Analysis of survival times using Bayesian networks

Helge Langseth
Norwegian University of Science and Technology, Department of Mathematical Sciences

ABSTRACT: In this paper we construct a Bayesian network to analyze survival times of mechanical equipment based on a set of covariates (e.g., environmental data, maintenance philosophy etc.). Both the predictive ability and the data exploratory features are investigated, and compared to the results obtained by standard methods. The methods are tested using a real-life dataset from the OREDA database.

1 INTRODUCTION

The proportional hazards model has been the state of the art for analysis of datasets of survival times since Cox regression was introduced in 1972 (Cox 1972). The purpose of the analysis is to look into the relationships between the different environmental conditions of some mechanical equipment (the covariates) and the survival times of that equipment (the response variable). This relationship is interesting for at least two reasons: First and foremost, it gives insight into the equipment's failure mechanisms and into the kind of environment the unit is exposed to. This insight makes the analyst able to propose cost effective actions to increase the survival time of the mechanical equipment. Secondly, the analyst will be able to predict availability of other equipment working under similar conditions, and thus be able to document critical reliability parameters for that equipment. Although Cox regression has many valuable features, like the ability to handle censored data and it's well understood model assumptions, some desired features are absent from the standard method. Like for other regression models, the main focus is on the statistically significant covariates, the hidden (second-order) influences are not as easily investigated. This is a problem, both for the deeper understanding of the dataset, and when it comes to handling of missing covariates. Furthermore, the proportional hazard assumption is not always appropriate (Henderson 1995).

In this paper we employ probabilistic graphical models to discover the covariance structure. Both the correlations between the different covariates and between covariates and the response variable are of interest. We will thereafter use the discovered relationships to predict the survival times of other units working under similar conditions. Furthermore, we will use the correlation structure to draw some qualitative conclusions. Our main concern is to understand the failure mechanisms in order to the reduce the rate of critical failures. The prediction of new failure times are merely to validate the obtained results. The method is used on a real-life dataset from the "Offshore REliability DAta" (OREDA) database (Sandtorv et al. 1996), (OREDA-97 1997). The OREDA data collection has been ongoing since the early eighties, and has for the last seven years been maintained by SINTEF Industrial Management. Up to the current phase IV, some 24.000 offshore equipment units with about 33.000 failures have been entered into the database.

In section 2 the data sample is presented in further detail, section 3 is devoted to some standard theory behind graphical probabilistic models, and to the results we obtain when we employ the methodology on our data sample. The results are then compared to the results from Cox-regression in section 4, and some conclusions are drawn in section 5.

Attribute	No. classes
Installation ID	6*
Geographic location	3
System code	3*
Exposure to environment	3
Gas turbine subunits	2
Design class	2
Manufacturer of unit	4*
Operating mode	3
Planned prev. maintenance	3*
Actual prev. maintenance	3*
Severity of a failure	2*
Time to Fail (Operating hrs.)	5*

Table 1: Number of classes of different attributes. Entries marked with $*$ indicates that the numberer of classes has been reduced through expert judgment. This was done both to ensure data quality, and to reduce the number of parameters to fit in the model. Note that Time to Fail and the PM values are changed from continuous to discrete variables because of software restrictions

2 THE DATA SAMPLE

To check the performance of the probabilistic graphical networks, we use parts of the OREDA IV Gas Turbine dataset. Only the subsystems "Control&Monitoring" and "Gas Generators" where included in the study. The dataset consists of 219 mechanical units on 29 different offshore installations, with a total of 2921 failures and 300 censored survival times.

Each event (failure or censoring) is described using twelve different attributes, ten to describe the inventory, one takes care of the *severity class*, and the last holds time to failure, see Table 1. Each component is followed only for a period of time, so both right- and left censoring exists in the dataset. A Nelson-Aalen plot did however not indicate any trend in the dataset. Although it is not necessary for the calculation methods to work, we will therefore assume that the survival times are drawn from an exponential distribution. Attributes describing historical performance (aggregated operational time, cumulative number of failures etc.) are thus not included in the study. Note that because of the large standard deviation of the exponential distribution, we can not expect very good results from the predictors.

