Fuzzy Logic for Markov Networks

Carles Noguera[1] and Igor Sedlár[2]

[1] Department of Information Engineering and Mathematics, University of Siena
Siena, Italy
carles.noguera@unisi.it
[2] Institute of Computer Science, Czech Academy of Sciences
Prague, Czech Republic
sedlar@cs.cas.cz

1 Introduction

Markov networks, also known as Markov random fields, are probabilistic graphical models that represent the dependencies between random variables using an undirected graph [6, 7]. Markov networks provide a compact representation of high-dimensional probability distributions, coupled with efficient inference algorithms and a clear visual representation. Unlike directed graphical models, such as Bayesian networks, Markov networks are useful for modelling phenomena where directionality cannot be naturally imposed on the relationships between random variables. Markov networks have found widespread application in fields as diverse as image analysis, natural language processing, bioinformatics, and social network analysis.

In this work, we adopt a fuzzy logic perspective on Markov networks. First, we argue that interpreting the potential functions of a Markov network in terms of truth degrees clarifies their nature and makes it easier to elicit the potential functions from experts. This interpretation of potential functions gives rise to the concept of a graded Markov network. Secondly, we introduce a two-layered fuzzy logic framework for reasoning about graded Markov networks. We propose that this logic could serve as a specification language for Markov networks, as well as a tool for reasoning about the specifications provided by experts or learned from data.

2 Markov networks

Recall that a **random variable** is a function $\mathscr{A}$ from a set $\Omega(\mathscr{A})$ (the sample space of the variable) to a set $\mathrm{Val}(\mathscr{A})$ (values of the variable). A random variable $\mathscr{A}$ is **discrete** if $\mathrm{Val}(\mathscr{A})$ is finite. We will work exclusively with discrete random variables. If $\boldsymbol{\mathscr{A}} = (\mathscr{A}_1, \ldots, \mathscr{A}_n)$ is a tuple of random variables, then an $\boldsymbol{\mathscr{A}}$-*state* is any tuple $\boldsymbol{a} \in \prod_{i=1}^{n} \mathrm{Val}(\mathscr{A}_i) =_{def} \mathrm{Val}(\boldsymbol{\mathscr{A}})$.

Definition 1. *A **Markov network** is a pair $N = (\boldsymbol{\mathscr{A}}, \boldsymbol{\Phi})$ where $\boldsymbol{\mathscr{A}} = (\mathscr{A}_1, \ldots, \mathscr{A}_n)$ is a tuple of discrete random variables and $\boldsymbol{\Phi} = (\phi_1, \ldots, \phi_m)$ is a tuple of **factors** over $\boldsymbol{\mathscr{A}}$, that is, pairs $\phi_i = (\boldsymbol{\mathscr{A}}_{\{i\}}, \mathfrak{p}_i)$ where $\boldsymbol{\mathscr{A}}_{\{i\}}$ is a sub-tuple of $\boldsymbol{\mathscr{A}}$ (the scope of ϕ_i) and $\mathfrak{p}_i \colon \mathrm{Val}(\boldsymbol{\mathscr{A}}_{\{i\}}) \longrightarrow \mathbb{R}^+$ (the potential function). We use the notation $\boldsymbol{a}_{\{i\}}$ for elements of $\mathrm{Val}(\boldsymbol{\mathscr{A}}_{\{i\}})$.*

A network $N = (\boldsymbol{\mathscr{A}}, \boldsymbol{\Phi})$ can be represented as a factor graph over variable vertices $\boldsymbol{\mathscr{A}}$, factor vertices $\boldsymbol{\Phi}$, with an edge connecting a factor vertex ϕ_i with the variable vertices in its scope.

Example 1. A situation where three friends go to a pub and choose between two drinks, beer (B) and kofola (K), can be represented by a Markov network as follows. Let $\mathscr{A} = (A, B, C)$ be random variables representing choices of the three friends and let $\boldsymbol{\Phi} = (\phi_i \text{ for } i = 1, \ldots, n)$ be factors representing various constraints (e.g. A is very likely to order beer, B always orders the same drink as A, C tends to order the same drinks as B and C would prefer kofola). A nework

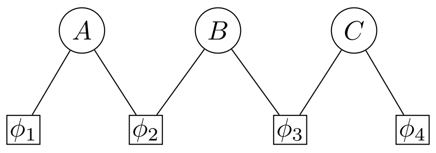

Figure 1: An example of a factor graph.

