

Quantitatively Evaluating Formula-Variable Relevance by Forgetting

Xin Liang¹, Zuoquan Lin¹, and Jan Van den Bussche²

¹ School of Mathematical Sciences, Peking University, Beijing 100871, China
`{xliang,lz}@pku.edu.cn`

² Hasselt University and Transnational University of Limburg, 3590 Diepenbeek, Belgium
`jan.vandenbussche@uhasselt.be`

Abstract. Forgetting is a feasible tool for weakening knowledge bases by focusing on the most important issues, and ignoring irrelevant, outdated, or even inconsistent information, in order to improve the efficiency of inference, as well as resolve conflicts in the knowledge base. Also, forgetting has connections with relevance between a variable and a formula. However, in the existing literature, the definition of relevance is “binary” – there are only the concepts of “relevant” and “irrelevant”, and no means to evaluate the “degree” of relevance between variables and formulas. This paper presents a method to define the formula-variable relevance in a quantitative way, using the tool of variable forgetting, by evaluating the change of model set of a certain formula after forgetting a certain variable in it. We also discuss properties, examples and one possible application of the definition.

Keywords: knowledge representation, forgetting, relevance, inconsistency.

1 Introduction

Forgetting [1] is an accessible tool for weakening formulas in knowledge bases. Variable forgetting is the most basic form of forgetting. The concept of forgetting has connections with the concept of independence, which is also discussed by Lin and Reiter [1], as well as Lang, Liberatore and Marquis [2]. They have defined the concept of independence in different approaches.

However, current approaches of defining independency only give the judgement of “dependent” or “independent”. In other words, these approaches only give a “black or white” answer, and ignore the intermediate conditions (i.e., there is no “greyscale”). For example, consider the two simple formulas: $p \wedge q$ and p . Intuitively, and by the definition presented by Lang, Liberatore and Marquis [2], both formulas are dependent of p . However, by intuition, the latter one is more “dependent” of p than the former one. How to characterize this “degree” of “dependency” (or “relevance”), in order to capture the difference between, say, the two formulas mentioned above, on a certain variable? This paper presents an

approach, which is based on the work of Lin and Reiter [1] and the work of Lang, Liberatore and Marquis [2], extending those ideas to the quantitative analysis of the sets of models of formulas. The key point is to measure the change of the set of models after forgetting. After the definition, we will discuss properties, examples, and one possible application of it.

2 Preliminaries

This paper is based on propositional logic. Let PS be a set of propositional variables (e.g., $p, q, r, p_1, p_2, \dots, p_n, \dots$, etc.), and $PROP_{PS}$ denotes the set of all propositional formulas defined on the set PS . Propositional variables occurring on the formulas in $PROP_{PS}$ are all in the set PS , and there exist formulas in $PROP_{PS}$ in which only some (not all) variable(s) in PS occur(s). The language of $PROP_{PS}$ consists of the commonly used logical connectives, such as $\neg, \rightarrow, \vee, \wedge, \leftrightarrow$. An interpretation is a truth-value assignment, assigning every variable in PS a truth value from $\{T, F\}$. Here we use a subset of PS to denote an interpretation: If a variable is in the subset, then it is assigned T , otherwise it is assigned F . Given an interpretation ω and a formula φ in $PROP_{PS}$, we can tell whether ω satisfies φ (denoted by $\omega \models \varphi$, stating that φ is true under ω), by the semantics of logical connectives. A model of a formula φ is an interpretation in which φ is true. $\text{Mod}(\varphi)$ denotes the set of models of φ with respect to the variables in the set PS . We say formula ϕ entails formula ψ , denoted by $\phi \models \psi$, if $\text{Mod}(\phi) \subseteq \text{Mod}(\psi)$. We say ϕ and ψ are logically equivalent, denoted by $\phi \equiv \psi$, if $\text{Mod}(\phi) = \text{Mod}(\psi)$.

Variable forgetting on propositional logic was proposed by F. Lin and R. Reiter [1]. Lang, Liberatore and Marquis proposed literal forgetting on propositional logic [2]. Here we only consider the semantic definition of variable forgetting: Let φ be a propositional formula, p be a variable in it, $\text{ForgetVar}(\varphi, p)$ denotes the result of *forgetting* p in φ , whose models can be obtained as follows:

$$\text{Mod}(\text{ForgetVar}(\varphi, p)) = \text{Mod}(\varphi) \cup \{\text{Switch}(\omega, p) \mid \omega \models \varphi\} , \quad (1)$$

where $\text{Switch}(\omega, p)$ means to change the truth-value assignment of p in the interpretation ω to the opposite one.

