POLITECNICO DI TORINO
Repository ISTITUZIONALE

A probabilistic semantics for process mining

Original
A probabilistic semantics for process mining / Vespa, M.. - ELETTRONICO. - 3914:(2024), pp. 1-6. (AIxIA Doctoral Consortium 2024 co-located with the 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024) Bolzano, Italy 25 novembre 2024).

Availability:
This version is available at: 11583/3016403 since: 2026-10-07T13:37:36Z

Publisher:
Sun SITE Central Europe

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

20 October 2026

A probabilistic semantics for process mining

Michela Vespa[1]

[1]Dipartimento di Ingegneria, Università di Ferrara, Via Saragat 1, Ferrara, Italy

Abstract

In declarative Process Mining (PM), accounting for uncertainty is essential to accurately model real-world business processes. Up to now, most traditional approaches have overlooked the possibility of integrating probability into process management. Starting from our previous works on this topic, we present an extension to our semantics that underlies a probabilistic declarative framework for PM, in such a way that we can manage uncertainty at multiple levels, from individual events to entire logs, by assigning probabilities reflecting a degree of belief or confidence in them. This framework is based on the Distribution Semantics of Probabilistic Logic Programming.

Keywords

Process Mining, Declarative language, Distribution Semantics, Probability theory

1. Introduction

Ongoing research in Process Mining (PM) is increasingly focusing on the role of uncertainty in business process management. Uncertainty in PM can manifest in multiple aspects of a process, ranging from process models to process data, i.e. events and event attributes, traces, and logs. For instance, real-world event logs may contain incomplete or noisy data, where some events/traces are missing or misrecorded. Various approaches have been explored to address uncertainty in *procedural* PM settings, dealing with structured, sequential process models, typically represented as flow-based notations like Petri nets or BPMN diagrams. In recent years, significant research built on foundational work by Pegoraro and Van der Aalst has been devoted to address this challenge with respect to event data [1]. This has been achieved through a framework designed to represent the control-flow dimension of uncertain events as Petri nets, involving stochastic process modeling techniques like stochastic Petri nets [2], behavioral nets [3, 4], and trace alignment [5, 6]. This research highlighted the complexities of managing uncertain event data within procedural models, focusing on strong uncertainty (unknown probability distributions for attribute values) at the *attribute level* of events.

However, a distinct approach can be taken when dealing with uncertainty in *declarative* PM, which focuses only on the constraints between activity sequences, rather than outlining exact workflows [7, 8]. For example, [9] introduced the notion of probabilistic process constraints, by associating probabilities to Declare constraints.

Starting from our previous work based on probabilistic declarative process specifications [10]

Doctoral Consortium at the 23rd International Conference of the Italian Association for Artificial Intelligence Bolzano, Italy, November 25-28, 2024

michela.vespa@unife.it (M. Vespa)

0009-0004-4350-8151 (M. Vespa)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-3914/short73.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

and probabilistic events [11], here we extend the underlying semantics in order to comprehensively handle uncertainty at all levels of *process data*, from traces to entire logs. This semantics is inspired by the Distribution Semantics (DS) [12] of Probabilistic Logic Programming (PLP) and handles uncertainty by expressing probabilities as a *degree of belief* (taking inspiration from [13],[14]) in traces and logs.

In [11], we treat uncertain events in a process trace by annotating them with a probability expressing the user's confidence in that event(s) happening. Here, we complete this framework by considering probabilities attached to traces as a whole, which results in *probabilistic logs*. For instance, there might be cases where a maintenance task, composed of different phases (the trace's events), is not logged correctly due to human error or system issues. If a technician recalls performing it but later finds no documentation of this in the system, he might estimate, based on his memory, interactions with colleagues, and standard operating procedures, that there is a 95% probability the inspection was completed as required. This would generate two possible logs, one with the trace included, with 0.95 probability, and the other log without the trace, but much less probable (0.05 probability). To the best of our knowledge, previous efforts in procedural PM have addressed either event-based or traced-based uncertainty separately, while our approach is new in handling probabilities from events to logs, offering an integrated semantics to manage and interpret uncertainty in process data at all levels of granularity.

2. Background: Distribution Semantics

PLP, notably through the Distribution Semantics, handles uncertain information by allowing probabilities in logic programs, which define probability distributions over a set of possible normal logic programs called "*worlds*" . In the following, the DS will be described with reference to the language of LPADs (Logic Programs with Annotated Disjunctions) [15], even if it underlies many other languages. A detailed survey of the DS in PLP can be found in [16]. In LPADs each program clause has a disjunction in the head with each atom annotated by a probability. When the clause body holds true, only one head atom is selected together with its probability.

