Gustavo's life and the speed needed to calculate it: a study into MAP and MPE queries

No Author Given

Faculty of Science, Vrije Universiteit, Amsterdam, The Netherlands

Abstract. This study conducts systematic research in exploring techniques for building Bayesian networks that model real-world situations. Specifically, it treats exact inference algorithms known as variable elimination at both theoretical and practical levels, using Most A Posteriori Estimate (MAP) and Most Probable Explanation (MPE) queries. Three heuristics are employed to investigate their effectiveness: min-fill, min-degree and random ordering. Furthermore, a modelled use case is tested on our Bayesian network reasoner to answer real-life questions.
The findings show that applying different heuristics do not yield statistically significant differences. Possible reasons behind this, including sample size and algorithm structure, are discussed. Furthermore, this study shows that Bayesian network reasoner can effectively generate answers to real-life questions and gives insights and reflections especially which are contrary to intuitive knowledge.

Keywords: Bayesian network reasoner; min-fill; min-degree

1 Introduction

A Bayesian network is a representative method to organize knowledge on a particular situation into a coherent and whole structure [1]. This type of network is built upon the directed acyclic graph (DAG), which is also known as the structure of the Bayesian network [2]. It captures two important parts of knowledge. Firstly, the variables represent the primitive propositions that we deem relevant to our domain; second, the edges convey information about the dependencies between these variables. These edges are often construed through the notion of causation in initial network construction, but not as a necessary posteriori explanation.

To fully and uniquely specify the network, a probability distribution for each variable needs to be further provided. This is known as the parameterization of the network, which is a localized process conditioned on every state of its direct parents as a quantified evaluation of the relationships.

One of the key representational aspects of Bayesian networks is known to be how these networks never require a quantitative specification between two variables that are not directly connected [3]. Since the Bayesian network is guaranteed to imply a unique value for every proposition, as long as that proposition can be expressed by the network variables, probabilities that quantify the relationship between a variable and its indirect causes or effects will be computed automatically by inference algorithms.

Exact algorithms are one of the two main paradigms of inference algorithms and are guaranteed to return correct answers. However, they tend to be more demanding computationally, as opposed to approximate inference algorithms, which are designed for automatically constructed Bayesian networks.

One main fundamental focus in the exact algorithms field is variable elimination [4]. Variable elimination is a process of removing variables iteratively from the network while still maintaining the ability to answer the queries of interest. However, there is still the problem of complexity since the two basic operations involved, marginalization and multiplication, take exponential time to calculate proportionate to the number of network variables. Hence, the performance of variable elimination will critically depend on the order in which variables are eliminated. Some orders are better than others in that they lead to constructing smaller intermediate factors. While it has been shown that computing an optimal order is an NP-hard problem, there are a number of heuristic approaches that tend to generate relatively good orders [5].

A number of greedy elimination heuristics have been proposed, suggesting ways to eliminate nodes from a graph G based on local considerations. The following are the two of the more common heuristics. *Min-degree*: eliminate the node having the smallest number of neighbours. *Min-fill*: eliminate the node that leads to adding the smallest number of fill-in edges.

1.1 Research question

In earlier research, the min-fill is suggested to outperform min-degree heuristic and be optimal for graphs whose treewidth is ≤ 2 [6]. However, to the researchers' knowledge, there is not much empirical evidence for this and direct comparisons are lacking. In this light, this study aims to further explore the effectiveness of different pruning heuristics using our Bayesian Network Reasoner, and additionally take tests on a fictitious use case of student life.

To study the effectiveness of various heuristics, Most A Posteriori Estimate (MAP) and Most Probable Explanation (MPE) Queries are used. Hence the research question is:

What is the effect of min-fill and min-degree heuristics on the performance speed of MAP and MPE queries in a Bayesian network reasoner compared to a random order heuristic?

1.2 Hypothesis

To answer the research question, a hypothesis is formulated. Based on previous research, we hypothesize the min-fill heuristic to outperform the random and min-degree heuristics and thus be the fastest to calculate. Additionally, it is hypothesized that the min-degree heuristic will outperform the random heuristic, because the use of certain heuristic is most likely better than nothing. Additionally, we expect MPE queries to be calculated faster than MAP queries in general, because MPE queries require a less specific answer. Finally, we expect the time needed to calculate queries to go up exponentially with the size of the network.