3 Formula-Variable Relevance

In this section, we define the method to evaluate formula-variable relevance in a quantitative way. First, let us consider the models of ψ_i and $\text{ForgetVar}(\psi_i, p)$ under the variable set $PS = \{p, q\}$, where i is a variable for the enumeration of propositional formulas, ranging from 1 to 8. The models of ψ_i and $\text{ForgetVar}(\psi_i, p)$ are listed in Table 1. (Just ignore the last column of Table 1, which will be useful later in this section.)

An intuitive idea to define the degree of formula-variable relevance $R_{\text{FV}}(\psi, p)$ is using the “amount” of the *increment* of models (from ψ to $\text{ForgetVar}(\psi, p)$).

Table 1. The models of ψ_i and $\text{ForgetVar}(\psi_i, p)$, as well as the value of formula-variable relevance, which is defined later in this section

i	ψ_i	$\text{Mod}(\psi_i)$	$\text{Mod}(\text{ForgetVar}(\psi_i, p))$	$R_{\text{FV}}(\psi_i, p)$
1	$p \wedge q$	$\{\{p, q\}\}$	$\{\{p, q\}, \{q\}\}$	$1/2$
2	$p \vee q$	$\{\{p\}, \{q\}, \{p, q\}\}$	$\{\emptyset, \{p\}, \{q\}, \{p, q\}\}$	$1/2$
3	p	$\{\{p\}\}$	$\{\emptyset, \{p\}\}$	1
4	$\neg p$	$\{\emptyset\}$	$\{\emptyset, \{p\}\}$	1
5	$p \rightarrow q$	$\{\emptyset, \{p\}, \{p, q\}\}$	$\{\emptyset, \{p\}, \{q\}, \{p, q\}\}$	$1/2$
6	$p \rightarrow p$	$\{\emptyset, \{p\}, \{p, q\}\}$	$\{\emptyset, \{p\}, \{q\}, \{p, q\}\}$	$1/2$
7	$p \leftrightarrow q$	$\{\emptyset, \{p, q\}\}$	$\{\emptyset, \{p\}, \{q\}, \{p, q\}\}$	1
8	$(p \wedge q) \vee ((\neg p) \wedge q)$	$\{\{q\}, \{p, q\}\}$	$\{\{q\}, \{p, q\}\}$	0

The underlying intuition is: if the model set of $\text{ForgetVar}(\psi, p)$ remains unchanged or little changed comparing to the model set of ψ , it means that p is already or almost “forgotten” in ψ , and ψ contains models in which p is *true* and almost the same amount of models in which p is *false* (and these models are almost the same of the former models in the valuation of variables except p). In this sense, p is more irrelevant with ψ under this circumstance, because the operation of “forgetting p in ψ ” does not change the model set of ψ too much, which means that the formula ψ *itself* does not have much relevance with the variable p . An extreme condition is that the model set of ψ remain unchanged after forgetting p , i.e. $\psi \equiv \text{ForgetVar}(\psi, p)$. This extreme case is equivalent to the “binary” definition of independence which is proposed in the existing literature [2]. If the model set of $\text{ForgetVar}(\psi, p)$ changes a lot after the forgetting operation, we can say that the formula ψ is more relevant with the variable p . This is because the operation of forgetting *does* change a lot in the model set of ψ , which could be seen as the result of the fact that ψ is more relevant with p . The more the models change (increase), the more relevant ψ and p are.

By the above intuition, we may give a formal definition of formula-variable relevance.

Definition 1. Let ψ be a formula in PROP_{PS} , p a variable occurring in ψ . The degree of formula-variable relevance $R_{\text{FV}}(\psi, p)$ is defined as:

$$R_{\text{FV}}(\psi, p) = \frac{|\text{Mod}(\text{ForgetVar}(\psi, p)) \setminus \text{Mod}(\psi)|}{2^{|PS|-1}}. \quad (2)$$

Proposition 1. For any ψ , and p occurring in ψ ,

$$0 \leq R_{\text{FV}}(\psi, p) \leq 1, \quad (3)$$

and for any $i \in \{0, 1, 2, \dots, 2^{|PS|-1}\}$, we can find some ψ and p , such that $R_{\text{FV}}(\psi, p) = i/2^{|PS|-1}$.