In OREDA, each failure is categorized as `Critical`, `Degraded`, `Incipient` or `Unknown`. A critical failure is one that causes immediate and complete loss of a system of providing its output. A degradation failure is one that compromises the function (but does not cease all of it). Incipient failures have no immediate effect upon function. For simplicity, we only distinguish between *critical* and *degraded* failures in this paper. Thus, `Incipient` failures are treated as `Degraded`, and `Unknown` are treated as `Missing`.

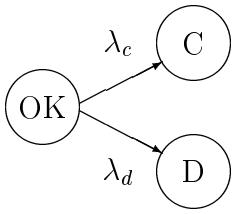

Figure 1: The failure mechanism we use, assumes that critical and degraded failures occur independently.

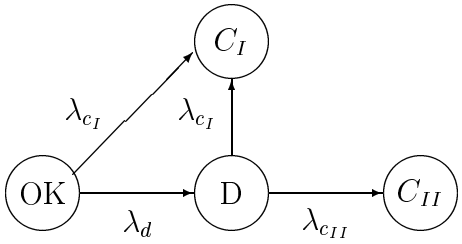

Figure 2: The component can either fail with a shock-type failure mechanism or go through a degradation process. The unit is continuously exposed for shock failures with rate λ_{C_I}, but can also fail in a degraded manner (state D), and thereafter fail critically (state C_{II}).

One critical feature of the statistical analysis of failure data is to model the failure mechanism. This model must provide means for how a failure develops, and how degraded and critical failures interact. In this paper we use a very simple failure mechanism; it is assumed that the degraded and critical failures are *independent* of each other, hence information about degraded failures will not help us in investigating the critical failures and vice versa. A more realistic model would regard the different failure modes as *competing risks*, and thus view a critical (degraded) failure as a censoring for the failure mechanism leading to degraded (critical) failures. Our failure mechanism is shown in Figure 1. A more sophisticated failure mechanism, proposed by (Hokstad and Frøvig 1996), was used by (Langseth et al. 1998) on the OREDA III re-

lease of the Gas Turbine dataset. In their model, the component can either fail with a shock-type failure mechanism, or go through a degradation process. The shock type failure mechanism leads to failure times drawn from the exponential distribution, the critical failures which come from the degradation process draw their survival times from a two-phase distribution, see Figure 2. Consult (Hokstad and Frøvig 1996) for details.

3 BAYESIAN NETWORKS

3.1 Background

We assume that we have N independent observations of a stochastic vector $\boldsymbol{X} = (X_1, \ \ldots, \ X_n)$. We label the covariates $X_1, \ \ldots, \ X_{n-1}$, and let X_n be the response variable. Our task is to establish a statistical model between the response and the covariates, and use this model to predict the response for new (unseen) vectors. To accomplish this, we employ *probabilistic graphical models* (Whittaker 1990) to estimate the full density function of $\boldsymbol{X}$. The probabilistic graphical models exploits the conditional independence structures in a dataset to create a probabilistic model, visualized by a graph. In this graph, each vertex represents an attribute of the data sample, and between each pair of vertices, there can be an edge. The edges are interpreted as an "influence path", i.e., the value of a vertex influences the value of all it's neighbors in the graph. The method has proven useful for modelling real-world problems like medical diagnosis and manufacturing control.

In this paper, we focus on a special type of the probabilistic graphical models called *Bayesian networks* (*belief network*, *Bayesian belief network*, *causal networks*), see (Pearl 1988) for a classic text. Here, all edges in the graph are *directed*, i.e., we can interpret an edge as a "mother and child relationship", and say that a parent vertex has direct influence over all its children. The graph is restricted to be *acyclic*, which means that no path in the graph leads from a vertex and back to itself (i.e., no vertex is it's own ancestor). In a statistical setting, a Bayesian network is a representation of the full multidimensional density function. It is a local representation, since all numerical values are used to quantify the interaction between one attribute and its ancestors in the graph. The absence of an edge between two vertices i and j, should be read as "X_i and X_j are independent conditioned on (some of) the other variables in the vector" (hence, the graph defines a *Markov property* over $\boldsymbol{X}$). For an example of a Bayesian network, see Figure 3. This Bayesian network is generated from the dataset, as explained in the following subsections.

