The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Approximate MMAP by Marginal Search

Alessandro Antonucci
IDSIA
Lugano (Switzerland)
alessandro@idsia.ch

Thomas Tiotto
Groningen Cognitive Systems and Materials
Groningen (The Netherlands)
t.f.tiotto@rug.nl

Abstract

We present a heuristic strategy for marginal MAP (MMAP) queries in graphical models. The algorithm is based on a reduction of the task to a polynomial number of marginal inference computations. Given an input evidence, the marginals mass functions of the variables to be explained are computed. Marginal information gain is used to decide the variables to be explained first, and their most probable marginal states are consequently moved to the evidence. The sequential iteration of this procedure leads to a MMAP explanation and the minimum information gain obtained during the process can be regarded as a confidence measure for the explanation. Preliminary experiments show that the proposed confidence measure is properly detecting instances for which the algorithm is accurate and, for sufficiently high confidence levels, the algorithm gives the exact solution or an approximation whose Hamming distance from the exact one is small.

Introduction

Probabilistic graphical models such as Bayesian networks and Markov random fields are popular tools for a compact generative description of the uncertain relations between the variables in a system (Koller and Friedman 2009). Reasoning with such models is achieved by inferential computations involving sums and maximizations among the local components (potentials or conditional probability tables).

Typical inference tasks in these models can be regarded as special cases of a general task called *marginal MAP* (MMAP). In a MMAP task a set of model variables should be *explained*, i.e., their joint most probable state should be detected, while some of the other variables are *observed* in a given state, and the remaining ones should be *marginalized*, i.e., summed out. Complexity analysis reveals that MMAP is a $\mathrm{NP}^{\mathrm{PP}}$-complete (Park and Darwiche 2004). Notable MMAP sub-cases correspond to situations in which: (i) there are no variables to explain and the problem corresponds to the computation of the probability of the observed variables; and (ii) there are no variables to marginalize and the problem is to find the most probable state of the variables to explain given an observation of all the other variables. The complexity of these two tasks, sometimes called, respectively, PR and MAP inference, is lower as PR is #P-complete and MAP is NP-complete. In practice MMAP is a much harder task than PR or MAP and, for instance, for singly-connected topologies polynomial solutions of PR and MAP can be derived while MMAP remains NP-hard (Koller and Friedman 2009). Despite such high complexity, as noticed in (Marinescu, Dechter, and Ihler 2018), MMAP is a very important task, as it corresponds to the case of a model with latent variables, which are commonly used in graphical models to express non-trivial dependency patterns. Various *anytime* algorithms providing lower and upper bounds to the optimal MMAP values have been proposed, e.g., (Mauá and de Campos 2012), and the state of the art is currently the bounding scheme based on stochastic search proposed on (Marinescu, Dechter, and Ihler 2018).

Marginal inference (MAR) is a third important MMAP sub-case for which only a single variable is explained. It is straightforward to reduce MAR to a number of calls to PR equal to the number of states of the variable to explain and its complexity remains therefore #P-complete. In this paper we reduce MMAP to a polynomial number of MAR calls. Given the evidence of the MMAP task, our procedure uses MAR to compute the marginal mass functions of the variables to explain and "move" to the observation the variables with the most extreme probabilistic values. The iteration of this procedure represents a heuristic approach to approximate MMAP. Different information-theoretic criteria can be considered to drive such a search for the most probable configuration in order to define scores to characterize the reliability of the corresponding explanation. These scores can be also used to decide when the procedure should be terminated, thus providing a partial-but-reliable MMAP explanation. The paper is organized as follows: we first review the existing work in the field and formalize the problem with the necessary notation, the heuristic strategy together and the scores are consequently described, and an empirical validation reported together with a discussion about relations with existing work and possible outlooks is finally provided.

Related Work

In (Butz, Hommersom, and van Eekelen 2018), a procedure similar to the one presented in this paper has been considered in the context of explainable AI and Bayesian networks. Rather than focusing on the algorithmic task, the goal of that procedure is to generate a linguistic explanation of the input evidence and a description of the reasoning process behind the model. Here we make explicit how such scheme would not necessarily return exact explanations and provide an information-theoretic score able to characterize their confidence level, this being proved to be more effective than the highest probability level considered in that paper.

Concerning the MMAP literature, most of the existing algorithms are exact schemes possibly giving anytime approximations (Park and Darwiche 2002; Mauá and de Campos 2012; Marinescu, Dechter, and Ihler 2014; 2018). Variational methods reducing the task to message propagation have been proposed instead for approximate inference (Jiang, Rai, and Daume 2011; Liu and Ihler 2013).