Note that if we have a set of variables PS' such that $PS' \supseteq PS$, then for a formula ψ defined on PS , and a variable $p \in PS$, we may get the same formula-variable relevance both under PS and PS' .

Example 1. Now we have some formulas denoted by ψ_i , and let us consider $R_{FV}(\psi_i, p)$ for each i . The readers may find different ψ_i 's and the corresponding results back in Table 1.

Next we show some properties of the relevance defined above.

Proposition 2. *Let p be a variable in PS , and ϕ, ψ two formulas defined on PS .*

1. *If ϕ and ψ are logically equivalent, then $R_{FV}(\phi, p) = R_{FV}(\psi, p)$.*
2. *If ϕ is a tautology or a contradiction, then $R_{FV}(\phi, p) = 0$.*
3. *If $p \models \phi$ and $\neg p \models \phi$, then $R_{FV}(\phi, p) = 0$.*
4. *If $\phi \models p$ or $\phi \models \neg p$, and ϕ is satisfiable, then $R_{FV}(\phi, p) > 0$.*

Now let us discuss the issue on computational complexity. We propose a problem (named COUNTREL), which has important connections with the problem of formula-variable relevance. COUNTREL is stated as follows:

- *Input.* A propositional formula ψ , and a variable p occurring in ψ . Let PS be all the variables occurring in ψ .
- *Output.* $|\text{Mod}(\text{ForgetVar}(\psi, p)) \setminus \text{Mod}(\psi)|$.

Theorem 1. *The problem COUNTREL is #P-complete.*

4 Defining Preference among Recoveries with Formula-Variable Relevance

Lang and Marquis [3,4] presented a method to resolve inconsistencies of a given knowledge base via variable forgetting. As we know, it is a common thing that inconsistencies may occur in a knowledge base due to different sources of knowledge or different views of the world for each person. In classical logics, if inconsistency lies in a belief set, then *ex falso quodlibet* occurs, which means that, all the well-formed formulas are theorems of the formal system. Obviously this is meaningless and the formal logic system becomes useless. To avoid this, researchers developed several means for inconsistency tolerance [5].

As one possible approach to resolve inconsistency, Lang and Marquis [3,4] proposed a concept called *recovery*, which is based on the concept of *forgetting vector*, and it plays a key role in the whole process. A forgetting vector is a vector whose items are sets of variables which are to be forgotten in each formula in the belief base, satisfying a certain *forgetting context*. For the notations, a forgetting vector is $\mathbf{V} = (V_1, \dots, V_n)$, where V_i is a set of variables to be forgotten in ϕ_i , and ϕ_i is a formula in belief set B (all the formulas in B could be listed as $\phi_1, \phi_2 \dots \phi_n$). The set of all forgetting vectors for a knowledge base B given a forgetting context \mathcal{C} is denoted as $\mathcal{F}_{\mathcal{C}}(B)$. A *recovery* for a knowledge base under a certain forgetting context is a forgetting vector after “forgetting” which the knowledge base is consistent. For the details, the readers may refer to the original paper by Lang and Marquis [3,4].

Then a problem occurs: For a certain (inconsistent) knowledge base, there may be various recoveries. The authors state that we can define a kind of *preference relation* among recoveries. The preference relation is a kind of binary relations. There are many ways to define the preference relation, reflecting different views of which vector of variable sets to “forget”.

Hereby we can apply the formula-variable relevance defined in the previous part of this paper to this circumstance. The idea is to “forget” the vector of variable sets which is least relevant to the formulas in the belief set.

To define the preference relation, we firstly define a ranking function according to formula-variable relevance. The idea is that for each element (variable set) of the forgetting vector, we evaluate the relevance between the corresponding source of knowledge (ϕ_i), and each variable in the variable set, then add them together.

Definition 2.

$$\text{Rank}_B(\mathbf{V}) = \sum_{i=1}^n \sum_{v \in V_i} R_{FV}(\phi_i, v) , \quad (4)$$

where $B = (\phi_1, \dots, \phi_n)$, $\mathbf{V} = (V_1, \dots, V_n)$.