2 Materials & Methods

2.1 Bayesian Network reasoner

Our Bayesian Network reasoner contains the following main functions: d-separation, network pruning, ordering, marginal distributions and MAP and MPE functions. Brief summaries for each function are provided below.

D-separation The d-separation function allows the reasoner to mechanically and efficiently derive (in-)dependencies among variables. Additionally, convergent, divergent and sequential valves are taken into consideration.

Network pruning To answer certain queries, a limited set of nodes and edges may be needed. The network pruning function can significantly reduce redundant structure to decrease network complexity by removing nodes not relevant for the query.

Ordering Three different ordering functions are built for the three competing heuristics, namely a min-degree function, min-fill function, and a random-ordering function. The random-ordering function generates a random elimination order, while the min-degree function eliminates nodes in order of the smallest number of neighbors and the min-fill function eliminates the nodes in order of smallest numbers of edges to be filled in, as defined in the Introduction.

Marginal Distribution The Marginal distribution function works to answer a priori marginal queries and posterior marginal queries within the network. These two types of queries both investigate the marginal distribution of some variables at interest. The only difference is that in posterior marginal queries, the evidence e is given, while in a priori marginal queries, it is not.

MAP & MPE MAP functions and MPE functions are used to generate answers to the corresponding queries. MAP queries and MPE queries are both used to identify the most probable instantiation of certain network variables given some evidence. While the MAP query searches for an instantiation $\mathbf{m}$ of variables $\mathbf{M}$, where $\mathbf{M}$ is a subset of the entire set of all network variables $\mathbf{X}$, the MPE query is the special case when $\mathbf{M}$ includes all network variables.

2.2 Experiment & Hypothesis testing

Based on the available literature, it was hypothesized that random ordering would be outperformed by min-degree ordering, while this would in turn be outperformed by min-fill ordering. To test this hypothesis, an experiment was conducted.

In this experiment, a random network of $N = 100$ nodes was created in steps of $n = 10$ randomly added nodes. Each node was randomly connected to one other node already in the network, and the conditional probability tables for this new child node were updated. After each step of 10 randomly added nodes, random MAP and MPE queries were generated and answered for each heuristic. The computation time needed to answer these queries was measured and saved. This experiment was repeated 10 times to improve statistical reliability of the results.

Based on our hypothesis, we expect the computing time for the min-fill queries to be the fastest, followed by the min-degree and random queries. We also expect the MPE queries to be calculated faster than the MAP queries.

2.3 Use case application

In this section, we discuss a number of reasoning problems that arise in real-world applications and show how each can be addressed by first modeling the problem using a Bayesian network. Then, some queries defined in the previous section are posed.

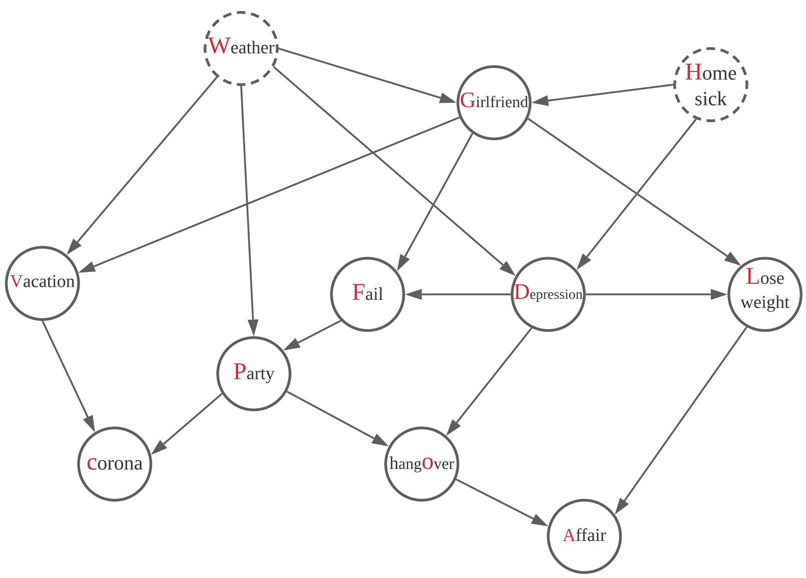

Fig. 1. use case

Use case scenario This use case represents a student scenario, in the aim of exploiting our Bayesian Network reasoner to not just theoretically deepen our intuition towards the mechanism, but also hopefully get some constructive insights for our own lives. Thus we try to depict a persona:

