Adaptable Markov Models in Industrial Planning

Jörg Gebhardt*, Frank Rügheimer†, Heinz Detmer‡ and Rudolf Kruse†
*Intelligent Systems Consulting, Leonhardstr. 19, D-29227 Celle, Germany
EMail: PD.Dr.Joerg.Gebhardt@t-online.de
†Dept. of Knowledge Processing and Language Engineering
Otto-von-Guericke-University of Magdeburg, Universitätsplatz 2, D-39106 Magdeburg, Germany
EMail: {ruegheim, kruse}@iws.cs.uni-magdeburg.de
‡Volkswagen Group, K-DOB-11, D-38436 Wolfsburg, Germany
EMail: Heinz.Detmer@Volkswagen.de

***Abstract*— A significant number of scientific and economic problems is characterised by a large number of interrelated variables. But with larger variable number, the domain under consideration may grow fast, so that analyses and reasoning become increasingly difficult. Graphical models allow to represent the combined distributions compactly and are suitable for dealing with uncertain and incomplete information.**

In this paper we describe their application to a problem of industrial planning. We also demonstrate how the iterative planning process can be supported by allowing the users to adapt the model using revision and updating operators. Moreover we discuss the problem of inconsistent inputs.

I. Introduction

A significant number of scientific and economic problems is characterised by a large number of variables. But with larger variable number, the domain under consideration may grow fast, so that analyses and reasoning become increasingly difficult. Sophisticated preprocessing techniques often help to reduce the dimensionality of the input space, but may not always be sufficient.

Other solutions are based on performing decomposition. They try to deal with this problem by making use of (conditional) independence between variables and work with lower dimensional subdoma)ns. The use of graphical models [1], [2], [3], [4] such as Bayesian networks [5] and Markov networks [6] for representing probability distributions, as well as possibilistic networks [7] in order to represent possibility distributions, exemplifies this approach.

We already have been presented with many successful applications in the field of knowledge representation and reasoning with inference networks, e.g. HUGIN [8] and POSSINFER [9]. More recent research has also brought advances in data-driven generation of such networks [10], [11]. Using this method a network can be learned from databases that summarise information required for dealing with the given problem.

However, one can not always rely on all relevant data to become available at the same time. Yet, in many applications and especially in planning, it is advantageous to have models available as early as possible, even if results are rough in the beginning.

Consider for example a newsagent working in a popular coastal holiday area, who wants to order their merchandise for a certain week. The demand for newspapers of different type will vary, depending on preferences, size and composition of current tourist population. At an early stage most of the available information will refer to variables like school holidays, hotel capacity or reservation status. Moreover, the newsagent may have access to data about comparable situations in earlier years. But reliable data about weather in the week they are planning for should hardly be obtainable. A graphical model generated from these sources may already produce rough estimates. Based on these estimates, even though probably not very accurate, the newsagent orders books and magazines (repeated orders for these items are possible if necessary). Also, there is word about a new type of rapidly growing algae that has appeared in some areas of the coast.

As the week to be planned for approaches, more data becomes available: the tourist department publishes statistics about the current season and how algae density affects tourism. Since our newsagent might be interested in the effect of this formerly unknown factor, they alter their model and supplement it with the additional data provided by the tourist department. At this point it would be convenient if a representation of the new influence could be integrated into the existing model without losing the benefits of any previously entered settings.

Finally, only a short time in advance, the dailies are to be ordered. By then the newsagent will require very detailed information about their customer spectrum and behaviour. Using observations from previous days, providing this information should prove not much of problem though. In our example scenario the model has become very accurate by now.

We have just seen some of the problems typically associated with planning problems, namely

- interrelated variables (hotel reservations, school holidays)
- uncertain and incomplete information, especially in the beginning
- pieces of evidence that address different subsets of the variables
- parameter values that are subject to change (tourist composition)
- integration of previously unobserved influences (algae)
- delayed availability of relevant facts.

While standard inference networks already provide good solutions with respect to the first three issues, solving the remaining problems requires some additional attention. All three of these points are akin in that they refer to alterations in an already existing model [12].