An annotated disjunctive clause C_i is of the form $h_{i1} : p_{i1}; \ldots ; h_{in_i} : p_{in_i} :- b_{i1}, \ldots, b_{im_i}$, where $h_{i1}, \ldots, h_{in_i}$ are logical atoms and $\{p_{i1}, \ldots, p_{in_i}\}$ are real numbers in the interval $[0, 1]$ such that $\sum_{k=1}^{n_i} p_{ik} \leq 1$; $b_{i1}, \ldots, b_{im_i}$ is indicated with $body(C_i)$. If $\sum_{k=1}^{n_i} p_{ik} < 1$, the head implicitly contains an extra atom $null$ that does not appear in the body of any clause and whose annotation is $1 - \sum_{k=1}^{n_i} p_{ik}$. We denote by $ground(T)$ the grounding of an LPAD T.

An *atomic choice* [17] is a triple (C_i, θ_j, k) where $C_i \in T$, θ_j is a substitution that grounds C_i and $k \in \{1, \ldots, n_i\}$ identifies one of the head atoms. (C_i, θ_j, k) means that, for the ground clause $C_i\theta_j$, the head h_{ik} was chosen. A set of atomic choices κ is *consistent* if only one head is selected from the same ground clause; we assume independence between the different choices. A *composite choice* κ is a consistent set of atomic choices [17]. The *probability* $P(\kappa)$ *of a composite choice* κ is the product of the probabilities of the independent atomic choices, i.e. $P(\kappa) = \prod_{(C_i,\theta_j,k)\in\kappa} p_{ik}$. A *selection* σ is a composite choice that, for each clause $C_i\theta_j$ in $ground(T)$, contains an atomic choice (C_i, θ_j, k). Let us indicate with S_T the set of all selections. A selection σ identifies a normal logic program w_σ defined as $w_\sigma = \{(h_{ik} \leftarrow body(C_i))\theta_j | (C_i, \theta_j, k) \in \sigma\}$. w_σ is called a (possible) *world* of T. Since selections are composite choices, we can assign a

probability to worlds: $P(w_\sigma) = P(\sigma) = \prod_{(C_i,\theta_j,k)\in\sigma} p_{ik}$.

We denote the set of all worlds of T by W_T. $P(W_T)$ is a probability distribution over worlds, i.e., $\sum_{w\in W_T} P(w) = 1$. A composite choice κ identifies a set of worlds $w_\kappa = \{w_\sigma | \sigma \in S_T, \sigma \supseteq \kappa\}$. The set of possible worlds associated to a set of composite choices K is $W_K = \bigcup_{\kappa\in K} w_\kappa$.

Example 1. *Consider the following LPAD T encoding the outcome of tossing a coin, which may be either fair or biased:*

$$
\begin{array}{ll}
(C_1) & heads(Coin) : 0.5; tails(Coin) : 0.5 : -toss(Coin), \backslash\!+\!biased(Coin). \\
(C_2) & heads(Coin) : 0.6; tails(Coin) : 0.4 : -toss(Coin), biased(Coin). \\
(C_3) & fair(Coin) : 0.9; biased(Coin) : 0.1. \\
(C_4) & toss(coin).
\end{array}
$$

If a coin is tossed, the probability of it landing heads or tails is influenced by whether it is fair or biased: if the coin is fair (\+biased), then it has an equal chance of landing heads or tails (0.5). If the coin is biased, then it is more likely to land heads with a probability of 0.6, and tails with a probability of 0.4. C_3 states that the coin is fair with a probability of 0.9 or biased with a probability of 0.1. C_4 asserts that a `coin` *is indeed tossed. Since we're only considering 1 coin, each rule has 1 grounding $\theta_1 = \{Coin/coin\}$. Here, T would have $2 \times 2 \times 2 = 8$ possible worlds.*

Given a goal G, its probability $P(G)$ can be defined by marginalizing the joint probability of the goal and the worlds: $P(G) = \sum_{w\in W_T} P(G, w) = \sum_{w\in W_T} P(G|w)P(w) = \sum_{w\in W_T : w\models G} P(w)$. The probability of a goal G given a world w is $P(G|w) = 1$ if $w \models G$ and 0 otherwise. $P(w) = P(\sigma)$, i.e. is the product of the annotations p_{ik} of the head atoms selected in σ. Therefore, the probability of G can be computed by summing the probability of the worlds where the goal is true. In practice, given a goal to solve, it is unfeasible to enumerate all the worlds where G is entailed. Inference algorithms, instead, find *explanations* for a goal: a composite choice κ is an *explanation* for G if G is entailed by every world of w_κ.

