Markov Network Revision: On the Handling of Inconsistencies

Jörg Gebhardt, Aljoscha Klose, and Jan Wendler

Abstract. Graphical models are of high relevance for complex industrial applications. The Markov network approach is one of their most prominent representatives and an important tool to structure uncertain knowledge about high-dimensional domains in order to make reasoning in such domains feasible. Compared to *conditioning* the represented probability distribution on given evidence, the important belief change operation called *revision* has been almost entirely disregarded in the past, although it is of utmost relevance for real world applications. In this paper we focus on the problem of *inconsistencies* during revision in Markov networks. We formally introduce the revision operation and propose methods to specify, identify, and resolve inconsistencies. The revision and its inconsistency management has proven to be successful in a complex application for item planning and capacity management in the automotive industry at Volkswagen Group.

1 Introduction

Today's scientific and economic problems are often characterised by a large number of variables. With a sufficiently high number of variables, the complexity of these problems grows quickly, so that analyses and reasoning processes become increasingly difficult. For this reason, lossless or approximating decomposition techniques are often necessary in order to efficiently cope with high dimensionalities. Decomposition is achieved by making use of (conditional) independencies between variables. Graphical models [15, 12, 1] have established themselves as one of the most popular tools to structure uncertain knowledge in this way, so that inference becomes feasible [14, 9]. Their most prominent representatives are Bayesian networks [14], which are based on directed graphs and conditional probability distributions, as well as Markov networks [13], which refer to undirected graphs and marginal probability distributions or factor potentials.

Jörg Gebhardt · Aljoscha Klose · Jan Wendler
ISC Gebhardt, Celle, Germany
e-mail: info@isc-gebhardt.de

C. Moewes et al. (Eds.): Computational Intelligence in Intelligent Data Analysis, SCI 445, pp. 153–165.
springerlink.com

When dealing with graphical models several non-trivial operations have to be considered. For the first step of knowledge representation, one needs learning [4, 11] and data fusion algorithms to get an appropriate structure and the initial distributions of a network. Further knowledge processing on a given network is realized, for example, by information retrieval, belief change, and inference operations, respectively [3, 4, 5].

The most discussed knowledge processing operation in the field of probabilistic graphical models is **focusing**, which can be achieved by performing any kind of evidence-driven conditioning on a set of input variables and propagating the new information. Instantiation of variables as it is usually implemented in diagnostic tools, can be considered as a special case of this operation, with all the probability mass assigned to one value per given variable.

It is surprising that other essential operations well-known from uncertainty management in knowledge-based systems seem to be overlooked in the scientific community of graphical modeling: They concern the two almost complementary operations of revision and updating, respectively. Compared to focusing, these operations are not restricted to pure information retrieval and simulation aspects, but reflect the task of belief change.

Revision refers to an alteration of the represented probability distribution within the frame of an existing model structure, i.e. although the probability of a state (element of the common domain) may be changed in the revision process, it is required that forbidden states (having a zero probability) do not change. Revision is performed by locally introducing new distributions into the Markov network. Like with focusing, local modifications of distributions are propagated. But in contrast to the operations used in information retrieval, changes made during revision are permanent, as the modified distributions replace those already stored in the model. The alterations to the model are the least ones required to integrate the new probability assignments. Therefore the maximum of the probabilistic interaction structure already represented in the model is preserved, which coincides with the so-called principle of minimal change [5].

If multiple local distributions in the network have to be modified, the desired revision is achieved by propagating the new assignments one after another (iterative proportional fitting [15]). Since any local change may affect other areas in the model, processing one of the assignments may invalidate part of the models' adaptations to previous assignments in the sequence. However, by iterating the process the model will often converge to a state of stable compromise, consistent with all assignments.

Nevertheless, if the assignments are in conflict with each other or affect zero probabilities of the initial distribution, the whole revision process will remain unstable and equilibrium cannot be achieved. In the first case there can be no consistent, accurate and complete model for contradictory evidence. In the latter case a solution can be achieved by applying an updating operator.

Updating is complementary to a revision operation in the sense that this operation locally introduces a new probabilistic interaction structure to a model, which means that it changes probabilities from zero to positive values and therefore defines new probabilistic dependencies between the involved variables. For this reason, it does, of course, not follow the principle of minimal change.