Background

We consider discrete random variables only. If X is a variable, the finite set Ω_X denotes its set of possible values, $|\Omega_X|$ is the cardinality of this set, and x is a generic element of Ω_X. A probability mass function P is a nonnegative real-valued map defined over Ω_X and normalized to one. For each $x \in \Omega_X$, $P(x)$ is the probability for $X = x$. The entropy of a mass function P over X is defined as $H[P(X)] := -\sum_{x\in\Omega_X} P(x)\log_{|\Omega_X|} P(x)$. Note that $H[P(X)] \in [0,1]$, being one with uniform mass functions and zero in the degenerate case of probabilities equal to zero and one. Given a joint variable $\boldsymbol{X} := (X_1, \ldots, X_n)$, we can similarly define a joint mass function $P(\boldsymbol{X})$. Given $\boldsymbol{x} \in \Omega_{\boldsymbol{X}}$ and $\boldsymbol{X}' \subset \boldsymbol{X}$, notation $\boldsymbol{x}^{\downarrow \boldsymbol{X}'}$ is used for the restriction to the variables in $\boldsymbol{X}'$ of the states in $\boldsymbol{x}$. A potential ϕ is just an un-normalized (but still non-negative) mass function. Say that for each $i = 1, \ldots, f$, ϕ_i is a potential over $\boldsymbol{X}_i \subset \boldsymbol{X}$ and such that $\cup_{i=1}^f \boldsymbol{X}_i = \boldsymbol{X}$. If this is the case we call $\boldsymbol{\Phi} := \{\phi_i\}_{i=1}^f$ a graphical model (GM) over $\boldsymbol{X}$. A GM defines a joint mass function over $\boldsymbol{X}$ such that $P(\boldsymbol{X}) \propto \prod_{i=1}^f \phi_i(\boldsymbol{X}_i)$. Note that both Bayesian networks and Markov random fields can be regarded as GMs. We are now in the condition of defining MMAP inference in GMs.

Definition 1 (MMAP) *Given a GM $\boldsymbol{\Phi}$ over $\boldsymbol{X}$, the partition $(\boldsymbol{X}_M, \boldsymbol{X}_S, \boldsymbol{X}_E)$ of $\boldsymbol{X}$, and an* observation $\boldsymbol{X}_E = \boldsymbol{x}_E$*, a MMAP task consists in the computation of state:*

$$\boldsymbol{x}_M^* := \operatorname*{arg\,max}_{\boldsymbol{x}_M \in \Omega_{\boldsymbol{X}_M}} \sum_{\boldsymbol{x}_S \in \Omega_{\boldsymbol{X}_S}} P(\boldsymbol{x}_M, \boldsymbol{x}_S, \boldsymbol{x}_E)\,, \qquad (1)$$

and the corresponding probability $\mathrm{p}^* := P(\boldsymbol{x}_M^*, \boldsymbol{x}_E)$.

If $\boldsymbol{X}_M = \emptyset$, MMAP is called PR and it only consists in the computation of $P(\boldsymbol{x}_E)$. Although both problems are NP-hard in general, PR is considerably simpler task (being #P-complete) compared to general (NP$^{\mathrm{PP}}$-complete) MMAP. Let us also define the MAR task.

Definition 2 (MAR) *Given a GM $\boldsymbol{\Phi}$ over $\boldsymbol{X}$, an observation $\boldsymbol{x}_E$ of the variables $\boldsymbol{X}_E \subset \boldsymbol{X}$, and a single variable $X \in \boldsymbol{X} \setminus \boldsymbol{X}_E$, a MAR task consists in the computation of $P(x|\boldsymbol{x}_E)$ for each $x \in \Omega_X$.*

It is straightforward to solve MAR by using PR to compute $P(x, \boldsymbol{x}_E)$ for each $x \in \Omega_X$, as the normalization of these joint probabilities gives the MAR conditional probabilities.

Approximating MMAP by Multiple MARs

As noticed in the previous section MMAP becomes simpler if $\boldsymbol{X}_M$ contains a single variable. Yet, as shown by the following example from (Liu and Ihler 2013), MMAP cannot be trivially reduced to a sequence of MAR tasks.