representing this scenario can be specified by the factor graph in Figure 1 and the following potential functions, for example:

a	$\phi_1(a)$
B	2
K	0.8

a	b	$\phi_2(a,b)$
B	B	2
B	K	0
K	B	0
K	K	2

b	c	$\phi_3(b,c)$
B	B	1.2
B	K	0.4
K	B	0.4
K	K	1.2

c	$\phi_4(c)$
B	0.8
K	1.4

There is a close relationship between Markov networks and *valued constraint satisfaction problems* [8]. Intuitively, factors correspond to *valued constraints* on states. However, factors play different roles in VCSP and MN, respectively. While VCSP aims at finding the optimal states, the primary role of a Markov network is to provide a compact representation of a probability distribution over states.

Definition 2. *For $N = (\boldsymbol{\mathcal{A}}, \mathbf{\Phi} = (\phi_1, \dots, \phi_m))$, we define:*

Weight *of* $\boldsymbol{a} \in \mathrm{Val}(\boldsymbol{\mathcal{A}})$ ***Normalising constant*** *of* N ***Probability*** *of* $\boldsymbol{a} \in \mathrm{Val}(\boldsymbol{\mathcal{A}})$

$$\mathsf{W}_N(\boldsymbol{a}) = \prod_{i=1}^{m} \phi_i(\boldsymbol{a}_{\{i\}}) \qquad Z_N = \sum_{\boldsymbol{b} \in \mathrm{Val}(\boldsymbol{\mathcal{A}})} \mathsf{W}(\boldsymbol{b}) \qquad \mathsf{P}_N(\boldsymbol{a}) = \mathsf{W}_N(\boldsymbol{a}) \cdot Z_N^{-1}$$

Probability of events $E \subseteq \mathrm{Val}(\boldsymbol{\mathcal{A}})$ is defined as $\mathsf{P}_N(E) = \sum_{\boldsymbol{a} \in E} \mathsf{P}_N(\boldsymbol{a})$ and conditional probability is defined as usual.

3 Graded Markov networks

Interpreting the numerical values of factors in Markov networks is a recognised challenge; see, for example, [7, pp. 107–108], [6, p. 106] and [2, p. 269]. Inspired by the connection between Markov networks and valued constraint satisfaction problems, we observe that this problem is mitigated within a particular class of Markov networks.

Definition 3. *A Markov network $N = (\boldsymbol{\mathcal{A}}, \mathbf{\Phi})$ is* ***graded*** *if $\phi_i(\boldsymbol{a}_{\{i\}}) \in [0,1]$ for all $\phi_i \in \mathbf{\Phi}$ and all $\boldsymbol{a} \in \mathrm{Val}(\boldsymbol{\mathcal{A}})$. A Markov network is* ***normal graded*** *if $\max_{\boldsymbol{a} \in \mathrm{Val}(\boldsymbol{\mathcal{A}})} \phi_i(\boldsymbol{a}_{\{i\}}) = 1$ for all $\phi_i \in \mathbf{\Phi}$ and all $\boldsymbol{a} \in \mathrm{Val}(\boldsymbol{\mathcal{A}})$.*

In graded MNs, $\phi_i(\boldsymbol{a}_{\{i\}})$ represents the *truth degree* of the imprecise statement "$\boldsymbol{a}_{\{i\}}$ is compatible with constraint i". Such statements can readily be provided by domain experts since people are used to scales (think of film ratings, satisfaction questionnaires, etc.). The real unit interval $[0,1]$ can be thought of as an infinitely fine scale.

It can be easily shown that for each Markov network N there is a normal graded network M, based on the same tuple of random variables $\boldsymbol{\mathcal{A}}$, which is equivalent to N in the sense that $\mathsf{P}_N(\boldsymbol{a}) = \mathsf{P}_M(\boldsymbol{a})$ for all $\boldsymbol{a} \in \mathrm{Val}(\boldsymbol{\mathcal{A}})$. (First, normalise globally by the highest value appearing in all potential tables. Then, normalise each table locally by the maximal value appearing in it.)