In the following contribution we focus our interest on the topic of handling inconsistencies in revision problems. The underlying research was triggered by an application at the automobile manufacturer Volkswagen Group, where ISC Gebhardt established Markov networks for the development of a world-wide software system for item planning and capacity management [2, 7, 6].

The paper is structured as follows: Section 2 introduces the complex item planning problem. In Section 3 we establish definitions to specify and define the revision problem. We discuss revision and identify inconsistency problems which may arise. In Section 4 we focus on inconsistencies, thereby differentiating between inner and outer consistency. Furthermore we introduce the degree of inconsistency of a revision problem. Finally, in Section 5 we present practical solutions to handle inconsistencies in a complex domain.

2 Real-World Application

2.1 Item Planning at Volkswagen Group

In contrast to many competing car manufacturers, Volkswagen Group favours a marketing policy that provides a maximum degree of freedom in choosing individual specifications of vehicles. That is, considering personal preferences, a customer may select from a large variety of options, each of which is taken from a so-called item family that characterises a certain line of equipment. Body variants, engines, gearshifts, door layouts, seat coverings, radios and navigation systems reflect only a small subset of the whole range of item families. In case of the VW Golf – Volkswagen's most popular car class – there are about 200 families with typically 4 to 8 values each, and a total range of cardinalities from 2 up to 150.

Of course, not all of the possible instantiations of item variables lead to valid vehicle configurations, since technical rules, restrictions in manufacturing and sales requirements induce a common rule system that limits the item combinations. Nevertheless, dealing with more than 10,000 technical rules in the Golf class and even more rules delivered by the sales programs for the special needs of different countries, there remains a huge number of correct vehicle specifications. In fact, compared to a total of 910,000 Golf cars in 2011, one can find only a small number of vehicles within the whole production line that have identical specifications.

The major aim of the productive system EPL (EigenschaftsPLanung, German for item planning) at Volkswagen Group was the development and implementation of a software solution that supports item planning, parts demand calculation, and capacity management with the aim of short-term as well as medium-term forecasts up to 24 months of future vehicle production.

In order to achieve high quality of planning results, all relevant information sources have to be considered, namely rules for the correct combination of items into complete vehicle specifications, samples of produced vehicles as a reflection of customers' preferences, market forecasts that lead to revision assignments of modified item rates for planning intervals, capacity restrictions, and production programs that fix the number of planned vehicles.

With respect to the logistics view, the most essential result to assess of the item planning process are the rates of all those item combinations that are known to be relevant for the demand calculation of parts, always related to a certain vehicle class in a certain planning interval. The importance of these item combinations arises from the fact that a vehicle can be interpreted as a large set of installation points, each of which is characterised by a set of alternative parts for the corresponding location. Which of the alternative parts has to be chosen at an installation point depends on its installation condition that can be specified by an item combination. Of course, at each installation point, all occurring installation conditions have to be disjoint, and their disjunction has to form a tautology. That is, given any correct vehicle specification, for each installation point we obtain a unique decision which of its alternative parts has to be used.

In the context of the Golf class, we find a total of about 70,000 different item combinations required as installation conditions for the whole set of installation points. The data structure that lists all installation points, their installation conditions, and the quantities of the referenced parts, is called a variants-related bill of material. The task of predicting the total demand of a certain part with respect to a future planning interval is to sum up the demands over all of its installation points. The demand at any installation point results from multiplying the rate of the item combination that represents its installation condition with the quantity and the total number of vehicles intended to be produced in the respective planning interval.

We conclude that calculating parts demand is reduced to a simple operation, whenever the rates of all involved item combinations can be computed.

2.2 Markov Network Model and Revision Operator

The first step in the project EPL was to search for an appropriate planning model that supports a decomposed representation of the qualitative and quantitative dependency structure between item families. We had to take into account that we deal with a finite set of discrete item variables, as well as that we get conditional independences induced by the given rule systems and customers' preferences.

Since logical rule systems can be transformed into a relational setting, and rates for item combinations may be identified as (frequentistic or subjective) occurrence probabilities within a large sample of historical or predictably valid vehicle specifications, Markov networks turned out to be the most promising environment to satisfy the given modelling purposes.

Once a basic prior Markov network for a certain planning interval has been generated, it becomes subject to a variety of planning operations which involve marketing and sales stipulations (e.g., installation rate of comfort navigation system increases from 20 % to 35 %) and capacity restrictions from logistics (e.g., maximum availability of seat coverings in leather is 5,000). These quantitative input data are strongly related to the planning interval itself and therefore not learnable, neither from historical data nor from the non-probabilistic rule system. They typically consist of predicted installation rates or absolute demands for single items, sets of items, or (sets of) item combinations, and are frequently related to refined planning contexts (e.g., VW Golf with all-wheel drive for the US market).

