

Handling Revision Inconsistencies: Towards Better Explanations

Fabian Schmidt¹(✉), Jörg Gebhardt¹, and Rudolf Kruse²

¹ ISC Gebhardt, Celle, Germany
schmidt@isc-gebhardt.de

² Otto-von-Guericke University, Magdeburg, Germany

Abstract. When dealing with complex knowledge, inconsistencies become a big problem. Due to the complexity of modern knowledge systems, usually a manual elimination of inconsistencies by a domain expert is preferable, since automated systems are most of the time not able to properly model and use the domain knowledge of an expert. In order to eliminate an inconsistency correctly, with respect to the specific domain, an expert needs a proper understanding of that inconsistency respectively the components that constitute it. Especially in our focus area of inconsistencies that occur during the revision of probability distributions, creating useful explanations is in most cases still a manual and hence expensive effort. In this work we discuss how to automatically create groupings of partitions created by revision assignments and how explanations can benefit from those grouped partitions.

1 Introduction

One important aspect of managing knowledge is the need to react to changes in beliefs quickly and frequently. Methods have been developed to adapt knowledge to new beliefs. In order to adapt knowledge properly, the **principle of minimal change** [10] should be respected. This principle states that no changes are to be made to the knowledge base that are not necessary to incorporate given new beliefs. This means the knowledge base after the incorporation of the new beliefs should be the closest knowledge base to the original one, in an information theoretic sense. The **revision operation** has been introduced as belief change operation that applies new beliefs respecting the principle of minimal change [7]. Further properties a revision operation should satisfy have been formulated as postulates in [1, 3]. How to approach revision algorithmically has been outlined in [5] and computational considerations have been made in [12]. This work focusses on the revision of probability distributions. In this field the revision operation has been successfully implemented for Markov networks [6, 8]. Markov networks are a member of a class of so called graphical models [2, 11, 13, 17], which are techniques to decompose high-dimensional probability spaces into a number of smaller low-dimensional probability spaces.

The growing complexity and interconnectedness of knowledge bases and increasing number of new beliefs lead almost inevitably to inconsistencies. Inconsistencies in knowledge bases however, pose a threat to the usability of any

knowledge system and should consequently be addressed. Handling inconsistencies is a multi-facet problem. In this work we focus on the handling of inconsistencies during the revision of probability distributions. Different important aspects of that problem have been introduced in [15]. Furthermore, two types of inconsistencies and a revision control algorithm have been described in [9].

One important aspect of handling inconsistencies properly is to try to eliminate them. Two types of elimination can be differentiated: the first type is the automated elimination during the revision; the second type is the manual elimination by domain experts that normally happens after the revision operation. In order to manually eliminate inconsistencies, domain experts need to gain a proper understanding of the underlying contradictions at the core of inconsistencies. Therefore, the creation of meaningful explanations for inconsistencies is important. Different components of explanations have been described in [14]. Furthermore, one automated method for creating explanations has been proposed in [15]. In that approach, a minimal explaining set of revision assignments is determined and used as explanation. In this work, we will discuss the grouping of partitions created by revision assignments and how they can be used to create easier to understand explanations that also work with more complex problems.

In Sect. 2 of this paper, we will formally introduce the revision operation, the revision factor and revision inconsistencies. Section 3 then discusses the grouping of partitions. Additionally, we will introduce our example application and present some test results. In Sect. 4, we conclude our work and give some ideas for further research.

2 Fundamentals

In this section we will describe revision assignments, the revision operation, the revision factor and what inconsistencies are in that context.

2.1 The Revision Operation

This paper focusses on the revision of probability distributions and we therefore define it in this context.

As mentioned before, the goal of (probabilistic) revision is to compute a posterior probability distribution which satisfies given new distribution conditions, only accepting a minimal change of the quantitative interaction structures of the underlying prior distribution.