> *Gustavo, caucasian male, age 24, year one AI master student at Vrije University Amsterdam, who comes from south Europe. He is somewhat bothered by the gloomy weather(W) during the winter season while missing home sun(H) (and everything related to it) a lot.*
> *There is a chance that Gustavo is facing exam failure(F) due to either depression(D), or by having a girlfriend(G), which is time consuming. When he is feeling down, Gustavo might want to cheer himself up by going to the parties(P), but who knows life could strike him a little harder still by giving him Corona(C).*
> *Meanwhile, having a girlfriend might increase the probability of him planning a Christmas vacation abroad(V), which could also increase exposure to Corona. Besides, Gustavo could be losing weight(L) either from trying to stay attractive in the relationship, or simply through being too depressed to eat. Our guy is basically a faithful young man, but being fit and charming and occasionally drunk(O), there are of course stupid times when Gustavo could have an affair(A).*

Modelling The Knowledge representation (KR) approach is used to construct the Bayesian networks for our query task, which is largely a subjective and intuitive capturing of knowledge. We consider this approach fit for a real-life modeling use case, modelled by 3 steps:

First, we define the network variables and their values. We partition network variables into three types, root nodes (evidence nodes), leaf nodes (query nodes), and intermediary nodes.

Second, we define the network structure (edges). In our case, we are guided by a causal interpretation of network structure. Hence, the determination of this structure is reduced to answering the following question about each network variable X: What is the set of variables that we regard as the direct causes/effects of X? Specifically, we distinguish three types of nodes: root nodes, that only have incoming edges, since these are evidence given in this case; leaf nodes, that only have outgoing edges, since these are results we are interested in; and intermediary nodes, which have both incoming and outgoing edges. Note that we still ask questions about intermediary nodes, even if they are not directly in our results.

Lastly, we define the network CPTs. In this case, the CPTs are determined completely from the problem statement by both objective considerations and reflection of subjective beliefs. While rationals are provided, value choices are made arbitrarily.

There are 11 variables altogether in the use case modeling (see fig. 1), including 2 root nodes, 2 leaf nodes, 7 intermediary nodes. All nodes except for root nodes have 2 incoming edges. It is best to think of these edges as signifying direct causal influences. Justification of variable paths and CPTs to be found in the appendix.

Inquiries In the spirit of exploiting the Bayesian Network reasoner to yield further insights from a practical angle, we raise the following queries, we cover all four types of queries defined in the previous section.

Priori marginal queries

1. What is Gustavo's chance of having an affair?
2. What is Gustavo's chance of catching corona?
3. What is Gustavo's chance of exam failure?

Posteriori marginal queries

1. Given: weather = *False*, homesick = *True*, what is Gustavo's chance of being happy and sound: fail = *False*, depression = *False*, girlfriend = *True*, party = *True*, vacation = *True*, loseweight = *True*, corona = *False*
2. Given: weather = *False*, homesick = *True*, what is Gustavo's chance of being down at rock bottom: fail = *True*, depression = *True*, corona = *True*, girlfriend = *False*, loseweight = *False*

MAP queries

1. Given: girlfriend = *True*, fail = *True*, is Gustavo likely to catch corona(C)?
2. Given: weather = *True*, loseweight = *True*, is Gustavo likely to catch a fling(A)?

MPE queries

1. Given: weather = *False*, homesick = *True*, what kind of situation is Gustavo most likely to be in:
   * failing exam(F)?
   * having a lovelife(G)?
   * or just a fling (A)?
   * socially active?(partying)(P)?
   * drinking too much(O)?
   * constantly in a mood (D)?
   * still healthy(C)?
   * still fit?(L)?
   * taking some time to enjoy himself(V)?

3 Results

To answer our research question and test our hypothesis, we conducted an experiment as described in Section 2.2. The results of this experiment are discussed below. Additionally, this section covers the results from task 3, which contained a set of queries to test our functions on a use-case.

3.1 Experimental results

After the results were generated, they were plotted in figure 2. In this figure, it seems that the average running time increases as the network size goes up. It also seems that the MEP min-fill, MAP min-degree and MAP min-fill queries perform worse on average than the MAP random, MEP random and MEP min-degree queries. This is contradictory to our hypothesis, so statistical analysis is needed.