In mathematical terms, this sort of additional information leads to competing partial or total changes of selected (conditional) low-dimensional probability distributions in the Markov network. Such changes can be interpreted as the basis for a revision operation, where a prior state of knowledge (represented by the initial Markov network) given new information (which is the new set of probability distributions) is revised to a posterior state of knowledge. The new information is thereby incorporated in the sense of the *principle of minimal change* [5].

In terms of the probabilistic framework, the task is to calculate a posterior Markov network that satisfies the new distribution conditions, only accepting a minimal change of the quantitative interaction structures of the underlying prior distribution.

3 Revision in Markov Networks

Before starting the discussion about inconsistencies in Markov networks we need to specify the revision problem and define its solution. Furthermore we reflect how the proposed solution can be calculated and discuss which problems will arise during the revision operation due to the complexity and human factor in real world applications.

3.1 Definitions

Suppose that we are given a **Markov network** $M = (H, \Psi)$ which represents a joint probability distribution $P(V)$ on a set $V = \{X_1, ..., X_n\}$ of variables with finite domains $\Omega(X_i)$, $i = 1, ..., n$. We assume that $H = (V, \{C_1, ..., C_m\})$ denotes a **hypertree** of which the C_i are the (maximal) cliques and $\Psi = (P(C_j))_{j=1}^{m}$ a **family of probability distributions** defined on the (maximal) cliques of H. In this setting, H and its associated **undirected dependency graph** $G(H)$ reflect the conditional independencies between the involved variables, and Ψ shows the resulting **factorization property** $P(V) = \prod_{j=1}^{m} P(C_j)/P(S_j)$, where S_j symbolize the **separators** in some representation of H as a **tree of cliques**.

In addition, let $\Sigma = (\sigma_s)_{s=1}^{S}$ be a so-called **revision structure** that 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 $\varnothing \neq R_s \subseteq V$ and $K_s \cap R_s = \varnothing$. We assume that σ_s is specified by a set of assignment components $P^*(\rho_s^{(k,l)}|\kappa_s^{(k)})$, where $\kappa_s{}^{(k)}$ is called its **context component** and $\rho_s{}^{(k,l)}$ its **revision component**, respectively. The context components are expected to specify a partitioning of $\Omega(K_s)$, and for each $k \in \{1, ..., k^*(s)\}$, the set

$\{\rho_s^{(k,l)} | l = 1, ..., l^*(s,k)\}$ forms a partitioning of $\Omega(R_s)$. Hence, each revision assignment specifies a modified probability distribution referred to the scheme $(R_s|K_s)$, separable into independent modifications of the distributions $P(R_s|\kappa_s^{(k)})$, given by the assignment components $P^*(\rho_s^{(k,l)}|\kappa_s^{(k)})$, $l = 1, ..., l^*(s,k)$. In case of the empty scheme $K_s = \varnothing$, we deal with an assignment of the (non-conditioned) probabilities $P^*(\rho_s^{(l)})$.

Finally, we suppose that for all $s = 1, ..., S$ there are cliques $C(s) \in \{C_1, ..., C_m\}$ such that $K_s \cup R_s \subseteq C(s)$. This guarantees that we do not have cross-over dependencies between cliques, which may not be expressible in the structure of the given Markov network.

Definition 1 (Solution of revision problems). Let $M = (H, \Psi)$ be a Markov network with associated joint probability distribution $P(V)$. Furthermore, let $\Sigma = (\sigma_s)_{s=1}^S$ be a revision structure.

A probability distribution $P_\Sigma(V)$ is called **solution of the revision problem** $(P(V), \Sigma)$, if and only if the following conditions hold:

(R1) **Revision assignments are satisfied:**

$$(\forall s \in \{1, ..., S\})(\forall k \in \{1, ..., k^*(s)\})(\forall l \in \{1, ..., l^*(s,k)\}) \Big(P_\Sigma(\rho_s^{(k,l)}|\kappa_s^{(k)}) = P^*(\rho_s^{(k,l)}|\kappa_s^{(k)})\Big)$$

(R2) **Preservation of interaction structure:**
Except from the modifications induced by the revision assignments, $P_\Sigma(V)$ preserves all probabilistic dependencies of $P(V)$.