More formally, in our setting, a revision operation (see [6, 9]) operates on a joint probability distribution $P(V)$ on a set $V = \{X_1, \dots, X_n\}$ of variables with finite domains $\Omega(X_i)$, $i = 1, \dots, n$. The purpose of the operation is to adapt $P(V)$ to new sets of beliefs. The beliefs are formulated in a so-called **revision structure** $\Sigma = (\sigma_s)_{s=1}^S$. This structure consists of **revision assignments** σ_s , each of which is referred to a (conditional) assignment scheme $(R_s|K_s)$ with a **context scheme** K_s , $K_s \subseteq V$, and a **revision scheme** R_s , where $\emptyset \neq R_s \subseteq V$ and $K_s \cap R_s = \emptyset$. The pair $(P(V), \Sigma)$ is called **revision problem**.

For example, in the revision assignment $(NAV=nav1 | Country=France) := 0.2$, which sets the probability for the navigation system *nav1* in the country *France* to 0.2, the context scheme K_s is $\{Country\}$ and the revision scheme R_s is $\{NAV\}$.

Revision assignments partition the probability space with respect to their context and revision schemes. Expanding on the previous example, suppose there are five different values for the variable *NAV* and three revision assignments, namely:

$$\begin{aligned} \sigma_1 &\stackrel{Def}{=} [(NAV = nav1 | Country = France) := 0.2] \\ \sigma_2 &\stackrel{Def}{=} [(NAV = nav2 | Country = France) := 0.25] \\ \sigma_3 &\stackrel{Def}{=} [(NAV = nav3 | Country = France) := 0.3] \end{aligned}$$

The revision assignments create the partitions $\{nav1\}$, $\{nav2\}$, and $\{nav3\}$ for the domain $\Omega(NAV)$ in the context of *Country=France*. Since the probabilities of those three assignments do not sum up to 1, they also together create a fourth partition containing $\{nav4, nav5\}$. We will use this type of partitions later for our grouping approach.

The result of the revision, and solution to the revision problem, is a probability distribution $P_\Sigma(V)$ which

- satisfies the revision assignments (the postulated new probabilities)
- preserves the probabilistic interaction structure as far as possible.

By preserving the interaction structure we mean that, except from the modifications induced by the revision assignments in Σ , all probabilistic dependencies of $P(V)$ are preserved. This requirement ensures that modifications are made according to the principle of minimal change.

It can be proven (see, i.e. [6]) that in case of existence, the solution of the revision problem $(P(V), \Sigma)$ is uniquely defined. This solution can be determined using iterative proportional fitting [17]. Starting with the initial probability distribution, this process adapts the initial probability distribution iteratively, one revision assignment at the time, and converges to a limit distribution that solves the revision problem, given there are no inconsistencies.

2.2 Revision Factors

In each iteration of the iterative proportional fitting process, partitions as explained earlier are multiplied with a so called **revision factor** in order to incorporate the current revision assignment. Consider a single revision assignment $\sigma^* \stackrel{Def}{=} P^*(\rho_s | \kappa_s)$ of a new probability for $\rho_s | \kappa_s$, with ρ_s and κ_s being single partitions.

With respect to the concept of minimal change, revising $P(V)$ by σ^* leads to the probability distribution $P_{rev}(V)$ which satisfies the condition $P_{rev}(\rho_s | \kappa_s) = P^*(\rho_s | \kappa_s)$ and $P_{rev}(V - K_s - R_s, \kappa_s, \rho_s) = P(V - K_s - R_s, \kappa_s, \rho_s)$.

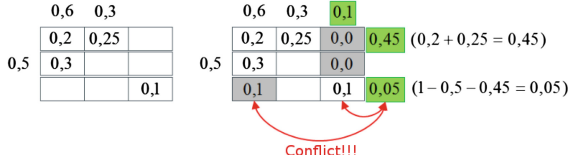


Fig. 1. Inner inconsistency

Using the general rule for factorisation, we obtain

$$\begin{aligned}
 P_{rev}(V - K_s - R_s, \kappa_s, \rho_s) &= P_{rev}(V - K_s - R_s | \kappa_s, \rho_s) P_{rev}(\rho_s | \kappa_s) P_{rev}(\kappa_s) \\
 &= P(V - K_s - R_s | \kappa_s, \rho_s) P^*(\rho_s | \kappa_s) P(\kappa_s) \\
 &= \frac{P^*(\rho_s | \kappa_s)}{P(\rho_s | \kappa_s)} P(V - K_s - R_s, \kappa_s, \rho_s),
 \end{aligned}$$

where $\frac{P^*(\rho_s | \kappa_s)}{P(\rho_s | \kappa_s)}$ describes the revision factor.