Now we have the ranking function, then the definition of preference relation is straightforward.

Definition 3. *The preference relation among forgetting vectors, based on formula-variable relevance, denoted by \sqsubseteq_{Rel} , is defined as follows:*

$$\mathbf{V} \sqsubseteq_{\text{Rel}} \mathbf{V}' \text{ if and only if } \text{Rank}_B(\mathbf{V}) \leq \text{Rank}_B(\mathbf{V}') . \quad (5)$$

Lang and Marquis [3,4] mentioned a property named *monotonicity*. The preference relation defined above satisfies this property.

Proposition 3. \sqsubseteq_{Rel} satisfies monotonicity, i.e. for all $\mathbf{V}, \mathbf{V}' \in \mathcal{F}_C(B)$, if $\mathbf{V} \subseteq_p \mathbf{V}'$, then $\mathbf{V} \sqsubseteq_{\text{Rel}} \mathbf{V}'$.

Note: $\mathbf{V} \subseteq_p \mathbf{V}'$ means that for all i , $V_i \subseteq V'_i$.

Example 2. Let us come to an example which was stated in the papers of Lang and Marquis [3,4]. It is an example of building tennis court and/or swimming pool. The knowledge base (which is inconsistent) and the forgetting context are given as follows [3,4], where “ \oplus ” is the “xor” connective, and formulas with this connective can be easily transformed to an equivalent formula only with the five basic connectives ($\neg, \rightarrow, \vee, \wedge, \leftrightarrow$):

- $\phi_1 = (s \rightarrow (s_r \oplus s_b)) \wedge (\neg s \rightarrow \neg s_r \wedge \neg s_b) \wedge (c_2 \leftrightarrow (s \wedge t)) \wedge (c_1 \leftrightarrow (s \oplus t)) \wedge (c_0 \leftrightarrow (\neg s \wedge \neg t))$;
- $\phi_2 = (c_0 \vee c_1) \wedge (s \rightarrow s_r)$;
- $\phi_3 = (s \vee t) \wedge (s \rightarrow s_b)$;
- $\phi_4 = s$;
- $\phi_5 = s \wedge t$;

- The forgetting context \mathcal{C} is the conjunction of the following formulas (where $forget(x, i)$ means that atom x may be forgotten in ϕ_i , and $Var(\phi)$ is all the variables occurring in ϕ):
 1. $\bigwedge_{x \in Var(\phi_1)} \neg forget(x, 1)$;
 2. $\bigwedge_{i=2}^5 (forget(s, i) \rightarrow forget(s_r, i)) \wedge (forget(s_r, i) \leftrightarrow forget(s_b, i)) \wedge (forget(c_0, i) \leftrightarrow forget(c_1, i)) \wedge (forget(c_1, i) \leftrightarrow forget(c_2, i))$.

In this example, in order to restore consistency, the authors gave 9 possible forgetting vectors as recoveries, which are listed in Table 2. Actually, we can choose any of them in order to resolve inconsistency. However, the preference defined by formula-variable relevance can give a criteria for choosing recovery. For the knowledge base B and the recoveries \mathbf{V}^1 to \mathbf{V}^9 , we may compute $\text{Rank}_B(\mathbf{V}^i)$ for each i , which is listed in Table 2.

Table 2. Different recoveries and the corresponding ranking values

i	\mathbf{V}^i	$\text{Rank}_B(\mathbf{V}^i)$
1	$\langle \emptyset, \{s, s_b, s_r\}, \{s, s_b, s_r\}, \{s, s_b, s_r\}, \{s, s_b, s_r\} \rangle$	13/4
2	$\langle \emptyset, \{t, s_b, s_r\}, \{t, s_b, s_r\}, \{t, s_b, s_r\}, \{t, s_b, s_r\} \rangle$	15/8
3	$\langle \emptyset, \{c_0, c_1, c_2, s_b, s_r\}, \{c_0, c_1, c_2, s_b, s_r\}, \{c_0, c_1, c_2, s_b, s_r\}, \{c_0, c_1, c_2, s_b, s_r\} \rangle$	13/8
4	$\langle \emptyset, \{s_b, s_r\}, \{s_b, s_r\}, \emptyset, \{t\} \rangle$	11/8
5	$\langle \emptyset, \{c_0, c_1, c_2\}, \{s_b, s_r\}, \emptyset, \emptyset \rangle$	5/4
6	$\langle \emptyset, \{s_b, s_r\}, \emptyset, \emptyset, \{t\} \rangle$	7/8
7	$\langle \emptyset, \emptyset, \emptyset, \{s, s_b, s_r\}, \{s, s_b, s_r\} \rangle$	3/2
8	$\langle \emptyset, \{c_0, c_1, c_2, s_b, s_r\}, \emptyset, \emptyset, \emptyset \rangle$	9/8
9	$\langle \emptyset, \emptyset, \{s_b, s_r\}, \emptyset, \{t\} \rangle$	1