3.2 *Discussion*

Essentially, the required preservation of the interaction structure coincides with the decision-theoretical presupposition that the revision operator does not modify the cross product ratios of conditional events outside the influence areas of the revision assignments (**principle of minimal change**).

It can be proven (see, f.e. [8]) that in case of existence, the solution of the revision problem $(P(V), \Sigma)$ is uniquely defined. $P_\Sigma(V)$ can be calculated as the limit probability distribution if the revision procedure of **iterative proportional fitting** with parameters Σ is applied to the initial distribution $P(V)$.

From a practical point of view, in most cases of real world applications of sufficient complexity, we have to take into account that revision problems $(P(V), \Sigma)$ specified by human experts are not solvable. The reason for this observation is the fact that revision structures $\Sigma = (\sigma_s)_{s=1}^S$ tend to contradict some of the restrictions given by the zero values of the initial probability distribution $P(V)$. Note that assignment components $P^*(\rho_s^{(k,l)}|\kappa_s^{(k)}) > 0$ may induce to change some probabilities $P(\omega) = 0$ to a strictly positive value. This kind of modification is not conform to the dependency preservation requirement of the revision operator, as zero probabilities show the absence of any interaction structure. Hence, a resulting probability $P_\Sigma(\omega) > 0$ would introduce a new interaction structure, which is the typical focus of the (in some sense complementary) updating operations.

Resulting probabilities $P_\Sigma(\omega) > 0$ may be introduced directly by only one revision assignment σ_s (which can be easily detected and coped with) or by any subset of Σ. In the latter case the revision structure Σ contains inconsistencies which cannot be detected and dealt with by trivial means. In order to deal with such inconsistencies we need first to analyse their properties and categorise these inconsistencies.

4 Categorisation of Inconsistencies

As already mentioned in the previous section, inconsistencies may occur during revision. Inconsistencies can be roughly classified into two categories. In this section we will differentiate between inner and outer (in-)consistency. Inner consistency is a property of a revision structure alone whereas outer consistency always depends on the initial distribution $P(V)$, especially its zero values. Moreover, we introduce inner and outer inconsistency criteria for revision problems which finally allows us to determine the degree of inconsistencies.

4.1 Definitions

In order to handle the typical inconsistencies which arise during the revision, we introduce the following definitions:

Definition 2 (Inner consistency). Let $(P(V),\Sigma)$ be a revision problem. A revision structure Σ shows the property of **inner consistency**, if and only if there exists a probability distribution that satisfies the revision assignments of Σ.

Note that this definition is conform to the condition (R1) of definition 1.

Definition 3 (Outer inconsistency). Let $(P(V),\Sigma)$ be a revision problem with the property of inner consistency. $(P(V),\Sigma)$ shows the property of **outer inconsistency**, if and only if there is no solution of this revision problem.

Definition 4 (ε-modification of a revision problem). Let $(P(V),\Sigma)$ be a revision problem, and let ε be a (sufficiently small) positive real number. Furthermore, assigning $r := |\{\omega \in \Omega(V) | P(\omega) = 0\}|$, let

$$P_\varepsilon(\omega) \stackrel{Df}{=} \begin{cases} P(\omega) \cdot (1 - r\varepsilon), & \text{if } P(\omega) > 0 \\ \varepsilon, & \text{if } P(\omega) = 0 \end{cases}$$

Then, $(P_\varepsilon(V),\Sigma)$, is called the **$\varepsilon$-modification** of $(P(V),\Sigma)$.

We now present some results that are useful for the recognition and handling of inconsistencies of revision problems.

4.2 Inner Inconsistencies

Theorem 1 (Inner inconsistency criterion for revision problems). *A revision problem $(P(V),\Sigma)$ shows the property of inner consistency, if and only if the revision procedure (iterative proportional fitting) applied to its ε-modification $(P_\varepsilon(V),\Sigma)$ converges to a limit distribution $P_\varepsilon^\Sigma(V)$.*

Corollary 1 (Sufficient condition for revisability). *In case of a strictly positive distribution $P(V) > 0$ and inner consistency of its structrue Σ of assignments, there always exists the uniquely determined solution $P^\Sigma(V) \equiv P_\varepsilon^\Sigma(V)$ of the revision problem $(P(V),\Sigma)$.*

To achieve inner consistency, one may restrict the revision structure Σ, so that it fulfills the following conditions:

1. $(\forall s \in \{1,...,S\})(\forall t \in \{1,...,S\})(s \neq t \Rightarrow R_s \cap R_t = \varnothing)$
2. $(\exists(V,<))(\forall s \in \{1,...,S\})(\rho \in R_s \Rightarrow (\forall \kappa \in K_s)(\kappa < \rho))$

The first condition forbids that two different assignment components (with possibly different context schemes) specify parts of the same revision scheme. The second condition ensures that no cyclic dependencies between the assignment components are possible.