In case of a solvable revision problem $(P(V), \Sigma)$ those factors converge towards one as the revision operation converges.

2.3 Inconsistencies in the Context of the Revision Operation

In case of inconsistencies, the revision will oscillate between multiple limit distributions. In the worst case there are as many limit distributions as there are revision assignments where each limit distribution satisfies the incorporation of one revision assignment. Furthermore, the revision factors will also be oscillating between different values in order to transfer the probabilities from one limit distribution to another one.

Inconsistencies have been analysed in [9] and two types of inconsistencies of revision problems have been distinguished:

Inner consistency of a revision structure Σ is given, if and only if a probability distribution exists that satisfies the revision assignments of Σ ; otherwise we refer to **inner inconsistencies** of Σ .

In Fig. 1, a simple example is shown where the given revision assignments already lead to an inconsistency without the consideration of the underlying interaction structure. The filled entries in the left table represent the revision assignments. In the right table consequences for the rest of the table are shown and one conflict is highlighted.

Given that Σ has the property of inner consistency, it is still possible that due to the zero probabilities of $P(V)$ the revision problem $(P(V), \Sigma)$ is not solvable, since a modification of the interaction structure of $P(V)$ would be necessary in order to satisfy the given revision assignments. Therefore, a second type of inconsistency is defined as follows:

Given that Σ has the property of inner consistency, the revision problem $(P(V), \Sigma)$ shows the property of **outer inconsistency**, if and only if there is no solution to this revision problem.

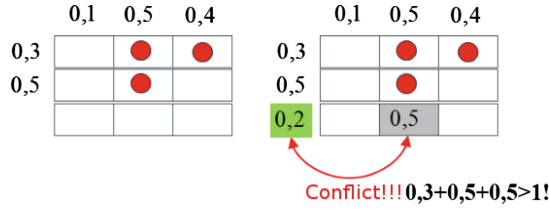


Fig. 2. Outer inconsistency

Figure 2 illustrates an outer inconsistency. In the left table again the numbers represent revision assignments. This time there are additional circles representing zero values that cannot be changed during the revision operation. As before, the right table shows consequences for the remaining table entries as well as an inconsistency.

For the purpose of creating explanations, we now know that there are two potential sources of contradictions. Namely, the revision structure Σ and the interaction structure of $P(V)$, or more precisely its zero-values.

3 Explaining Revision Inconsistencies

In the case of inconsistency the revised distribution P_{rev} does not exist. After cancelling the revision we obtain P_{appr} , a distribution that approximates P_{rev} . Due to the inconsistency there exists at least $\sigma \in \Sigma$ where $P_{appr}(\sigma) \neq P^*(\sigma)$. In order to support the manual elimination of inconsistencies, our explanations aim to highlight the core contradiction that caused $P_{appr}(\sigma^*) \neq P^*(\sigma^*)$ for one chosen revision assignment σ^* . In previous works we proposed a minimal explaining set of revision assignments that together constitute the contradiction as described in [15] as explanation. Such set is effective as long as the number of elements in it is moderately small, or the explained inconsistency doesn't spread over too many dimensions. With an increasing number of participating variables and items in the set, explanations get more and more incomprehensible. Therefore, even after the introduction of an automated method for finding minimal explaining sets, there are still requests for manual analysis of inconsistencies, because the automatically created explanation is not sufficient.

We identified two aspects that most manually created explanations cover, and are currently not incorporated into our automated system: A relevant extract of the interaction structure as well as a meaningful grouping of partitions, created by the revision assignments. In this work, we focus on the automated acquisition of meaningful groupings of partitions.