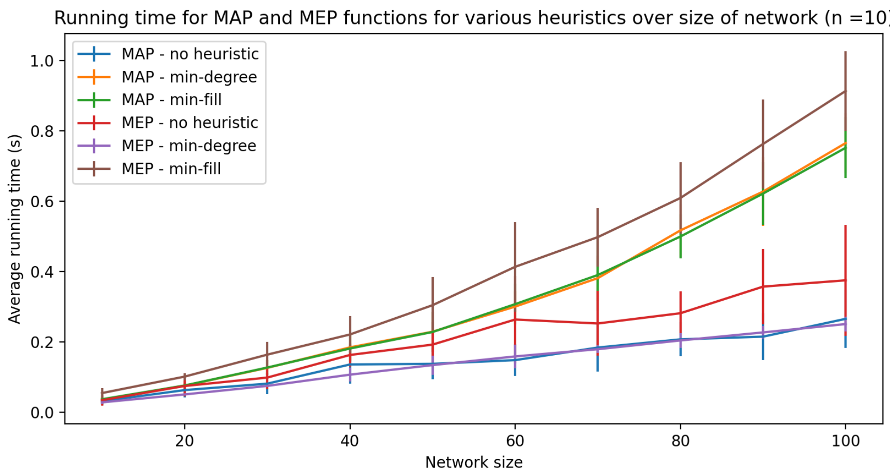

Fig. 2. Results experiment 2. On the x-axis, the size of the network is given. On the y-axis, the average running time over 10 runs is shown with error. In this figure, it seems that the MEP min-fill, MAP min-degree and MAP min-fill queries perform worse than the other queries, especially as the size of the network gets bigger.

3.2 Statistical analysis

To analyze these results, the groups were first tested for normality using the Shapiro-Wilk test. Then, a Kruskal-Wallis test was performed to compare the means of the various groups. Lastly, a post-hoc Dunn test was performed to look at the differences between groups. For all tests, a value of $\alpha = 0.05$ was used.

Shapiro-Wilk To test for the normality of distribution, a Shapiro-Wilk test was performed. The results of the test are shown in table 1 below.

From this table, we can see that the groups are not normally distributed. For this reason, a Kruskal-Wallis non-parametric ANOVA was used for testing.

Kruskal-Wallis and Post-hoc The Kruskal-Wallis test gave a result of $H = 10.26, p = 0.06813$. This result is not significant. This means that we cannot reject the null hypothesis that the samples came from the same distribution. However, a post-hoc test was still performed to see the differences between groups.

Group	Test statistic	p-value
MAP - no heuristic	0.949	0.000705
MAP - min-degree	0.920	$1.49 * 10^{-5}$
MAP - min-fill	0.917	$9.89 * 10^{-6}$
MPE - no heuristic	0.930	$5.30 * 10^{-5}$
MPE - min-degree	0.923	$2.035 * 10^{-5}$
MPE - min-fill	0.955	0.00168

Table 1. Results of Shapiro-Wilk test. All groups are significant, meaning that we do not reject the null hypothesis that these groups are not normally distributed.

From table 2, we can see that there is only one significant result: between the MAP random and the MPE min-fill heuristics. However, since the results of the Kruskal-Wallis test are not significant, we cannot interpret this result. It is still interesting to note from this table, however, that the MAP heuristics perform quite differently from each other, while the MPE heuristics perform much more similarly.

Group	MAP- no heuristic	MAP- min-degree	MAP- min-fill	MPE- no heuristic	MPE- min-degree	MPE- min-fill
MAP- no heuristic	1	0.0794	0.0816	0.343	0.848	**0.0227**
MAP- min-degree	0.0794	1	0.990	0.420	0.0516	0.560
MAP- min-fill	0.0816	0.990	1	0.427	0.0532	0.591
MPE- no heuristic	0.343	0.420	0.427	1	0.254	0.183
MPE- min-degree	0.848	0.0516	0.0532	0.2544	1	**0.0135**
MPE- min-fill	0.0227	0.560	0.591	0.1829	0.0135	1

Table 2. Results of the Dunn post-hoc test. Significant results are shown in bold.

3.3 Use-Case Query results

In task 3, a number of queries were answered.