System	Operating Mode		
	Running	Standby	Interm.
Crude Oil	.97	.02	.01
Gas	.05	.94	.01
Power	.13	.78	.09

Table 2: The distribution for "Operating mode", conditioned on "System code"

Every distribution function can be expressed through a Bayesian network. Number the vertices in the graph, such that all the parents of each vertex are labeled before the vertex itself (this is always possible in an acyclic graph). Then, through the *chain rule*, it is known that any statistical distribution can be expressed as

$$f(\boldsymbol{x}) = \prod_{i=1}^{n} f(x_i | X_1 = x_1, \ \ldots, \ X_{i-1} = x_{i-1}) \quad (1)$$

which is the factorization of a Bayesian network with a completely connected independence graph. If we denote the parents of vertex x_i in the graph by π_i, $\pi_i \subseteq \{x_1, x_2, \ \ldots, \ x_{i-1}\}$, we can rewrite (1) as

$$f(\boldsymbol{x}) = \prod_{i=1}^{n} f(x_i | x_1, \ \ldots, \ x_{i-1}) = \prod_{i=1}^{n} f(x_i | \pi_i) \quad (2)$$

This highlights the modular setting of the Bayesian networks. Instead of defining the full multidimensional density function, we only have to quantify the conditional density function for each vertex given its parents. These local distributions are in general far easier to elicit than the overall distribution, both when they are gathered from an expert and when they are extracted from data. The local distribution functions $f(x_i|\pi_i)$ in (2) uniquely determines the global distribution function (Heckerman et al. 1995). In Table 2, the local distribution function for "Operating mode" conditioned on "System" is given as an example.

The Bayesian networks distinguish themselves from standard regression models as they try to manifest not only the correlation between the covariates and the response variable, but also the correlation *between the different covariates*. This

is useful both when some important covariate is missing from the dataset, and when one tries to capture the effect of indirect influences.

When data is used to determine the independence graph, a popular algorithm starts with the complete graph. Then, edges are incrementally excluded like in an ANOVA procedure. At each step, all parameters in the model are fitted using Maximum Likelihood methods, and an approximate χ^2-tests is used to determine if the edge can be excluded or not, see (Whittaker 1990). This is the standard approach for undirected models, implemented among others in MIM (Edwards 1987). For *directed* models, the leading approach is to use Bayesian statistics. The analyst indicates his prior belief in each model, and the observed data is used to generate the posterior probability for each possible model given the dataset. The most probable Bayesian network(s) given the observed data will be selected, see (Heckerman et al. 1995) for a discussion. To avoid the subjectivity in the model fitting, the same prior belief can be assigned to all models, and the data is left alone to decide the best network representation. This is the method implemented in the "Bayesian Knowledge Discoverer" (Ramoni and Sebastini 1996).

Decision Networks (Horwitz et al. 1988) is an extension of Bayesian networks where the analyst can define external actions (like a change in the preventive maintenance interval or a component's material type) to be enforced onto the system. Each possible outcome is given a utility (e.g., cost of extra maintenance counting for the increased system availability) and the decision network can be employed to find the best action. A problem with this approach is that the effect of each action must be entered into the system, i.e., we have only the subjective probability assessments of an expert to rely on. If the effects are not well-understood, this will jeopardize the quality of the calculated results. Decision networks will not be the focus of the current paper.

To make predictions with Bayesian networks, one calculates the marginal density function for the response variable conditioned on the observed covariates. The mean value of this distribution will then serve as our prediction; note that we thus implicitly have decided to use a a quadratic loss function in the calculation. This calculation is in general NP-hard (Cooper 1987). However, efficient algorithms, both exact (Pearl 1986), (Lauritzen and Spiegelhalter 1988) and approximate (Jordan et al. 1998), have been developed. The techniques have been proven to manage many large-scale problems. Simulation techniques, like the Gibbs sampler (Geman and Geman 1984) are also in frequent use.

3.2 Calculation method

To generate the most probable Bayesian network from the dataset, we used the *Bayesian Knowledge Discoverer (BKD)* (Ramoni and Sebastini 1996). The program only handles categorical data, hence the continuous attributes (in our case "Time To Failure" and the PM-related attributes) were discretized before the calculation started. This software is not designed for survival time analysis, and is thus not able to treat censored data. After finding the qualitative network representation, we therefore used BUGS (Gilks et al. 1994) to estimate the conditional distribution functions $f(x_i|\pi_i)$ in (2). BUGS can handle continuous attributes, and it is also straight forward to analyze censored survival times within BUGS' modelling language.