3.1 Grouping

Grouping together partitions, created by the revision structure Σ , has two positive effects on explanations. First, it reduces the number of elements presented

to a domain expert. And second, it magnifies the core feature of the interaction structure that contributes to an inconsistency. To visualise this effect, Fig. 3 shows the interaction structure between two variables. On the left side the original interaction structure is shown. The right side presents the condensed version after the manual grouping of partitions. Relations between the groups of elements are more clearly visible on the right side.

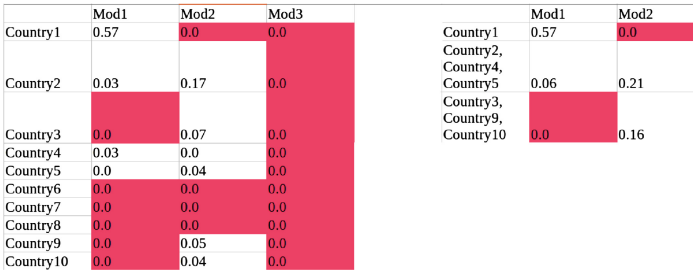


Fig. 3. Left: original interaction structure, right: interaction structure after manual grouping

The biggest challenge when grouping elements automatically is to find a suitable similarity measure in order to decide which elements should be grouped. For the grouping of partitions created by revision assignments, we found that the revision factor can be used. The revision factor is applied to partitions created by the revision structure in every iteration of the revision operation in order to adapt probabilities according to the specified revision structure. During the revision operation very often partitions that would be grouped together by a data analysis expert, are adapted with the same or very similar revision factors. The revision factor converges for every partition once the revision operation converges. In the case that the revision converges towards exactly one limit distribution (and hence does not show inconsistencies), the revision factors will converge towards one. Otherwise, the revision factors will differ and provide a suitable similarity measure. The fact that they are converging means that we can use them as a base for further analysis, and in our case as a measure of similarity.

One of the most common class of methods to group objects by similarity are clustering algorithms. In order to choose a suitable algorithm from that class we have to analyse our grouping problem. In general, we are interested in a result that has the least possible number of clusters, but still explains the chosen inconsistency. The exact number of clusters is unknown and most likely changes between problems. Another aspect is the number of elements we need to cluster. In our case we are dealing with somewhere between a couple of hundred to a couple of thousand revision assignments. However, that number can usually already drastically be reduced before starting the clustering. We decided to use hierarchical clustering [4, 16], since the hierarchy provides us with an easy method

	mod1	mod2	mod3	mod4	mod5	revision factor
nav1	0,1443	0,0134	0,0000	0,0000	0,0000	1,0857
nav2	0,0000	0,0000	0,0800	0,2200	0,0000	0,8000
nav3	0,0037	0,0014	0,0000	0,0000	0,0000	1,0857
nav4	0,4519	0,0352	0,0000	0,0000	0,0000	1,0857
nav5	0,0000	0,0000	0,0000	0,0000	0,0379	1,0857
nav6	0,0000	0,0000	0,0000	0,0000	0,0121	1,0857
revision factor	0,9211	0,9211	1,2500	1,2500	0,9211	
Grouped						
		mod1+mod2+mod5		mod3+mod4		
$\sim(\text{nav2})$		0,7000		0,0000		
nav2		0,0000		0,3000		

Fig. 4. Above: original interaction structure, below: interaction structure after automated grouping by revision factors

to progressively test groupings with an increasing number of clusters until we found the best solution.

The resulting algorithm groups partitions that are derived from the revision structure Σ and clusters them using the revision factors that are observed when the revision operation converges. The result is a set of groupings of partitions that can then be used to create a more concise view on the relevant part of the interactions structure of the resulting probability distribution P_{rev} .

The grouping created in Fig. 3 could technically be achieved by deriving it from the interaction structure. However, this is a very simple example we choose to illustrate the effect of grouping. In real applications, the interaction structure is only one component that influences the similarity between different partitions. The clustering of partitions using the revision factors is able to also find similarity between structurally different partitions. Furthermore, partitions that look structurally similar in certain projections might indeed have differences caused by higher level dependencies that remain hidden during a structural analysis of a given projection.