However, these conditions for the revision structure are quite restricting: Considering the presupposed distinction of context and revision schemes ($K_s \cap R_s = \varnothing$) as well as the inclusion of $K_s \cup R_s$ in one of the cliques, it turns out that (1) and (2) lead to a so-called chain graph which reflects the dependencies induced by the given revision structure. This means that all dependencies of the involved variables may be specified with the aid of a composition of directed acyclic graphs and undirected graphs. For practical purposes, whenever possible, it is desirable to establish such a dependency graph. An alternative is to use techniques of locating and removing (inner) inconsistencies rather than preventing them (see section 5).

4.3 Outer Inconsistencies

Lemma 1 (Theoretical outer inconsistency criterion for revision problems). *Given the inner consistency of its structure Σ of assignments, we observe an outer inconsistency of a revision problem $(P(V),\Sigma)$, if and only if*

$$(\exists \omega \in \Omega(V))(P(\omega) = 0 \ \wedge \ P^*(\omega) > 0)$$

holds for all probability distributions $P^*(V)$ that satisfy the revision assignments (R1).

Theorem 2 (Practical outer inconsistency criterion for revision problems). *Given the inner consistency of its structure Σ of assignments, we observe an outer inconsistency of a revision problem $(P(V),\Sigma)$, if and only if*

$$(\exists \omega \in \Omega(V))(P(\omega) = 0 \ \Rightarrow P_\varepsilon^\Sigma(\omega) >> P_\varepsilon(\omega))$$

is satisfied.

As a consequence, given inner consistency, the outer inconsistency proof for a particular revision problem $(P(V), \Sigma)$ can be obtained by application of the revision procedure to the ε-modification $(P_\varepsilon(V), \Sigma)$ of this revision problem, and testing the limit distribution $P_\varepsilon^\Sigma(V)$ with respect to the outer inconsistency criterion.

4.4 Degree of Inconsistencies

Further investigations on mass flows in inconsistency situations finally lead to the following theorem that gives a basis to handle inconsistencies of non-solvable revision problems:

Theorem 3. *Given the assumptions of the previous theorem, let*

$$Inconsistent\_tuples(P(V), \Sigma) \stackrel{Df}{=} \left\{\omega \in \Omega(V) | P(\omega) = 0 \wedge P_\varepsilon^\Sigma(\omega) >> P_\varepsilon(\omega)\right\}$$

denote the set of all tuples that are involved in the outer inconsistencies of a revision problem $(P(V), \Sigma)$. Then, this set consists of all invalid tuples ω that need significant probability mass flows (quantified by $P_\varepsilon^\Sigma(\omega)$) from ω to any valid tuples in order to remove the existing inconsistencies.

After inconsistent tuples are located one can determine the degree of the inconsistency which is given by

$$Inconsistency\_mass(P(V), \Sigma) \stackrel{Df}{=} \sum \left\{P_\varepsilon^\Sigma(\omega) | \omega \in Inconsistent\_tuples(P(V), \Sigma)\right\}.$$

$Inconsistency\_mass(P(V), \Sigma)$ reflects the whole probability (inconsistency) mass which has to be transferred to any tuples of $\Omega(V) - Inconsistent\_tuples(P(V), \Sigma)$.

5 Practical Solutions to Handle Inconsistencies

With the analysis of inconsistencies from the last section one can identify the inconsistent tuples and distribute the inconsistency mass to other tuples using the ε-modification of a revision problem. However, applying the ε-modification makes it necessary to hold in memory all tuples within the cliques C_j, even the tuples which had a zero probability before. Note that in the original revision problem zero probabilities do not need to be represented explicitly.