Example 2. *(Ex.1 cont.) To determine the overall probabilities of the coin landing on heads or tails, we need to ask the probability of the 2 goals* heads *and* tails. *Each goal is true in 4 worlds out of the 8:*

$$P(\textit{heads}) = (0.5 \times 0.6 \times 0.9) + (0.5 \times 0.6 \times 0.1) + (0.5 \times 0.4 \times 0.9) + (0.5 \times 0.6 \times 0.1) = 0.51$$

$$P(\textit{tails}) = 1 - P(\textit{heads}) = (0.5\times0.4\times0.1)+(0.5\times0.6\times0.9)+(0.5\times0.4\times0.1)+(0.5\times0.4\times0.9) = 0.49$$

3. Probabilistic events, traces and logs

In this Section, we present our semantic framework for addressing uncertainty across events, traces, and logs, building upon the DS. This approach acknowledges that in certain domains, complete observation of a process instance may not be feasible, leading to uncertainty related to events, traces and even logs. We can assign a probability to events [11], obtaining probabilistic traces, or to traces as a whole, obtaining probabilistic logs. Probability always reflects the degree of belief or confidence of the user in the happening of the event or the trace.

With a finite alphabet of symbols Σ, representing activity names or descriptors, we can define:

Definition 1 (Trace and Log). *A* Trace *is a finite, ordered sequence of symbols over* Σ*, denoted as* $t \in \Sigma^*$*, where* Σ^* *represents the infinite set of all possible finite sequences (sentences)* t*. Syntactically, a trace is expressed as* $t = \langle e_1, e_2, \ldots, e_n \rangle$*,* $e_i \in \Sigma$*, where* n *is the* length *of the trace, and* e_i *(for* $i \in 1 \ldots n$*) represents the* i*-th event in the trace. A log* $\mathcal{L}$ *consists of a finite set of such traces.*

Definition 2 (Probabilistic Event [11]). *A* Probabilistic Event *is a couple Prob:EventDescription, where EventDescription is a symbol describing an event (EventDescription* $\in \Sigma$*), while* Prob $\in [0, 1]$ *is the probability that the event happened. A probability value of 1 means the event happened, and we will refer to it as "certain".*

For example, the probabilistic event 0.8:early_mobilization in a trace of a medical log describes the event of a patient's early mobilization after surgery with probability 0.8, reflecting our degree of belief associated with the event's occurrence. In [11], we defined a trace where at least one event is probabilistic as a probabilistic trace.

Now we extend our framework to probabilistic logs, driven by real-world scenarios where traces may not be accurately captured due to factors like software or hardware malfunctions and human error. As a consequence, there is no certainty of the happening of some process instance. However, due to the domain's characteristics, it may be the case that the *whole instance* (trace) happened with a certain probability.

Definition 3 (Probabilistic Log). *A* probabilistic log $\mathcal{L}_p$ *is a log where at least one trace* t_i *is annotated with a probability* p_i*. A probability value of 1 means the trace certainly happened and the value will be omitted.*

Instead of considering the happening of the single events in a trace, as in Def. 2, here we are inquiring about the certainty of the process instance as a whole: it certainly happened or maybe it happened with a degree of confidence.

Example 3. *In hospitals, patients are first admitted to the emergency department following an initial screening known as triage. In exceptional situations, such as during serious emergencies, the triage process might be performed but not recorded in the log. The probabilistic log:*

$$\mathcal{L}_p = \{ \quad t_1, \quad 0.9 : t_2, \quad t_3, \quad 0.7 : t_4 \quad \}$$

describes a scenario in which the process instances t_1 *and* t_3 *were observed and recorded, while* t_2 *was not observed but there is a high probability (0.9) that it happened. Similarly,* t_4 *was not observed but there is a fair probability (0.7) that it happened.*

We propose a straightforward extension of Sato's distribution semantics, as done in [11], to the case of probabilistic logs.