A transition from static models to adaptable ones that can be supplemented or altered as the necessity arises, can enable us to deal with this situation. Thus it would be rewarding to provide data-fusion techniques. Our work refers to the problem of item planning and calculation of part demand in the automotive industry. It was part of a project called EPL (Eigenschaftsplanung, item planning) [13], which was aimed at developing a supportive tool for the mentioned tasks.

II. Item Planning in the Volkswagen group

The problem of item planning and the closely related calculation of part demand in automobile production usually have to be solved several weeks or even a couple of months in advance, so that it is possible to adjust logistics and distribution of production resources accordingly. Nevertheless, unforeseen events or additional information may require reviewing earlier results. Therefore item planning has to be seen as an iterative process.

In order to best meet customers needs, the Volkswagen Group follows a marketing strategy based on highly adaptable car models. The customer is offered choice between several alternatives for each item of equipment. Also there may be adaptations for local markets, e.g. right hand drive variants for the UK. This strategy leads to a very large number of possible item combinations. But, because of technical, legal or simply practical issues, not every imaginable variant will represent a valid, functional and profitable product.

This special situation has to be considered when calculating part demand. The traditional approach of multiplying variant specific demand with estimated variant frequency and adding up over all fabricated variants, requires the number of possible variants to be low. Therefore this method cannot be applied here. Fortunately it is viable to identify short characteristic item combinations which are sufficient to determine whether a specific part is required in a car or not. So instead of working with complete variant specifications, the problem of parts demand calculation can be reduced to estimating the frequency of these characteristic combinations alone. In our application the number of item combinations for calculating part demand was about 100 000 per vehicle class.

At Volkswagen, each automobile model, e.g. Golf V, is described by a set of variables called item families. In the case of Golf V the number of these item families well exceeds 200. To address a specific variant of a model, each item family is associated with exactly one alphanumeric code that denotes the precise item to be implemented. To reflect the relations between item families, the item distribution over all variants is modelled using a Markov network. This approach allows the high-dimensional probability distribution to be decomposed, using some item families or groups of item families to be (conditionally) independent from one another. The engine type, for instance, can be expected to be strongly related to the type of gear, but knowledge about it presumably does not tell much about interior colour. Also, the presence of heated front seats may require a more powerful generator to be installed.

A. Data Sources

In the project EPL several major information sources had to be considered:

- a system of rules, derived from:
  - technical restrictions
  - laws according to local jurisdiction
  - economic issues
  - local market requirements
  - capacity restrictions
- samples from historical production data
- analyses on market trends
- changes in item palette

The rules system provides a description of the (automobile) model as used by the sales department. It is required to restrict the model to those variants, that represent valid products. The rules system is relational in nature and describes sets of items from different families that must or must not occur together. Currently it consists of circa 10 000 technical rules and even more sales-oriented ones.

Data about past production can under certain conditions be used for estimating demand. But the sample should be carefully selected, as the data does not always represent market demand. For example, special offers and temporarily capacity restrictions can influence the statistics significantly. If an appropriate sample is found, it can be used to initialise the model.

Another data source are analyses of market trends. They are provided by external tools or experts. The experts may also specify additional knowledge or estimates based on sources inaccessible to the system. After resolving any detectable contradictions the information can be integrated into the model. As an illustration one could, for instance, think of a newly introduced tax on certain fuels that causes a market's preference to shift from one motor type to a different one. Data on past production cannot reflect this, but experts could probably provide good predictions for the changes in future demand with respect to motor type. Finally one has to consider occasional changes in the item palette. This may lead to some item combinations to become invalid or previously impossible ones to become valid. Such events require the model to be adapted using revision and updating operators (see IV-A and IV-B).

III. Representation and Retrieval

EPL employs a model of the joined probability distribution over all item families. Assume we had 200 item families with each of them having only two alternative values (in fact there may be up to 100), then we would end up with $2^{200} > 10^{60}$ possible combinations. A direct representation would be infeasible, not to mention time required to perform any calculations on such a model.

Instead of that, the distribution is represented by a Markov network, where the original probability distribution is specified by a family of distributions on the cliques of a conditional independence graph with hypertree structure.