In our application domain the sizes of average cliques and largest cliques differ among the automobil models. In typical automobile models (like the Golf class) the largest cliques contain about $40,000$ non-zero tuples, but the maximal theoretical size[1] of these cliques is greater than 10^{14}, which makes it infeasible to apply the

[1] Size of all tuples including the zero-value tuples.

ε-modification to these cliques. Therefore practical solutions are needed to handle inconsistencies.

The main idea for the practical solution presented in this paper is to prioritise and group the revision assignments as well as apply several revisions[2] with consistency checking (and adaptation of revision assignments if necessary) until all revision assignment groups are incorporated. With this strategy it is possible to locate and remove all inconsistencies without the need to differentiate between inner and outer inconsistencies anymore.

In the following we will speak about inconsistencies between revision assignments. Please note that the intial distribution $P(V)$, especially its zero-values, is always part of such inconsistencies but will be regarded as not adaptable to solve inconsistencies and therefore $P(V)$ is not mentioned all the time.

5.1 Prioritising and Grouping the Revision Assignments

Given a potentially inconsistent revision problem $(P(V), \Sigma)$ one can often specify which revision assignments σ_s are more important than others, so that in case of an inconsistency between these revision assignments only the least important one should be adapted.

However, sometimes it is impossible to decide which one of two revision assignments is more important. In fact it might be needed that two or more revision assignments get the same priority. Such revision assignments are grouped together. In case of an inconsistency within such a group all its revision assignments should be adapted according to the principle of minimal change.

The set of revision assignments is divided into n partitions $\mathbb{S}_i$, so that $\Sigma = \bigcup_{i=1}^{n} \mathbb{S}_i$ and $\mathbb{S}_i \neq \varnothing$, $\mathbb{S}_i \cap \mathbb{S}_j = \varnothing$ for any $1 \leq i, j \leq n$ with $i \neq j$.

5.2 Iterative Revision with Consistency Checking

After the revision assignments are grouped and ordered we can start with an empty set Σ_0 which is consistent with the initial probability distribution $P(V)$. In each iteration we take the consistent set Σ_{i-1} $(1 \leq i < n)$ and perform a meta revision by adding the revision assignments of $\mathbb{S}_i$. This meta revision results in a new consistent set Σ_i where the revision assignments of $\mathbb{S}_i$ are adapted (where necessary) to achieve consistency with Σ_{i-1}.

Each meta revision operates using (up to) two phases. In the first phase a single revision is performed on the set $\Sigma_{i-1} \cup \mathbb{S}_i$. If this revision converges, the set is consistent, otherwise the revision assignments of $\mathbb{S}_i$ introduce an inconsistency. This inconsistency is resolved by applying partition mirrors to $\mathbb{S}_i$. By applying these partition mirrors, the revision converges and the revision assignments $\sigma_s \in \mathbb{S}_i$ are adapted to σ_s^* if necessary. The main idea of partition mirrors is to mirror variables into the network structure, couple the states of these variables to their origins by

[2] For detailed information about a single revision step, please see [8].

suitable initial distributions, and reformulate the assignments in order to set probabilities to these new variables. For detailed information about partition mirrors see [10].

After n steps every revision assignment has been tested for consistency with the other revision assignments. In case of inconsistencies the least important revision assignments have been adapted so that we finally have a consistent set Σ_n of revision assignments.

The resulting algorithm reads:

$i := 0$; $\Sigma_i := \varnothing$; (∗ initializing empty set ∗)
repeat (∗ iterative meta revision ∗)
$i := i + 1$ (∗ iteration counter increment ∗)
phase 1: test consistency
  do single revision with $\Sigma_{i-1} \cup \mathbb{S}_i$
  if probability distribution converges:
    $\Sigma_i := \Sigma_{i-1} \cup \mathbb{S}_i$
  else {
phase 2: applying partition mirrors
  do single revision with $\Sigma_{i-1} \cup \mathbb{S}_i$ applying partition mirrors for $\mathbb{S}_i$
  $\Sigma_i := \Sigma_{i-1} \cup \mathbb{S}_i^*$ where $\mathbb{S}_i^*$ contains adapted revision assignments }
until $i = n$ (∗ all revision assignments incorporated ∗)

The result of each iteration (meta revision) is a consistent set of revision assignments Σ_i as well as a modified probability distribution $P^{\Sigma_i}(V)$. The i-th meta revision can be performed on either the initial probability distribution $P(V)$ or on the resulting distribution of the prior iteration $P^{\Sigma_{i-1}}(V)$. Using the resulting distribution of the iteration before, one benefits from the already incorporated assignments Σ_{i-1}.