Figure 4 shows an example where using revision factors finds groupings that would not have been visible through structural analysis alone. When analysing the structure of the table above, one would find three partitions for the variable NAV. Namely $\{\text{nav1}, \text{nav3}, \text{nav4}\}$, $\{\text{nav2}\}$, and $\{\text{nav5}, \text{nav6}\}$. Similarly for the variable MODEL: $\{\text{mod1}, \text{mod2}\}$, $\{\text{mod3}, \text{mod4}\}$, and $\{\text{mod5}\}$. However, the analysis of the revision factors revealed that there is an even more concise grouping shown in the table below.

For those reasons structural analysis in general only gives a partial view on the picture. Another aspect is that the interaction structure is relatively static as it does not change during the revision, except from the changes induced by the revision structure. On the other hand, revision factors change dynamically in every iteration of the revision and hence reflect the dynamics of the revision operation to a certain degree. In this way, even dependencies that are not visible

in a given projection are considered. This is especially important since manual analysis, most of the time, is restricted to schemes that the data analysis expert considers important. Furthermore, in case of inconsistencies the way the revision factors change already gives certain insight into the nature of inconsistencies.

All the mentioned properties of the grouping of partitions using the revision factor as a similarity measure lead to groupings that can be used to provide a concise explanation for a given inconsistency.

3.2 Application

The presented approach is tested at the Volkswagen Group in their system for estimating part demands and managing capacities for short- and medium-term forecasts, called EPL (EigenschaftenPLANung: item planning). The system combines several heterogeneous input sources such as rules describing buildable vehicle specifications, production history reflecting customer preferences, and market forecasts leading to stipulations of modified item rates, and capacity restrictions that are modelled as revision assignments. Those sources are fused into Markov Networks and the revision operation is then used to estimate the part demands. More details of the modelling of EPL can be found in [6, 7]. EPL is currently the biggest industrial application for Markov networks. Using EPL the demands for more than a hundred different model groups from multiple car manufacturers of the Volkswagen Group are estimated every week. In case of the VW Golf—being Volkswagens most popular car class—there are about 200 item families with typically 4 to 8 (but up to 150) values each, that together fully describe a car to be produced, and many of these families can directly be chosen by the customer.

3.3 Experimental Results

In our productive system we are currently using an explanation based on a minimal explaining set of revision assignments as described in [15]. Compared with that approach we achieved two improvements by using the presented clustering method. First, we reduced the number of elements needed to explain an inconsistency (see 1). Second, we achieved some significant performance improvements since we are not searching for a minimal explaining set of revision assignments.

Each example comes from a different model group and planning inconsistency. In all the provided examples the clustering algorithm finds an explanation with less elements faster than the algorithm we were using previously. The last example in Table 1 is an extreme case. Even with an optimised version of the original algorithm, the problem is practically not explainable as 680 revision assignments in the explaining set are hard to understand. Furthermore, with 3 h and 50 min of processing time, the problem is probably faster analysed manually. The clustering algorithm still finds 169 clusters which are similarly hard to understand. However, reaching this result in just 7.5 min instead of 3 h and 50 min, means we can resort to a manual analysis significantly earlier. The three examples were chosen to show results for problems with different complexity.

Table 1. Example reduction of the computation time and number of elements in the explanation when using clustering to group partitions

Example	1	2	3
time using the minimal explaining set algorithm	0.807 s	4.027 s	3h49 min
# revision assignments in the minimal set	10	8	680
time using clustering method	0.229 s	1.996 s	7.5 min
# clusters	3	5	169

Although the clustering algorithm finds smaller explanations faster, the type of explanation is slightly different compared to a minimal explaining set of revision assignments and in most cases relies on the dependency structure to understand what the clusters actually mean. The minimal explaining alone is enough to understand simple problems. However, for more complex inconsistencies the clustering algorithm becomes the better choice.