A. Model Initialisation

The task of finding an appropriate decomposition usually requires searching the space of possible graphs. Algorithms have been presented for both probabilistic [14] and possibilistic networks [10], [15]. However, in our case, we can take advantage of the prior knowledge given by the rules system. Instead of testing a large number of possible independence graphs, a model structure is generated directly from the rules system. Because the rules were originally designed to cover relevant dependances it is reasonable to assume that most, if not all of the important interactions are represented in the rulebase. Thus variables that often occur together in the rulebase can be expected to have a complex structure of interaction. If such variables are assigned to the same clique the information about their interaction will be preserved in a local distribution of the resulting network. In constrast absence of such rules suggests that no direct relation between those variables exists.

Given the structure, we also have to learn the local distributions on the cliques (hyperedges in the corresponding hypergraph) themselves. In approximation, the required information can be extracted from the examples in the historic database.

The problem of clique size ought to be given additional attention here, since the benefit of decomposition depends on the cliques to become sufficiently small to handle. If the rule structure does not lead to a suitable model structure one can try dropping some of the less important rules. This is equivalent to introducing additional independence assumptions and can lead to simpler, but less accurate models.

B. Application of the System

Once the model has been generated, it can be used to support planners in decision making. Depending on what tasks are to be performed one can choose operations for accessing or altering the state of the model. Common queries simply refer to marginalizations of the represented distribution. If there is at least one clique that contains all the variables that are concerned, the desired edge distribution can be retrieved by simply marginalizing the local distribution in that clique. But even if this does not apply, the distribution can be obtained by combining distributions from multiple cliques using the independence relations described in the graphical component of the model.

C. Focusing

Another important aspect is the simulation of possible scenarios, e.g. possible versions for market development, but also situations like capacity restrictions or supply shortage. The relevant operation is called focusing. Focusing [12] can be achieved by performing any kind of evidence-driven conditioning on a set of input variables and propagating the information. The planner then can observe, how the distributions for other variables and their combinations are affected.

Instantiation of variables, as it is usually implemented in diagnostic tools, can be considered a special case of this operation, with all the probability mass assigned to one value per given variable.

At the user interface the probabilities retrieved from the model are being reinterpreted in terms of relative frequency. Provided the total number of cars to be produced is available, absolute numbers are shown as well. This alternative presentation can help those users, who do not have a background in statistics or probability theory.

By combining multiple requests it is even possible to implement wildcards or groups of possible variable combinations to be handled together, allowing queries like: “What number of cars will be of either blue or green colour?”. In addition to that the interface can be used to introduce a context, e.g. request the percentage of cars equipped with sunroof among those that have cruise control and additional airbags installed. This is achieved by calculating a conditional probability from both a request concerning the combination from the context (here: cruise control and airbags) alone and another one for all of the items combined.

IV. Belief Change in Markov Networks

As indicated before, it is sometimes necessary to adapt the model to recent knowledge. For this purpose one needs revision and updating operators [16], [17], [18], [12].

A. Revision

Revision is the alteration of the represented distribution within the frame of an existing model structure, i.e. although the frequency of item combinations may be changed in the revision process, it is required that the state of forbidden item combinations does not do so. The operation is based on the principle of minimal change. Revision is performed by locally introducing edge distributions into the Markov network. Like with focusing, the local modifications of distributions are propagated. But in contrast to the operations used in retrieval, the changes made during revision are permanent, as the modified clique distributions replace those already stored in the model. The alterations to the model are the least ones required to integrate the new settings. Therefore most of the probabilistic interaction structure already represented in the model is preserved.

If multiple local distributions have to be changed, the desired revision is achieved by propagating the settings one after another. Since any local change may affect other areas in the model, processing one of the settings may well invalidate part of the models adaptations to previous settings in the sequence. However, by iterating the process the model can often converge to a state of stable compromise consistent with all settings.

Still, if the listed settings are in conflict with each other or affect completely forbidden item combinations, it will remain unstable and equilibrium cannot be reached. In the first case this is due to an external conflict (there can be no consistent, accurate and complete model for contradictory evidence), in the latter the updating operator can be employed.