For a priori marginal queries, we answered (1) Gustavo's chance of having an affair? *True* = 0.44, *False* = 0.56; (2) Gustavo's chance of catching corona? *True* = 0.42, *False* = 0.58; (3) Gustavo's chance of exam failure? *True* = 0.53, *False* = 0.47.

For a posteriori marginal queries, the answers to (1) Given weather = *False*, homesick = *True*, Gustavo's chance of being happy and sound: $p = 0.00017$, which is noticably small. All query variables form 128 value combinations in total, and this particular combined probability comes in the 100th. (2) Given:

weather = *False*, homesick = *true*, what is Gustavo's chance of being down at rock bottom: $p = 0.004$. Of 32 value combinations of all variables in focus, this combination probability comes in the 21st.

For MAP queries, (1) Given: girlfriend = *True*, fail = *True*, Gustavo is suggested unlikely to catch Corona; (2) Given: weather = *False*, loseweight = *True*, Gustavo is suggested to be unlikely to catch a fling.

For MPE queries, Given weather = *False*, homesick = *True*, Gustavo is suggested to not be failing exams (Fail = *False*); won't have a lovelife (Girlfriend = *False*); won't have friends with benefits either (Affair = *False*); is not partying (Party = *False*); correspondingly not overdoing drinking (Hangover = *False*); not very moody (Depression = *False*); still healthy (Corona = *False*); not losing weight though (Loseweight = *False*); and not taking vacation apparently (Vacation = *False*).

4 Conclusion and Discussion

4.1 Experimental conclusion and discussion

Experimentation for task 2 yields insignificant results, as was shown in the previous section, which means that we cannot accept our hypothesis, and instead come to an conclusion that different MAP and MPE heuristics do not suggest to perform differently.

While this is not the results we had expected, this conclusion is interesting in and of itself. This data being somewhat inconclusive means that more experimentation is necessary to find the effectiveness of these heuristics. Perhaps the reason that data is sparse on this research topic is because results often come back inconclusive, making it difficult to draw clear conclusions about the workings of these heuristics.

Another explanation for these results might be that our experiment size was too small. With only 10 runs, the variation in data was large and this will diminish the power of the statistical tests performed. More data means more reliable averages and thus the results could be seen more clearly.

Finally, a reason could be in the algorithm itself. The functions to sum-out, maximize-out and order were written by hand, meaning that these heuristically ordering functions could turn out to be more complex than the random ordering method, which resulted in elongated computational time.

Future research Since our results were still not quite conclusive, further research should focus on gathering more information on why and how these heuristics take effect. Literature on direct comparison of heuristics performances is still sparse at the moment. One way of doing this could be to rerun the experiments conducted in this study with a larger sample size, or by employing different variables, such as including the notion of treewidth.

4.2 Use-Case Query conclusions

For the use-case, we purposefully modelled a case reflecting the life of our own peers and yield some interesting results that might shine a light or two on how students could live life.

For our fictitious hero Gustavo being under a lousy weather and being far away from home, it is suggested that his chances of either having it all or losing it all with regard to study, love, health, or life is very small. He is as unlikely to suck at everything as to win with everything.

While this aligns with our common senses, one interesting finding to note though, with other variables basically distribute more or less balanced, it just seems that given a bad weather and being lonely, there is a greater chance that one would not easily get depressed, yet more unlikely to have a relationship either. This turns out to be a fun contradiction to our initial assumptions, where we assumed that being in bad weather would increase one's depression level while being alone should increase one's inclination for intimacy.

References

1. Judea Pearl. Bayesian netwcrks: A model cf self-activated memory for evidential reasoning. In *Proceedings of the 7th conference of the Cognitive Science Society, University of California, Irvine, CA, USA*, pages 15–17, 1985.
2. David Edwards. *Introduction to graphical modelling.* Springer Science & Business Media, 2012.
3. Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19, 2000.
4. Nevin L Zhang and David Poole. A simple approach to bayesian network computations. In *Proc. of the Tenth Canadian Conference on Artificial Intelligence*, 1994.
5. Chris Harbron. Heuristic algorithms for finding inexpensive elimination schemes. *Statistics and Computing*, 5(4):275–287, 1995.
6. Adnan Darwiche. *Modeling and reasoning with Bayesian networks.* Cambridge university press, 2009.