3.3 Qualitative analysis

We have two goals for this analysis: We want to identify which covariates directly influence the "Time to Fail", and which govern the "Severity class". The motivation for the first task is obvious, if we can increase the survival times, we increase system performance. The second is also intuitive; given that a failure occurs, we would like to have a *degraded* failure, and not a *critical*. Pursuing these goals, we can make some immediate observations from the Bayesian network in Figure 3, which was judged the most probable network by BKD:

- There are two obvious candidates for increasing system performance: Change "External environment" to reduce the fraction of critical failures, and tweak the PM intervals to increase survival time.

- For a unit with no missing covariates, only the parents of "Time to Fail", namely "Installation Code", "Actual PM" and "Planned PM", will be used for the prediction of the survival time. If some of these attributes are absent from the system description, the other covariates will (indirectly) be used in the prediction.

- The "Planned PM" vertex is directly influencing "Time to Fail", giving the expression that

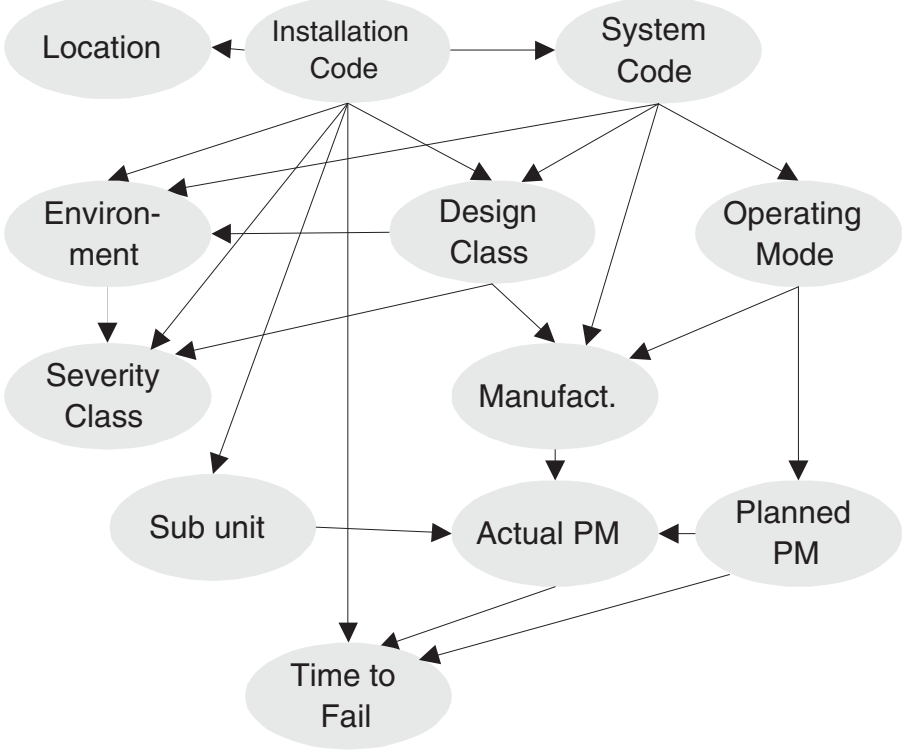

Figure 3: A Bayesian network for the Gas Turbine dataset

one can change the survival times merely by making new plans. The "Planned PM" vertex should in this setting be regarded as an indicator of safety concern at management level (for which no information is entered directly into the system) and not be read literally.

- The only path from "Subunit" to "Time To Failure" goes through "Actual PM". Hence, the time to failure is independent of what subunit we are investigating, as long as the actual preventive maintenance interval is known. That is, according to the data, the survival times of equipment from Control&Monitoring and Gas Generators are drawn from the same distribution, as long as the other covariates are known to be identical.

- "Severity Class" is not directly influenced by preventive maintenance (neither actual nor planned). This means that changing the PM interval will *not* change the fraction of failures that are critical, and the model proposed by (Hokstad and Frøvig 1996), see Figure 2, is hence too complex for our dataset. The same result was found by (Langseth et al. 1998) for the Control&Monitoring subunits. However, they discovered a relationship between performed PM and severity class for the Gas Generators that is not found in our analysis.