B. *Updating*

The idea of revision is to reassign probability mass in such a way that the original probability distribution is preserved as much as possible. If the total probability assigned to an item is reduced and no further settings apply, all combinations with this item will have their probability reduced, yet their ratio will be preserved. A problem occurs, however, if probability mass is supposed to be assigned to an item combination currently associated with a rate of zero. Since the probability of all extended item combinations has to be zero as well, no information about their ratios can be derived. The updating operation avoids this problem by asking the user for a reference combination. Since the intended users are experts for their domain, they should be able to provide such a combination from experience. The missing information on probabilistic interaction structures is simply copied from the reference combination.

Restricting on a subset of the variables allows each user to focus on their domain of expertise, while the graphical model handles global consequences caused by local changes in the distribution.

V. Dealing with Inconsistency

The input provided by a user consists of a list of settings. Each setting specifies an estimated rate for an item combination or a set of item combinations. From the user's point of view, it would be convenient if any setting entered could be successfully integrated. However there can be settings or combinations of settings that, if accepted, would violate model consistency. The following demonstrates an inconsistent set of inputs:

1) 30% of the production will have radio type A
2) 60% of the production will have radio type B
3) 30% of the production will have a navigation system on board
4) 50% of the cars equipped with a navigation system will have radio type C

The first two settings are consistent with each other, provided that there is at least one additional value for "radio" type to which the remaining 10% of cars can be allocated. The third statement does not refer to radio type at all, so it is fully compatible with the previous ones. However, when adding the final statement, a contradiction becomes apparent: half of the cars having a navigation system represent 15% of the total production. But in this case we would assign radios for 105% of the total production. Since every car must have exactly one value for radio ("none" being a valid variable value), this is a contradiction, leading to unsuccessful revision or updating operations.

The first step in dealing with inconsistencies is to detect them. But this is not always easily accomplished. As illustrated in the example, the presence of inconsistency may not be obvious. When checking consistency one does not only have to consider a single setting but all parallel settings as well. Additionally, in revision no probability mass must be assigned to previously forbidden item combinations. Thus the current model itself has to be regarded.

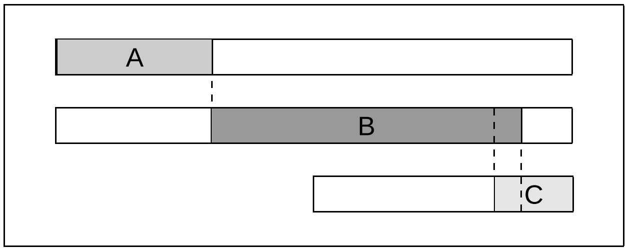

Fig. 1. An inconsistent set of inputs

Fortunately revision and updating operations can help identify and localise inconsistency. However, if the operations are not carried out successfully, they will require a considerable amount of computation. Therefore it would be desirable to detect inconsistent settings beforehand. Moreover, if inconsistent settings are found they can be modified to resolve the contradiction. Since comprehensively testing for consistency can be very complex itself, a compromise between the amount of preprocessing provided and the cost caused by undetected inconsistency is required.

In the EPL data, most of the inconsistency can be found by relatively simple means. Subset relations provide a good approach to testing. Any variant that matches the item combination (engine A, gear C, battery F) will also match (engine A) alone. Therefore the rate assigned to the abbreviated combination must be greater or equal to the rate for the more specific tuple. Also, if there was a further combination (engine A, battery E) the minimum rate for (engine A) would have to be increased by the rate of that second combination. This is due to battery type E and F being mutually exclusive. The maximum rate (1.0 if no further information is available) for (engine A) can be used to calculate upper boundaries for the frequency of the more specific item combinations. Finally the distribution of probability mass within the item families can lead to even more restrictive boundaries.

For the revision operation, we also have to ensure that the settings in the input do not assign a positive rate to previously forbidden item combinations. Therefore we obtain additional restrictions from the existing model. Whenever a new setting is to be tested, it will be initialised with the restrictions from the model. If the rate specified by the setting is outside the interval, the setting will either be rejected completely or – depending on the user's preference – be modified. We then build up a data structure that represents the item combinations the setting refers to, their indicated rate, and the references to the previous settings with respect to the subset relation. In the next step the rate boundaries are modified using the information from the related settings.