Example 1 (Wheather Dilemma) *Variable R and D denote, respectively, whether or not it is a rainy day in Irvine, and whether or not Alice is going to the office by car. Accordingly let us assume $\Omega_D := \{rainy, sunny\}$ and $\Omega_R := \{walk, drive\}$. The assessments for the marginal probability $P(rainy) = 0.4$ and the conditional probabilities $P(drive|rainy) = .875$ and $P(drive|sunny) = 0.5$ are sufficient to compute the joint mass function $P(R, D)$ displayed in Table 1. State $sunny$ is the one with the highest marginal probability for R and, similarly, $drive$ has the highest marginal for D. Yet, the most probable joint state of (R, D) is $(rainy, drive)$ (see bold numbers in Table 1).*

r	d	$P(r,d)$	$P(r)$	$P(d)$
$sunny$	$walk$	0.30	**0.60**	0.35
$rainy$	$walk$	0.05	0.40	-
$sunny$	$drive$	0.30	-	**0.65**
$rainy$	$drive$	**0.35**	-	-

Table 1: Joint and marginal probabilities for Example 1

The above example shows that a most probable joint configuration is not necessarily a combination of most probable marginal configurations. This is perfectly acceptable for non-independent variables. If we regard the identification of the most probable joint state as a MMAP task and the identification of the two most probable marginal states as a MAR task, Example 1 shows that MMAP cannot be trivially reduced to a sequence of MAR tasks over the variables to explain. Yet, in the following example we show that a more sophisticated scheme could be more effective in achieving such reduction.

Example 2 (Solving the Wheather Dilemma) *Consider the same setup as in Example 1. As shown in Table 1, the marginal mass functions of the two variables are $P(R) = [.6, .4]$ and $P(D) = [.35, .65]$. Among these two mass functions, $P(D)$ is the one with the most extreme value, i.e., $drive$. We regard such an "extreme" state as an evidence and, consequently, compute $P(R|drive) = [6, 7]/13$. Thus, the most probable (conditional) state of R is $rainy$. In other words, the most probable joint state of the two variables is the combination of the most probable marginal state for the variable with the most extreme marginal probability combined the most probable posterior state for the other variable after promoting the first to an evidence.*

Example 2 suggests a heuristic strategy for MMAP tasks. In that example, the most probable configuration of the two variables is obtained sequentially by first explaining a variable, whose most probable state is promoted to evidence, and finally explaining the other. Note that starting from the variable with the most extreme values might be important, e.g., explaining R before D leads to a wrong conclusion.

The most extreme probabilistic value in the example can be intended as a proxy for the most informative (i.e., least entropic) mass function. The difference between the two descriptors might be important only when comparing mass functions over variables with different cardinality. Consider for instance the ternary mass function $P(X') := [.2, .1, .7]$ and the binary mass function $P(X'') := [.8, .2]$, while $\max_{X' \in \Omega_{X'}} P(X') > \max_{X'' \in \Omega_{X''}} P(X'')$ we have $H[P(X')] < H[P(X'')]$. When comparing binary variables the two descriptors are instead equivalent, as the entropy is a monotone function of the probability of the most probable state. With more than two states, again having the highest most probable state might not imply that the entropy is lower, e.g., $H[[0.75, 0.24, 0.01]] < H[[0.80, 0.10, 0.10]]$.

We are now in the condition of presenting our heuristic reduction of MMAP to MAR. This is detailed in Algorithm 1. Given a MMAP instance in input (line 1), a copy of the variables to be explained, the observed ones and their states are stored (line 2). The procedure consists in computing the MAR for each variable to be explained (lines 4-6), find the least entropic one (line 7), and its most probable state (line 8). Such a variable-state pair is moved to the evidence (lines 9-11). The procedure is iterated until all the variables to be explained are moved to the evidence (line 3). The resulting explanation is the restriction to the variables to be explained of the evidence generated in this way (line 13).

Algorithm 1 MMAP2MAR

```
 1: input: $(\boldsymbol{X}_M, \boldsymbol{X}_E, \boldsymbol{x}_E)$
 2: $(\boldsymbol{X'_M}, \boldsymbol{X'_E}, \boldsymbol{x'_E}) \leftarrow (\boldsymbol{X}_M, \boldsymbol{X}_E, \boldsymbol{x}_E)$
 3: while $\boldsymbol{X}'_M \neq \emptyset$ do
 4:     for $X \in \boldsymbol{X}'_M$ do
 5:         compute $P(X|\boldsymbol{x}'_E)$
 6:     end for
 7:     $X^* \leftarrow \arg\min_{X \in \boldsymbol{X}'_M} H[P(X|\boldsymbol{x}'_E)]$
 8:     $\tilde{x}^* \leftarrow \arg\max_{x^* \in \Omega_{X^*}} P(x^*|\boldsymbol{x}'_E)$
 9:     $\boldsymbol{X}'_M \leftarrow \boldsymbol{X}'_M \setminus \{X^*\}$
10:     $\boldsymbol{X}'_E \leftarrow \boldsymbol{X}'_E \cup \{X^*\}$
11:     $\boldsymbol{x}'_E \leftarrow \boldsymbol{x}'_E \cup \{\tilde{x}^*\}$
12: end while
13: output: $\boldsymbol{x}^*_M \leftarrow \boldsymbol{x}_E'^{\downarrow \boldsymbol{X}_M}$
```