In addition, using $[0,1]$ instead of $\mathbb{R}^+$ (or even $\mathbb{R}^{+\infty}$) enables a better grasp on the relative sizes of degrees. Indeed, while all closed intervals of real numbers are homeomorphic, in $[0,1]$ we have a fixed standard arithmetics that allows e.g. to see the number 0.5 as the middle point of the interval in a canonical way, whereas the exact location of the middle point of $\mathbb{R}^{+\infty}$ would actually depend on what particular homeomorphism one uses to map it back to $[0,1]$.

4 A fuzzy logic for Markov networks

Let σ be the first-order signature with individual constants A_k, a_k for $k \in \omega$ and with individual variables $Var = \{x_k \mid k \in \omega\}$, where the only relational symbol is identity.

Definition 4. *We define (σ-)**parameters** and (σ-)**formulas** as follows:*

$$p := f_i(\alpha) \mid P(\alpha) \qquad\qquad \varphi := \mathbb{W}(\alpha) \mid p \mid \bar{c} \mid \varphi \to \varphi \mid \varphi \cdot \varphi \mid \varphi + \varphi$$

*where α is a quantifier-free first-order σ-formula (an **event formula**) and $c \in [0,1] \cap \mathbb{Q}$.*

Example 2. The structure of the Markov network from Example 1 can be expressed by the formula

$$\mathbb{W}(A = x \wedge B = y \wedge C = z) \leftrightarrow f_1(A = x) \cdot f_2(A = x \wedge B = y) \cdot f_3(B = x \wedge C = z) \cdot f_4(C = z)$$

and the potential functions of the network are defined by formulas such as

$$f_3(B = \textsc{b} \wedge C = \textsc{k}) \leftrightarrow \overline{0.25}$$

Constraints elicited from experts, such as 'C is more likely to order kofola than beer' can be expressed by $f_4(C = \textsc{b}) \to f_4(C = \textsc{k})$.

The syntax of our logic builds on the syntax of two-layered probabilistic fuzzy logics [3, 4], to which we add parameters. The role of parameters $f_i(\alpha)$ is to directly specify potential functions of Markov networks. Parameters $P(\alpha)$ are used in probabilistic queries, see Example 3 below. The 'upper layer' of the syntax is an extension of Product Łukasiewicz logic [5] with rational constants and the addition operator $+$.

Ground terms, formulas and parameters are defined as those without occurrences of $x \in Var$. A ground σ-interpretation is a set Δ of ground atomic σ-formulas where the relation $\{\langle t_1, t_2\rangle \mid t_1 = t_2 \in \Delta\}$ is an equivalence relation on the set of ground σ-terms. The set of ground σ-interpretations (resp. parameters) is denoted by $GrInt(\sigma)$ (resp. $GrPar(\sigma)$).

Definition 5. *A (σ-)**model** is $\mathcal{M} = (\mathcal{S}, \mathcal{I}, \mathcal{V})$ where $\mathcal{S} = (\Omega, \Sigma, \mu)$ is a measure space and*

$$\mathcal{I} : \Omega \to GrInt(\sigma) \qquad\qquad \mathcal{V} : GrPar(\sigma) \to [0,1]$$

*An **evaluation** is $\mathfrak{e} : Var \to GrTm(\sigma)$.*

The semantics of event formulas is specified as follows: $\|\alpha\|^{\mathcal{M},\mathfrak{e}} = \{w \in \Omega \mid (\mathcal{I}(w), \mathfrak{e}) \models \alpha\}$, where $\models$ is defined as usual in first-order logic. We assume that $\|\alpha\|^{\mathcal{M},\mathfrak{e}} \in \Sigma$ for all $\alpha \in EvFm$.

Definition 6. *The **interpretation** of σ-formulas in a model $\mathcal{M}$ given an evaluation $\mathfrak{e}$ is defined as follows:*

$$[\![\mathbb{W}(\alpha)]\!]^{\mathcal{M},e} = \mu\left(\|\alpha\|^{\mathcal{M},e}\right) \qquad [\![p[x_1,\dots,x_n]]\!]^{\mathcal{M},e} = \mathcal{V}(p[\mathfrak{e}(x_1),\dots,\mathfrak{e}(x_n)])$$

$$[\![\overline{c}]\!]^{\mathcal{M},e} = c \qquad [\![\varphi \to \psi]\!]^{\mathcal{M},e} = 1 - [\![\varphi]\!]^{\mathcal{M},e} + [\![\psi]\!]^{\mathcal{M},e} \qquad [\![\varphi \star \psi]\!]^{\mathcal{M},e} = [\![\varphi]\!]^{\mathcal{M},e} \star [\![\psi]\!]^{\mathcal{M},e}$$

for $\star \in \{\cdot, +\}$.