With highly inconsistent revision assignments it may be practical to skip phase one of the meta revision and assume that the assignments in $\mathbb{S}_i$ introduce a new inconsistency. By skipping phase one, calculation time can be saved but the application of partition mirrors also introduces additional calculation time.

If inconsistencies are rare it is possible to further group the revision assignments so that the number of single revisions can be reduced. In case of an inconsistency with a set $\bigcup_{\text{any}\, i} \mathbb{S}_i$, this set has to be divided again to locate the $\mathbb{S}_i$ which introduces the inconsistency.

6 Conclusion

In this paper we analysed inconsistencies which may occur during Markov network revision. We identified two inconsistency categories, namely inner inconsistency and outer inconsistency. With the help of the *ε-modification of a revision problem* we analysed the differences and special properties of these two inconsistency categories. Consequently, it was possible to specify the degree of inconsistencies based on the number of tuples involved as well as their probability mass. Not only theoretical considerations to identify and resolve inconsistencies are presented, although a practical approach to handle inconsistencies was proposed in this work. This approach is based on prioritising and grouping revision assignments so that iterative revision operations can be used to identify and resolve inconsistencies efficiently.

Automatically resolved inconsistencies are very beneficial for the user since it reduces the manual effort drastically. However, sometimes additional information is needed in order to explain to the user why some assignments have been adapted. Therefore an automatically generated explanation of inconsistencies would be very helpful. Such an explanation could be determined in two steps. In the first step a minimal set of revision assignments causing the inconistency could be generated. In the second step an argumentation line could be given in order to explain the inconsistency. The automatic explanation of inconsistencies seems to be more complicated than expected (especially the second step). This task is subject to further research.

Acknowledgements. The authors thank Dr. Heinz Detmer and Dirk Schäfer from Volkswagen Group for supporting our work in the context of the software system EPL. Furthermore we give thanks to Katharina Tschumitschew, Fabian Schmidt, and Martin Försterling for their useful advice when writing this contribution.

References

[1] Borgelt, C., Kruse, R.: Graphical Models—Methods for Data Analysis and Mining. J. Wiley & Sons, Chichester (2002)
[2] Detmer, H., Gebhardt, J.: Markov-Netze für die Eigenschaftsplanung und Bedarfsvorschau in der Automobilindustrie. KI – Künstliche Intelligenz (03/01) (2001) (in German)
[3] Gabbay, D., Smets, P. (eds.): Handbook of Defeasable Reasoning and Uncertainty Management Systems. Belief Change, vol. 3. Kluwer Academic Press, Netherlands (1998)
[4] Gabbay, D., Smets, P. (eds.): Handbook of Defeasable Reasoning and Uncertainty Management Systems. Abductive Reasoning and Learning, vol. 5. Kluwer Academic Press, Netherlands (2000)
[5] Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. MIT Press, Cambridge (1988)
[6] Gebhardt, J., Kruse, R.: Knowledge-Based Operations for Graphical Models in Planning. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 3–14. Springer, Heidelberg (2005)
[7] Gebhardt, J., Detmer, H., Madsen, A.: Predicting Parts Demand in the Automotive Industry – An Application of Probabilistic Graphical Models. In: Proc. Int. Joint Conf. on Uncertainty in Artificial Intelligence UAI 2003, Bayesian Modelling Applications Workshop, Acapulco, Mexico (2003)
[8] Gebhardt, J., Borgelt, C., Kruse, R., Detmer, H.: Knowledge revision in Markov Networks. Special Issue "From Modelling to Knowledge Extraction" 11(2-3), 93–107 (2004)
[9] Jensen, F.: An Introduction to Bayesian Networks. UCL Press, London (1996)
[10] Klose, A., Wendler, J., Gebhardt, J., Detmer, H.: Resolution of inconsistent revision problems in Markov Networks. In: 6th International Conference on Soft Methods in Probability and Statistics. AISC. Springer (2012)

[11] Kollar, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
[12] Lauritzen, S.L.: Graphical Models. Oxford University Press (1996)
[13] Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, Series B 2(50), 157–224 (1988)
[14] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, 2nd edn. Morgan Kaufman, New York (1992)
[15] Whittaker, J.: Graphical models in applied multivariate statistics. Wiley, Chichester (1990)