- A clique is a maximal set of variables in a graph that are all pairwise linked, (Spiegelhalter et al. 1993). In our graph we find, among others, one clique comprising environmental variables ("Installation code", "System code", "External Environment" and "Design Class"), one describing the unit in more detail ("System code", "Operating mode", "Manufacturer"), one for geographic location ("Geographic location" and "Installation code") and one describing PM ("Actual PM" and "Performed PM"). We can make a new graph of these cliques, Figure 4, to show how these groups interact. The vertices "Time to fail" and "Severity class" were not included in any clique to highlight their role in the analysis. The "Subunit"-vertex was added to the clique graph for completeness. Only "Maintenance" and "Environment" directly influences "Time to fail".

One might argue that the Bayesian network shown in Figure 3 is just one of many possible network configurations, and one therefore not should make qualitative interpretations of the network. To verify our results we have randomly selected subsets of the data, and generated the most probable Bayesian network from each of these subsets. Although the generated networks are not identi-

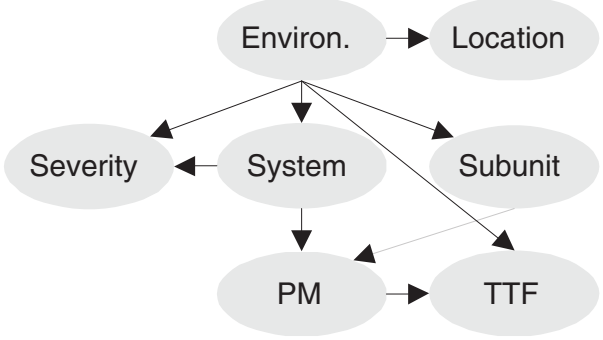

Figure 4: Graph of selected cliques

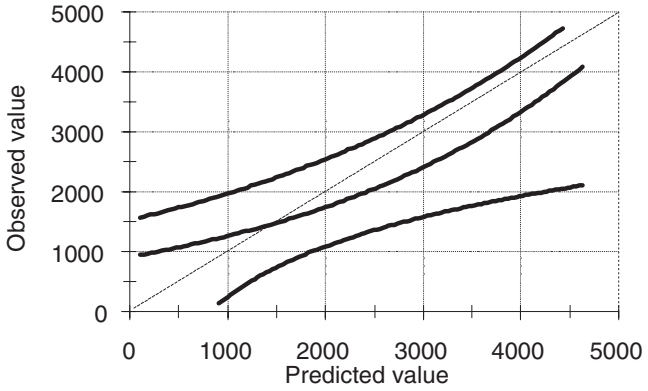

Figure 5: Plot of network predictions versus observed values

cal, the comments stated above are true for all the data subsets. The results thus appear to be valid, and not due to stochastic variations in the dataset. Log-Likelihood ratio tests wherer also employed in this validation.

3.4 Quantitative analysis

To produce numerical results, we partitioned the dataset into two parts, the *learning set* and the *test set*. The data in the test set was not used when the Bayesian network was generated. The underlying assumption is that the data in the test set and training set are from the same survival time distribution. Step one of the partitioning was to assign all censored survival times to the training set; this was done to ease the model verification. Then, the remaining failure times were parted randomly, with 30% chance of being selected to the test set. The resulting test set consisted of about 147 operating years comprising 883 failures.

Figure 5 shows observed versus predicted survival times. The smoothed trendline is drawn together with an approximate 95% prediction interval. These lines were generated by separating the datapoints into fifty bins according to their predicted values. Then the average values together with observed prediction intervals were calculated for each bin. We can see that the observed and predicted values are clearly correlated.

3.5 Model criticism

Although the Bayesian networks comprise a large class of models, we should always test the goodness of fit before any conclusions are made. We can test the model in several ways, the most obvious is to plot the covariates against the residual and look for trends. Further, as we have assumed that all data are draw from the exponential distribution, we should plot the residuals against calendar time, to verify this assumption. Neither of these plots indicated any significant departure from the expected results. A more complete treatment of model criticism through *monitors* are given in (Spiegelhalter et al. 1993).

4 THE "BASELINE" RESULTS

We have chosen to let Cox-regression play the role of the baseline predictor in this paper. The target of this work has thus not been to make the Cox-regression as sophisticated as possible, but rather to use the calculation scheme that seems to be the current standard among practitioners. We have therefore not performed any model validation, just accepted the model for what it is.