Definition 4 (Selection σ over a probabilistic log $\mathcal{L}_p$). *A* Selection $\sigma(\mathcal{L}_p)$ *is defined as a composite choice containing an atomic choice* (t_i, k) *for each trace* $t_i \in \mathcal{L}_p$*. A selection* $\sigma(\mathcal{L}_p)$ *identifies a world* w_σ *in this way:* $w_\sigma = \{t_i | (t_i, 1) \in \sigma\}$*.*

Example 4. *Given the probabilistic log* $\mathcal{L}_p$ *described in Example 3, four selections are possible, generating four corresponding worlds:*

$$\sigma_1(\mathcal{L}_p) = \{ \quad (t_2, 1), \quad (t_4, 1)\} \qquad\qquad w_{\sigma_1}(\mathcal{L}_p) = \{ \quad t_1, \quad t_2, \quad t_3, \quad t_4 \quad \}$$

$$\sigma_2(\mathcal{L}_p) = \{ \quad (t_2, 1), \quad (t_4, 0)\} \qquad w_{\sigma_2}(\mathcal{L}_p) = \{ \quad t_1, \quad t_2, \quad t_3 \quad \}$$
$$\sigma_3(\mathcal{L}_p) = \{ \quad (t_2, 0), \quad (t_4, 1)\} \qquad w_{\sigma_3}(\mathcal{L}_p) = \{ \quad t_1, \quad t_3, \quad t_4 \quad \}$$
$$\sigma_4(\mathcal{L}_p) = \{ \quad (t_2, 0), \quad (t_4, 0)\} \qquad w_{\sigma_4}(\mathcal{L}_p) = \{ \quad t_1, \quad t_3 \quad \}$$

Note that traces t_1 and t_3 always appear in the generated worlds as they are certain. A possible world $w_{\sigma_i}(\mathcal{L}_p)$ represents a *possible log*, determined by the presence or absence of individual uncertain traces. A selection over such a log determines which traces are considered to be part of a possible realization of the log.

Definition 5 (Probability of a Selection $\sigma(\mathcal{L}_p)$)**.** *The probability of a selection $\sigma(\mathcal{L}_p)$ over a probabilistic log $\mathcal{L}_p$ is defined as:*

$$P(\sigma(\mathcal{L}_p)) = \prod_{(t_i,1)\in\sigma(\mathcal{L}_p)} p_i \prod_{(t_i,0)\in\sigma(\mathcal{L}_p)} (1 - p_i)$$

The probability of a selection corresponds to the probability of a possible log (i.e., a possible world), obtained by multiplying the probabilities associated to each alternative (presence or absence of a trace) as these are considered independent of each other. This gives a probability distribution over the logs, i.e. $\sum_i P(\sigma_i(\mathcal{L}_p)) = P(w_{\sigma_i}(\mathcal{L}_p)) = 1$.

Example 5 (Ex. 4 cont.)**.** *The probabilities of the four selections $\sigma_i(\mathcal{L}_p)$ are:*

$$P(\sigma_1(\mathcal{L}_p)) = P(w_{\sigma 1}(\mathcal{L}_p)) = 0.9 \times 0.7 = 0.63 \quad P(\sigma_3(\mathcal{L}_p)) = P(w_{\sigma 3}(\mathcal{L}_p)) = 0.1 \times 0.7 = 0.07$$
$$P(\sigma_2(\mathcal{L}_p)) = P(w_{\sigma 2}(\mathcal{L}_p)) = 0.9 \times 0.3 = 0.27 \quad P(\sigma_4(\mathcal{L}_p)) = P(w_{\sigma 4}(\mathcal{L}_p)) = 0.1 \times 0.3 = 0.03$$

Note that 0.63+0.27+0.07+0.03=1. The 4 realizations of the log, with very different probabilities in this case, highlight the fact the very high (low) values of confidence in the happening of some traces may generate logs with much higher (lower) confidence than others. This means that a user can rank the probabilistic realizations of the logs from the one with highest confidence (the most probable) to the one with the lowest confidence.

4. Conclusions and Future Work

In this work, we presented a unified framework inspired by the distribution semantics of Probabilistic Logic Programming, which integrates our recently proposed probabilistic semantics for process events to handle uncertainty at various granularity levels: not only events, but also traces and logs. In the future, we plan to extend this framework to include proof procedures for conformance checking for probabilistic logs.

5. Acknowledgments

Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani | Università degli Studi di Ferrara

Research funded by the Italian Ministerial grant PRIN 2022 “Probabilistic Declarative Process Mining (PRODE)”, n. 20224C9HXA - CUP F53D23004240006, funded by European Union – Next Generation EU. Research funded by the Italian Ministry of University and Research through PNRR - M4C2 - Investimento 1.3 (Decreto Direttoriale MUR n. 341 del 15/03/2022), Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 8 "Pervasive AI" - CUP J33C22002830006, funded by the European Union under the NextGeneration EU programme".

References

[1] M. Pegoraro, W. M. van der Aalst, Mining uncertain event data in process mining, 2019 International Conference on Process Mining (ICPM) (2019) 89–96. URL: https://api.semanticscholar.org/CorpusID:199490116.