By convention the settings are processed in the order of their priority. In case of conflict, the less important setting is modified or if necessary completely rejected. When altering a setting, the target rate will be set to the nearest possible rate, i.e. one of the two boundaries.

VI. Software Development

The already mentioned project EPL was initiated in 2001 by Corporate IT, Sales, and Logistics of the Volkswagen Group. The aim was to establish for all trademarks a common item planning system that reflects the modelling approach based on Markov networks.

System design and most of the implementation work of EPL is currently done by Corporate IT. The mathematical modelling, theoretical problem solving, and the development of efficient algorithms, extended by the implementation of a new software library called MARNEJ (MARkov NEtworks in Java) for the representation and the above-mentioned functionalities on Markov networks have been entirely provided by ISC Gebhardt.

The worldwide rollout of the system EPL to all trademarks of the Volkswagen Group will be realized during the year 2004. Up to 15 system developers implement the client-server architecture in Java. The planned configuration uses 6 to 8 Hewlett Packard machines with 16 GB of main memory and 4 AMD Opteron 64-Bit-CPUs each, and a terabyte storage device. The system is running Linux and an Oracle database system.

References

[1] E. Castillo, J. M. Guitérrez, and A. S. Hadi, *Expert Systems and Probavilistic Network Models*. New York: Springer-Verlag, 1997.

[2] S. L. Lauritzen, *Graphical Models*. Oxford University Press, 1996.

[3] J. Whittaker, *Graphical models in applied multivariate statistics*. Chichester: Whiley, 1990.

[4] R. G. Cowell, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*. New York: Springer-Verlag, 1999.

[5] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 2nd ed. New York: Morgan Kaufman, 1992.

[6] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society, Series B*, vol. 2(50), pp. 157–224, 1988.

[7] J. Gebhardt, "Learning from data: Possibilistic graphical models," in *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, D. M. Gabbay and P. Smets, Eds. Dordrecht: Kluwer Academic Publishers, 2000, vol. 4: Abductive Reasoning and Learning, pp. 314–389.

[8] S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen, "HUGIN—a shell for building bayesian belief universes for expert systems." in *Proc. 11th Int. J. Conf. on Artificial Intelligence*, 1989, pp. 1080–1085.

[9] J. Gebhardt and R. Kruse, "POSSINFER—a software tool for possibilistic inference," in *Fuzzy Set Methods in Information Engineering: A Guided Tour of Applications*, D. Dubois, H. Prade, and R. Yaeger, Eds. John Wiley & Sons, 1995.

[10] C. Borgelt and R. Kruse, "Learning possibilistic graphical models from data," *IEEE Transactions on Fuzzy Systems*, pp. 159–172, 2003.

[11] ——, "Operations and evaluation measures for learning possibilistic graphical models," *Artificial Intelligence*, pp. 385–418, 2003.

[12] P. Gärdenfors, *Knowledge in flux: modeling the dynamics of epistemic states*. Cambridge, Mass.: MIT Press, 1988.

[13] H. Detmer and J. Gebhardt, "Markov-Netze f ür die Eigenschaftsplanung und Bedarfsvorschau in der Automobilindustrie," *KI – K ünstliche Intelligenz*, no. 03/01, 2001, (in German).

[14] C. N. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inform. Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[15] C. Borgelt and R. Kruse, "Local structure learning in graphical models," in *Planning based on Decision Theory (Proc. 6th Int. Workshop, Udine, Italy 2002)*, ser. CISM Courses and Lectures 472, Springer, Ed., Wien, Austria, 2003, pp. 99–118.

[16] J. Gebhardt, "The revision operator and the treatment of inconsistent stipulations of item rates," Project EPL: Internal Report 9. ISC Gebhardt and Volkswagen Group, K-DOB-11, Wolfsburg, Germany, 2001.

[17] J. Gebhardt and H. Detmer, "Revisions and updating of probabilistic networks for item planning and capacity management," in *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002)*, Annecy, July 2002.

[18] J. Gebhardt, H. Detmer, and A. Madsen, "Predicting parts demand in the automotive industry – an application of probabilistic graphical models," in *Proc. Int. Joint Conf. on Uncertainty in Artificial Intelligence (UAI'03, Acapulco, Mexico)*, Bayesian Modelling Applications Workshop, 2003.