^{i,l}$. A new entity x_z , added at timestep t and afflicted with further evidence $E_t(x_z) = o$, is grouped into all i parfactored partitions $P_t^{E(x_z)}$ as given in Eq. (3). In case of having splits for g_t^i , the new entity x_z is put into the group $g_t^{i,l} \in P_t^{E(x_z)}$ excluding group $g_t^{i,1}$, such that for all remaining groups, it is $\max(|g_t^{i,l}|)$ with $l > 1$. As $g_t^{i,1}$ denotes the initial parfactor, it is specifically excluded here. Only in case of having no splits for g_t^i , the new entity x_z is added to $g_t^{i,1}$ since no other option is available (Approach (iii)). This is similarly done for all i parfactored partitions $P_t^{E(x_z)}$. We name the procedure *grouping-in* for easier reference. In case of having more than one additional observation for x_z , all partitions in P_t have to be handled according to the above grouping-in procedure. Evidence parfactor groups over partitions in P_t are always disjoint in terms of the entities they represent, except for all parfactor groups $g_t^{i,1}$. Having no splits in a parfactor denotes an edge case, requiring to further check if $g_t^{i,1}$ may already represent x_z . The parfactor group $g_t^{i,1}$ may only already represent x_z , if while adding x_t to the model, more than one further observation (evidence) is available. If so, the above grouping-in procedure runs m times, with m denoting the number of further observations for x_z . Thus, in terms of having no splits in parfactor, we have to ensure, that x_z is not put into $g_t^{i,1}$ multiple times. In general, this approach is promising as we potentially have observed how other entities behave in context of the model, and accordingly assume that new entities may behave similarly.

For (ii), a new parfactor group $g_t^{i,j+1}$ is created based on the initial parfactor distribution $g_t^{i,1}$ weighted by all matching evidence parfactor groups as per Eq. (3). In comparison to the first approach, this approach is relaxing the assumption that new entities may behave similarly to already existing entities, and allows for behaviour as defined by the initial distribution by weighting. To weighting the potentials of the initial distribution, we first accumulate all matching evidence parfactored $g_t^{i,l}$ with $l > 1$ of the partition $P_t^{E(x_z)}$, normalize the aggregated evidence parfactor to derive the probability distribution and use these probabilities as weights on the initial distribution potentials. More specifically, for a new entity x_z with observation $E(x_z) = o$ an aggregated evidence parfactor $g_t^{i,agg} \in P_t^{E(x_z)}$ is created by building a joint distribution over evidence parfactor $g_t^{i,j}$, $1 < j < l$. For evidence parfactored all range values (potentials) matching the observation o are set to 1 and all remaining range values are set to 0. Thus, the joint distribution of the aggregated evidence parfactor only refers to range values for which an observation is available. The aggregated evidence parfactor $g_t^{i,agg}$ and its probability distribution can then be used to weight the potentials of the initial parfactor distribution $g_t^{i,1}$, thus to create a new parfactor distribution group $g_t^{i,j+1}$ for x_z with $\phi_t^{i,j+1}(\mathcal{A}^i)_{C^{i,j+1}} = g_t^{i,1} \cdot g_t^{i,agg}$.

Similar to case (i), if having multiple partitions due to multiple observations for x_z , all partitions in P_t have to be handled according to this grouping-in procedure. This as well includes handling of the edge case described in (i).

In case of (b) having *no additional evidence* for x_z , we assume that the initial distribution (iii) of the parfactor $g_t^{i,1}$ denotes the best fit for the new entity.

Restoration Restoration, thus adding an entity x_i , which was previously already part of the model, back to it, denotes a special case w.r.t. the *Addition*-case. When restoring an entity x_i , we can potentially make use of past evidence, observed for the entity x_i . Therefore we make use of the books established while doing bookkeeping during removal. Restoring historical evidence is simply done, by applying again the set of evidence resulting from bookkeeping.

4 Conclusion and Outlook

This paper introduces the semantics of PD²RMs, covering over time changing domains in relational dynamic probabilistic models. As part of the formalism, we address the challenge of new evidence, further forming models if time proceeds. As such, we are the first providing semantics for changing domain objects, discussing on how to apply and transfer historical knowledge to new entities to enable for more accurate query answering related to those changes. In contrast, in existing research the effect of changing domain sizes on the probability distribution itself has been addressed. Moving forward, we run an empirical evaluation testing the proposed approaches to validate accuracy of each strategy. Further, scaling probability distributions as proposed in research have to be addressed.

References

- Finke, N.; Gehrke, M.; Braun, T.; Potten, T.; and Möller, R. 2020. Investigating maturity of probabilistic graphical models for dry-bulk shipping. In Jaeger, M., and Nielsen, T. D., eds., *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, 197–208. PMLR.
- Gehrke, M.; Braun, T.; and Möller, R. 2018. Lifted Dynamic Junction Tree Algorithm. In *Proceedings of the International Conference on Conceptual Structures*, 55–69. Springer.
- Jaeger, M., and Schulte, O. 2018. Inference, learning, and population size: Projectivity for srl models.
- Jain, D.; Barthels, A.; and Beetz, M. 2010. Adaptive markov logic networks: Learning statistical relational models with dynamic parameters. In *ECAI*, volume 215, 937–942.
- Poole, D.; Buchman, D.; Kazemi, S. M.; Kersting, K.; and Natarajan, S. 2014. Population size extrapolation in relational probabilistic modelling. In Straccia, U., and Cali, A., eds., *Scalable Uncertainty Management*, 292–305. Cham: Springer International Publishing.
- Sato, T. 1995. A Statistical Learning Method for Logic Programs with Distribution Semantics. In *Proceedings of the 12th International Conference on Logic Programming*, 715–729. MIT Press.
- Shalizi, C. R., and Rinaldo, A. 2013. Consistency under sampling of exponential random graph models. *The Annals of Statistics* 41(2):508–535.
- Weitkämper, F. 2020. Scaling the weight parameters in markov logic networks and relational logistic regression models.