4 Conclusion

The maintenance of knowledge is one of the most important topics in our current times. Aside from the storage and retrieval of knowledge, adapting it to ever changing business environments is one of the biggest challenges. Methods like the revision operation have been introduced in order to suit the need for proper adaptation of knowledge according to new beliefs. However, with more complex knowledge and longer lists of changed beliefs, inconsistencies are practically unavoidable and need to be handled appropriately. One aspect of handling inconsistencies is their elimination which requires their understanding. In order to help domain experts to understand inconsistencies, explaining them is an essential part of the elimination process. In this work we introduced the idea of grouping partitions created by revision assignments as a method to create more concise explanations while at the same time we save valuable processing time. Our experiments showed a significant improvement in both categories but also revealed some areas for future research.

In this paper, as well as in our previous works, it became quite apparent that providing explanations without the corresponding excerpt of the dependency structure reduces the understandability of an explanation considerably. However, identifying and presenting the relevant parts of the structure, is a non-trivial task even for data analysis experts. In order to improve the expressiveness of automatically created explanations, we are interested in finding and presenting those excerpts of the structure in an automated manner as well.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. *J. Symbolic Log.* **50**(02), 510–530 (1985)

2. Borgelt, C., Steinbrecher, M., Kruse, R.: *Graphical Models - Representations for Learning, Reasoning and Data Mining*. Wiley, Chichester (2009)
3. Darwiche, A.: On the logic of iterated belief revision. *Artif. Intell.* **89**(1–2), 1–29 (1997)
4. Defays, D.: An efficient algorithm for a complete link method. *Comput. J.* **20**(4), 364–366 (1977)
5. Gabbay, D.: Controlled revision - an algorithmic approach for belief revision. *J. Logic Comput.* **13**(1), 3–22 (2003)
6. Gebhardt, J., Borgelt, C., Kruse, R., Detmer, H.: Knowledge revision in markov networks. *Mathware Soft Comput.* **11**(2–3), 93–107 (2004)
7. Gebhardt, J., Detmer, H., Madsen, A.L.: Predicting parts demand in the automotive industry-an application of probabilistic graphical models. In: *Proceedings of the International Joint Conference on Uncertainty in Artificial Intelligence* (2003)
8. Gebhardt, J., Klose, A., Detmer, H., Ruegheimer, F., Kruse, R.: Graphical models for industrial planning on complex domains. In: del la Riccia, G., e.a. (ed.) *Decision Theory and Multi-Agent Planning*, pp. 131–143. No. 482 in *CISM Courses and Lectures*. Springer (2006)
9. Gebhardt, J., Klose, A., Wendler, J.: Markov network revision: on the handling of inconsistencies. In: Moewes, C., Nürnberger, A. (eds.) *Computational Intelligence in Intelligent Data Analysis. SCI*, vol. 445, pp. 153–165. Springer, Heidelberg (2013)
10. Gärdenfors, P.: *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge (1988)
11. Kruse, R., et al.: *Computational Intelligence. A Methodological Introduction*. Springer, London (2013)
12. Nebel, B.: Base revision operations and schemes: representation, semantics and complexity. In: *Proceedings of the 11th ECAI 1994, Amsterdam, The Netherlands*, pp. 341–345. Wiley (1994)
13. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo (1991)
14. Schmidt, F., Gebhardt, J., Kruse, R.: Handling revision inconsistencies: creating useful explanations. In: *HICSS-48, Proceedings*, 5–8 January 2015, Koloa, Kauai, HI, USA, pp. 1–8. IEEE Computer Society (2015)
15. Schmidt, F., Wendler, J., Gebhardt, J., Kruse, R.: Handling inconsistencies in the revision of probability distributions. In: Pan, J.-S., Polycarpou, M.M., Woźniak, M., de Carvalho, A.C.P.L.F., Quintián, H., Corchado, E. (eds.) *HAIS 2013. LNCS*, vol. 8073, pp. 598–607. Springer, Heidelberg (2013)
16. Sibson, R.: Slink: An optimally efficient algorithm for the single-link cluster method. *Comput. J.* **16**(1), 30–34 (1973)
17. Whittaker, J.: *Graphical Models in Applied Multivariate Statistics*. Wiley, New York (1990)