According to our criteria, we should choose the recovery \mathbf{V}^6 . Actually, we implemented a computer program computing formula-variable relevance.

5 Related Work and Conclusions

In the existing literature, relevance between a variable and a formula is concerned, but in a “binary” way, i.e., there are only two cases – “relevant” and “irrelevant”, not concerning the measurement of the “degree” of relevance. For example, Lang, Liberatore and Marquis [2] defined various kinds of Literal-/Variable-independence, and the authors also showed the connections between independence and forgetting. In the paper where forgetting was firstly proposed in the field of artificial intelligence [1], Lin and Reiter defined a kind of irrelevance as follows: Let φ be a propositional formula, q a query, and p a ground atom, then we say that p in φ is *irrelevant* for answering q iff φ and $\text{ForgetVar}(\varphi, p)$ are equivalent w.r.t. q . This definition is based on relevance with respect to a certain query q .

Our work is inspired by the independence defined in the existing literature above, and adds quantitative measurement of formula-variable relevance.

In conclusion, our paper, based on the traditional idea of forgetting and relevance, proposed an approach to evaluate how relevant a variable and a formula are. Also, we have applied this approach to define preference relations in a forgetting-based approach of inconsistency resolving in the existing literature.

For the future work, note that Lang, Liberatore and Marquis [2] discussed literal forgetting, and also, Xu and Lin [6] discussed formula forgetting. We may employ the methods in these papers to define “formula-literal relevance” and the relevance between formulas. Another noticeable issue is that forgetting has been defined in other logics [7,8,9]. Quantitatively defining relevance by forgetting in these logics is a topic worth studying.

Acknowledgements. The authors wish to thank Chaosheng Fan, Kedian Mu, Geng Wang, Dai Xu, Xiaowang Zhang, and other people with whom we have had discussions on this topic. We would also like to thank the anonymous reviewers for the helpful comments. This work is supported by the program of the National Natural Science Foundation of China (NSFC) under grant number 60973003, the Research Fund for the Doctoral Program of Higher Education of China, and the Graduate School of Peking University.

References

1. Lin, F., Reiter, R.: Forget it! Working Notes of AAAI Fall Symposium on Relevance, pp. 154–159 (1994)
2. Lang, J., Liberatore, P., Marquis, P.: Propositional independence. *Journal of Artificial Intelligence Research* 18, 391–443 (2003)
3. Lang, J., Marquis, P.: Resolving inconsistencies by variable forgetting. In: *International Conference on Principles of Knowledge Representation and Reasoning*, pp. 239–250. Morgan Kaufmann Publishers (2002)
4. Lang, J., Marquis, P.: Reasoning under inconsistency: A forgetting-based approach. *Artificial Intelligence* 174(12–13), 799–823 (2010)
5. Bertossi, L., Hunter, A., Schaub, T. (eds.): *Inconsistency Tolerance*. LNCS, vol. 3300. Springer, Heidelberg (2005)
6. Xu, D., Lin, Z.: A prime implicates-based formulae forgetting. In: *2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 3, pp. 128–132. IEEE (2011)
7. Wang, Z., Wang, K., Topor, R., Pan, J.Z.: Forgetting concepts in DL-lite. In: *Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008*. LNCS, vol. 5021, pp. 245–257. Springer, Heidelberg (2008)
8. Eiter, T., Wang, K.: Semantic forgetting in answer set programming. *Artificial Intelligence* 172(14), 1644–1672 (2008)
9. Zhang, Y., Zhou, Y.: Knowledge forgetting: Properties and applications. *Artificial Intelligence* 173(16), 1525–1537 (2009)