Appendix CPTs

Scenario : bad weather		*Scenario : being homesick*	
Weather (w)	Θw	Homesick(H)	ΘH
T	0.2	T	0.6
F	0.8	F	0.4

Assumption: bad weather and homesick(feeling lonely) increase one's inclination for intimacy
Assumption: weather has a bigger impact than nostalgia

Weather	Homesick	Girlfriend(G)	ΘG|W, H	
T	T	T	0.6	*good weather, lonely (++)*
T	T	F	0.4	
T	F	T	0.2	*good weather, not lonely (+)*
T	F	F	0.8	
F	T	T	0.9	*bad weather, lonely (++++)*
F	T	F	0.1	
F	F	T	0.7	*bad weather , not lonely (+++)*
F	F	F	0.3	

Assumption: bad weather and exam failure increase one's decision for ligtening the mood: go to parties
Assumption: weather has a bigger impact than exam failure

Weather	Fail	Party(P)	ΘP|W, F	
T	T	T	0.6	*good weather, fail (++)*
T	T	F	0.4	
T	F	T	0.3	*good weather, not fail (+)*
T	F	F	0.7	
F	T	T	0.9	*bad weather, fail (++++)*
F	T	F	0.1	
F	F	T	0.7	*bad weather, not fail (+++)*
F	F	F	0.3	

Assumption: party going and vacation abroad increase chance of Corona infection
Assumption: party going has a bigger impact than vacation abroad

Party	Vacation	Corona	ΘC|P, V	
T	T	T	0.9	*party, vacation (++++)*
T	T	F	0.1	
T	F	T	0.7	*party, no vacation(+++)*
T	F	F	0.3	
F	T	T	0.6	*no party, vacation(++)*
F	T	F	0.4	
F	F	T	0.1	*no party, no vacation(+)*
F	F	F	0.9	

Assumption: being in a relationship and being depressed can both lead one to lose weight
Assumption: relationship and depression has an equal effect

Girlfriend	Depression	Loseweight	ΘL|G, D	
T	T	T	0.8	*have gf, depressed (++++)*
T	T	F	0.2	
T	F	T	0.6	*have gf, not depressed (++)*
T	F	F	0.4	
F	T	T	0.6	*no gf, depressed (++)*
F	T	F	0.4	
F	F	T	0.2	*no gf, not depressed (+)*
F	F	F	0.8	

Assumption: being fit and being drunk can both increase the chance of having an affair
Assumption: being fit has a bigger impact than being drunk

Loseweight	Hangover	Affair	ΘA|L, O	
T	T	T	0.8	*fit, drunk (++++)*
T	T	F	0.2	
T	F	T	0.7	*fit, not drunk(+++)*
T	F	F	0.3	
F	T	T	0.6	*not fit, drunk(++)*
F	T	F	0.4	
F	F	T	0.1	*not fit, not drunk (+)*
F	F	F	0.9	

Assumption: relationship (distraction & time consuming) and depression both increase chances of exam failure
Assumption: depression has a bigger impact than relationship

Girlfriend	Depression	Fail(F)	ΘF|G, D	
T	T	T	0.8	*have gf, depressed (++++)*
T	T	F	0.2	
T	F	T	0.6	*have gf, not depressed (++)*
T	F	F	0.4	
F	T	T	0.7	*no gf, depressed (+++)*
F	T	F	0.3	
F	F	T	0.1	*no gf, not depressed (+)*
F	F	F	0.9	

Assumption: bad weather and be in relationship increase one's decision to go on vacation abroad
Assumption: weather and relationship has an equal impact

Weather	Girlfriend	Vacation(V)	ΘV|W, G	
T	T	T	0.6	*good weather, have gf(++)*
T	T	F	0.4	
T	F	T	0.2	*good weather, no gf(+)*
T	F	F	0.8	
F	T	T	0.8	*bad weather, have gf (++++)*
F	T	F	0.2	
F	F	T	0.6	*bad weather, no gf (++)*
F	F	F	0.4	

Assumption: party going and depression can increase the chance of hangover
Assumption: party going and depression has an equal effect

Party	Depression	Hangover	ΘO|P, D	
T	T	T	0.9	*party, depressed (++++)*
T	T	F	0.1	
T	F	T	0.6	*party, not depressed (++)*
T	F	F	0.4	
F	T	T	0.6	*not party, depressed (++)*
F	T	F	0.4	
F	F	T	0.2	*not party, not depressed (+)*
F	F	F	0.8	