We are fitting a parameter vector $\boldsymbol{\beta}$ in the model $\lambda = \lambda_0 e^{-\boldsymbol{\beta}^T \boldsymbol{z}}$, where λ_0 is the baseline failure rate and $\boldsymbol{z}$ is the vector of covariates. The numerical results for $\boldsymbol{\beta}$ where calculated by `SPSS`. Note that the p-values are calculated using the Wald statistic, and are reported at covariate level, although the calculations are using a vector of binary values to represent the different levels of each covariate.

It is interesting to note that two of the parameters which were found to have direct influence over "Time to Fail" in Section 3, "Actual PM" and "Planned PM" are, together with "Location", the parameters which are *not* significant at 10% level. One reason why the Cox regression fails to include the PM related attributes in the model, may be that they have a very high rate (about 50%) of missing values. Note that varieties of Cox-regression handle missing values better than our method These have however not been employed in this paper. The scatterplot of the network predictions versus the regression predictions shows

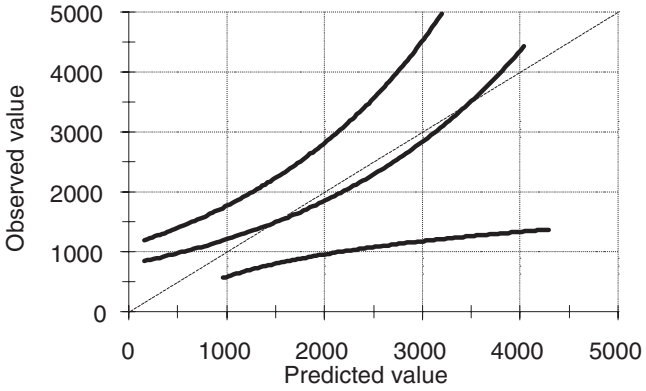

Figure 6: Plot of the Cox-regression predictions versus observed value

Parameter	p-value
Installation	.0348
Location	.7653
Sub unit	.0001
Environment	.0004
System code	.0000
Design	.0427
Manufacturer	.0000
Operating mode	.0591
Planned PM	.4318
Actual PM	.2432
Severity	.0013

Table 3: The parameters in the Cox-regression with p-values

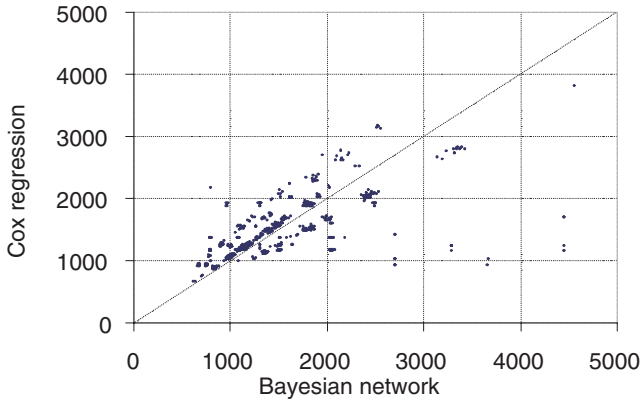

Figure 7: Scatterplot of the network predictions versus the Cox-regression predictions

that the two methodologies extract similar patterns from the data. The predictions from the Bayesian network and the regression model are correlated with $\rho = .62$.

To check if one method is better than the other, we note that the number of data points where the predictions from the Bayesian network are better than the regression predictions is (under the null hypothesis that both methods are equally good) drawn from the binomial distribution with n=883 and $p = 1/2$. When n is this large, we can approximate this by a Normal distribution, and fit the parameters $\mu = np = 441.5$ and $\sigma^2 = np(1-p) = 14.9^2$. The observed value is $x = 515$, i.e., the network predictions are better with p-value 4E-7. The standardized value for the logarithm of the prediction error was 1.88 for the Bayesian network and 1.89 for the Cox regression. Hence, the predictions from the Bayesian network are usually the closest, but the methods are at the same level in prediction quality. Note, however, that when the Cox-regression model is made, it is with prediction in mind. The Bayesian network was generated to fit the complete density functions. Methods for selecting the network best suited for prediction are also available (Spiegelhalter et al. 1993).

5 CONCLUSIONS

In this paper we have used a Bayesian network to represent the OREDA phase IV Gas Turbine dataset. The generated probabilistic graphical network helped analyzing the data, both qualitatively and quantitatively. The quantitative results were compared to those generated by a Cox regression model. The prediction results were fairly good, although neither the network nor the regression model were able to predict future survival

times with very high accuracy.