Note that $\|\varphi\|^{\mathcal{M},e} \in \mathbb{R}^+$, not necessarily $[0,1]$. The situation here is similar to the Łukasiewicz unbound logic [1].

Definition 7. ***Entailment*** *is a relation between a pair* (Δ, Γ) *on the one hand and* φ *on the other hand, where* Δ *is a set of event formulas and* $\Gamma \cup \{\varphi\}$ *is a set of formulas, defined as follows:* $\Delta, \Gamma \models \varphi$ *iff, for all* $\mathcal{M}$*,*

if $\quad \|\alpha\|^{\mathcal{M},e} = \Omega$ *for all* $\alpha \in \Delta$ *and all* $\mathfrak{e}$*, and* $[\![\psi]\!]^{\mathcal{M},e} \geq 1$ *for all* $\psi \in \Gamma$ *and all* $\mathfrak{e}$*,*

then $\quad [\![\varphi]\!]^{\mathcal{M},e} \geq 1$ *for all* $\mathfrak{e}$*.*

Example 3. Probabilistic queries in Markov networks correspond to specific entailment problems $\Delta, \Gamma \models \varphi$ in our logic. Let Δ be a relevant set of event assumptions that contains, e.g. formulas such as $A = a_i \leftrightarrow \neg \bigvee_{j \in I_A \setminus \{i\}} (A = a_j)$ for I_A representing the set of values of the random variable A. Let Γ contain a formula representing the structure of the intended network and formulas defining its potential functions. Moreover, let Γ contain the *query assumption*

$$\mathbb{W}(A = \textsc{b} \wedge B = \textsc{b} \wedge C = \textsc{k}) \leftrightarrow P(A = \textsc{b} \wedge B = \textsc{b} \wedge C = \textsc{k}) \cdot \mathbb{W}(\top)$$

where $\mathbb{W}(\top)$ represents the normalising constant.[1] Finally, let φ be the *query* such as $\overline{c} \to P(A = \textsc{b} \wedge B = \textsc{b} \wedge C = \textsc{k})$ for some desired threshold $c \in [0,1] \cap \mathbb{Q}$.

Finding a complete axiomatisation and determining decidability and complexity of our logic are natural open problems.

References

[1] P. Cintula, B. Grimau, C. Noguera, and N. J. J. Smith. These degrees go to eleven: fuzzy logics and gradable predicates. *Synthese*, 200(6), Oct. 2022. doi:10.1007/s11229-022-03909-2.

[2] F. G. Cozman. Languages for probabilistic modeling over structured and relational domains. In P. Marquis, O. Papini, and H. Prade, editors, *A Guided Tour of Artificial Intelligence Research, Volume 2*, pages 247–283. Springer, 2020. doi:10.1007/978-3-030-06167-8_9.

[3] P. Hájek, L. Godo, and F. Esteva. Fuzzy logic and probability. In P. Besnard and S. Hanks, editors, *UAI'95: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 237–244, San Francisco, 1995. Morgan Kaufmann.

[4] P. Hájek, L. Godo, and F. Esteva. Reasoning about probability using fuzzy logic. *Neural Network World*, 10(5):811–824, 2000.

[5] R. Horčík and P. Cintula. Product Łukasiewicz logic. *Archive for Mathematical Logic*, 43(4):477–503, May 2004. doi:10.1007/s00153-004-0214-6.

[6] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, Cambridge, Mass., 2009. URL: https://dl.acm.org/doi/book/10.5555/1795555.

[7] J. Pearl. *Probabilistic Reasoning in Intelligent Systems. Networks of Plausible Inference.* Elsevier, 1988. doi:10.1016/c2009-0-27609-4.

[8] T. Schiex, H. Fargier, G. Verfaillie, et al. Valued constraint satisfaction problems: Hard and easy problems. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, pages 631–639, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[1]Note that if α and β are inconsistent in a model, then $\mathbb{W}(\alpha \vee \beta) \leftrightarrow \mathbb{W}(\alpha) + \mathbb{W}(\beta)$ is valid in the model.