[2] S. J. J. Leemans, A. F. Syring, W. M. P. van der Aalst, Earth movers' stochastic conformance checking, in: T. Hildebrandt, B. F. van Dongen, M. Röglinger, J. Mendling (Eds.), Business Process Management Forum, Springer International Publishing, Cham, 2019, pp. 127–143.

[3] M. Pegoraro, W. M. van der Aalst, Mining uncertain event data in process mining, in: 2019 International Conference on Process Mining (ICPM), 2019, pp. 89–96. doi:10.1109/ICPM.2019.00023.

[4] M. Pegoraro, M. S. Uysal, W. M. P. van der Aalst, Efficient construction of behavior graphs for uncertain event data, in: W. Abramowicz, G. Klein (Eds.), Business Information Systems, Springer International Publishing, Cham, 2020, pp. 76–88.

[5] M. Pegoraro, B. Bakullari, M. S. Uysal, W. M. P. van der Aalst, Probability estimation of uncertain process trace realizations, in: J. Munoz-Gama, X. Lu (Eds.), Process Mining Workshops, Springer International Publishing, Cham, 2022, pp. 21–33.

[6] G. Bergami, F. M. Maggi, M. Montali, R. Peñaloza, Probabilistic trace alignment, in: 2021 3rd International Conference on Process Mining (ICPM), 2021, pp. 9–16. doi:10.1109/ICPM53251.2021.9576856.

[7] M. Pesic, H. Schonenberg, W. M. van der Aalst, Declare: Full support for loosely-structured processes, in: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 2007, pp. 287–287. doi:10.1109/EDOC.2007.14.

[8] M. Pesic, Constraint-based workflow management systems : shifting control to users, Phd thesis 1 (research tu/e / graduation tu/e), Industrial Engineering and Innovation Sciences, 2008. doi:10.6100/IR638413, proefschrift.

[9] A. Alman, F. M. Maggi, M. Montali, R. Peñaloza, Probabilistic declarative process mining, Inf. Syst. 109 (2022) 102033. doi:10.1016/J.IS.2022.102033.

[10] M. Vespa, E. Bellodi, F. Chesani, D. Loreti, P. Mello, E. Lamma, A. Ciampolini, Probabilistic compliance in declarative process mining, in: Accepted for publication at the 3rd International Workshop on Process Management in the AI Era (PMAI 2024), 2024, pp. 1–12.

[11] M. Vespa, E. Bellodi, F. Chesani, D. Loreti, P. Mello, E. Lamma, A. Ciampolini, M. Gavanelli, R. Zese, Probabilistic traces in declarative process mining, in: Accepted for publication at the 23rd International Conference of the Italian Association for Artificial Intelligence (AIXIA 2024), 2024, pp. 1–14.

[12] T. Sato, A statistical learning method for logic programs with distribution semantics, in: L. Sterling (Ed.), Logic Programming, Proceedings of the Twelfth International Conference on Logic Programming, Tokyo, Japan, June 13-16, 1995, MIT Press, 1995, pp. 715–729.

[13] F. Riguzzi, E. Bellodi, E. Lamma, R. Zese, Reasoning with probabilistic ontologies, in: Q. Yang, M. J. Wooldridge (Eds.), Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, AAAI Press, 2015, pp. 4310–4316. URL: http://ijcai.org/Abstract/15/613.

[14] F. Riguzzi, E. Bellodi, E. Lamma, R. Zese, Epistemic and statistical probabilistic ontologies, in: F. Bobillo, R. Carvalho, P. C. G. da Costa, N. Fanizzi, K. B. Laskey, K. J. Laskey, T. Lukasiewicz, T. Martin, M. Nickles, M. Pool (Eds.), Proceedings of the 8th International Workshop on Uncertain Reasoning for the Semantic Web (URSW2012), Boston, USA, 11 November 2012, number 900 in CEUR Workshop Proceedings, Sun SITE Central Europe, Aachen, Germany, 2012, pp. 3–14.

[15] J. Vennekens, S. Verbaeten, M. Bruynooghe, Logic programs with annotated disjunctions, in: B. Demoen, V. Lifschitz (Eds.), 20th International Conference on Logic Programming (ICLP 2004), volume 3131 of *LNCS*, Springer, 2004, pp. 431–445. doi:10.1007/978-3-540-27775-0_30.

[16] E. Bellodi, The distribution semantics in probabilistic logic programming and probabilistic description logics: a survey, Intelligenza Artificiale 17 (2023) 143 – 156. doi:10.3233/IA-221072.

[17] D. Poole, The Independent Choice Logic for modelling multiple agents under uncertainty, Artificial Intelligence 94 (1997) 7–56.