Predictive power is limited by four considerations (Korn and Simon 1990):

1. Inadequate models
2. Sampling variability when fitting a (correctly specified) class of models
3. Lack of explanatory power of the correctly fitted model
4. Problems of extrapolating to new data

In our case, the biggest contributor to the lack of predicting power for our model is variability in the exponential distribution, i.e., the explanatory power of the correctly fitted model is to low. To exemplify this, we have created an "optimal predictor". We call it "optimal" because it draws from an exponential distribution with parameter $\widehat{\lambda} = 1/t_{Observed}$ for each new observation $t_{Observed}$. Hence, we have removed the variability from item 1 and 2 above (assuming the exponential distribution is correct). The standardized value of the logarithmic prediction error was 1.31 for this predictor, compared to 1.88 for the Bayesian network and 1.89 for the Cox regression. Hence, 70% of the prediction error comes from the variability in the dataset, only 30% of the error was induced by poor estimates of the prediction density.

The generated Bayesian network has been useful for qualitative observations, and it gives a deeper understanding of the dataset which is useful in further analysis. We verified the quality of the extracted patterns by letting the Bayesian network predict future survival times. The obtained results were on a quantitative level similar to that of the Cox-regression model.

REFERENCES

Cooper, G. F. (1987). Probabilistic inference using belief networks is NP-hard. Technical Report KSL 87-27, Medical Computer Science Group, Stanford University.

Cox, D. R. (1972). Regression models and life-tables (with discussion). *Journal of the Royal Statistical Society, Series B 34*, 187–220.

Edwards, D. E. (1987). A guide to MIM. Technical Report 87/1, Statistical Research Unit, University of Copenhagen.

Geman, S. and D. Geman (1984). Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, 721–741.

Gilks, W. R., A. Thomas, and D. J. Spiegelhalter (1994). A language and program for complex bayesian modelling. *The Statisticien 43*, 169–178.

Heckerman, D., D. Geiger, and D. M. Chickering (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning 20*, 197–243.

Henderson, R. (1995). Problems and prediction in survival-data analysis. *Statistics in Medicine 14*, 161–184.

Hokstad, P. and A. T. Frøvig (1996). The modelling of degraded and critical failures for components with dormant failures. *Reliability Engineering and System Safety 51*(2), 189–199.

Horwitz, E. J., J. S. Breese, and M. Henrion (1988). Decision theory in expert systems and artificial intelligence. *Journal of Approximate Reasoning 2*, 247–302. Special issue on Uncertainty in Artificial Intelligence.

Jordan, M. I., Z. Gharamani, T. S. Jaakkola, and L. K. Saul (1998). An introduction to variational methods for graphical models. In M. I. Jordan (Ed.), *Learning in Graphical Models*. Kluwer Academic Publishers. To appear.

Korn, E. L. and R. Simon (1990). Measures of explained variation for survival data. *Statistics in Medicine 9*, 487–503.

Langseth, H., K. Haugen, and H. Sandtorv (1998). Analysis of OREDA data for maintenance optimisation. *Reliability Engineering and System Safety*. To appear.

Lauritzen, S. L. and D. J. Spiegelhalter (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society 50*, 157–224.

OREDA-97 (1997). *Offshore Reliability Data* (3 ed.). Distributed by Det Norske Veritas, P.O.Box 300, N–1322 Høvik. Prepared by SINTEF, N–7034 Trondheim, Norway.

Pearl, J. (1986). Fusion propagation and structuring in belief networks. *Artificial Intelligence 29*(3), 241–288.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. San Mateo: Morgan Kaufman.

Ramoni, M. and P. Sebastini (1996). Learning bayesian networks from incomplete databases. Technical Report KMi-TR-43, Knowledge Media Institute, The Open University.

Sandtorv, H., P. Hokstad, and D. W. Thompson (1996). Practical experiences with a data collection project: The OREDA project. *Reliability Engineering and System Safety 51* (2), 159–167.

Spiegelhalter, D. J., A. Phillip, S. L. Lauritzen, and R. G. Cowell (1993). Bayesian analysis in expert systems. *Statistical Science 8*, 219–282.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate statistics*. Chichester: John Wiley & Sons.