Overall, Algorithm 1 requires a polynomial number of MAR calls, this number being clearly quadratic with respect to the cardinality of $\boldsymbol{X}_M$. In order to understand the kind of approximation induced by Algorithm 1, let us consider just for the sake of readability a simpler (MAP) task with both $\boldsymbol{X}_E$ and $\boldsymbol{X}_S$ empty and three variables to be explained, i.e., $p^* := \max_{x_1, x_2, x_3} P(x_1, x_2, x_3)$. By considering the *chain rule* with the natural order over the three variables, we have:

$$p^* = \max_{x_1, x_2, x_3} P(x_3|x_2, x_1) P(x_2|x_1) P(x_1)\,, \qquad (2)$$

while, assuming that the order induced by the least entropic marginals is the natural one, Algorithm 1 returns:

$$\tilde{p} := \max_{x_3} P(x_3|\tilde{x}_2, \tilde{x}_1) \max_{x_2} P(x_2|\tilde{x}_1) \max_{x_1} P(x_1)\,, \qquad (3)$$

where $\tilde{x_1}$ is the $\arg\max_{x_1} P(x_1)$ and $\tilde{x_2}$ is $\arg\max_{x_2} P(x_2|\tilde{x}_1)$. If we also set $\tilde{x_3} := \arg\max_{x_3} P(x_3|\tilde{x_1}, \tilde{x_2})$, by chain rule and Equation 3 we have $\tilde{p} = P(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ and thus, by Equation (2), $p^* \geq \tilde{p}$. The result, which remains valid for general MMAP instances, says that in general Algorithm 1 gives a lower bound for MMAP tasks.

After any iteration of the while loop in Algorithm 1, $H[P(X^*|\boldsymbol{x}_E)]$ can be intended as a confidence measure of the heuristic action of moving $(X^* = \tilde{x}^*)$ to the evidence. Algorithm 2 depicts a more cautious version of Algorithm 1 that moves a most probable state to the evidence only if the minimum entropy of the marginal mass functions is below a threshold ϵ (line 8). If this is not the case, the iteration ends (line 14). After termination the values of the explained variables can be extracted from the evidence. Yet, unlike Algorithm 1, in this case only the variables of $\boldsymbol{X}_M$ not in $\boldsymbol{X}'_M$ are explained (line 17).

Algorithm 2 ϵ-MMAP2MAR

```
 1: input: $(\boldsymbol{X}_M, \boldsymbol{X}_E, \boldsymbol{x}_E, \epsilon)$
 2: $(\boldsymbol{X'_M}, \boldsymbol{X'_E}, \boldsymbol{x'_E}) \leftarrow (\boldsymbol{X}_M, \boldsymbol{X}_E, \boldsymbol{x}_E)$
 3: while $\boldsymbol{X}'_M \neq \emptyset$ do
 4:     for $X \in \boldsymbol{X}'_M$ do
 5:         compute $P(X|\boldsymbol{x}'_E)$
 6:     end for
 7:     $X^* \leftarrow \arg\min_{X \in \boldsymbol{X}'_M} H[P(X|\boldsymbol{x}'_E)]$
 8:     if $H[P(X^*|\boldsymbol{x}'_E)] < \epsilon$ then
 9:         $\tilde{x}^* \leftarrow \arg\max_{x^* \in \Omega_{X^*}} P(x^*|\boldsymbol{x}'_E)$
10:         $\boldsymbol{X}'_M \leftarrow \boldsymbol{X}'_M \setminus \{X^*\}$
11:         $\boldsymbol{X}'_E \leftarrow \boldsymbol{X}'_E \cup \{X^*\}$
12:         $\boldsymbol{x}'_E \leftarrow \boldsymbol{x}'_E \cup \{\tilde{x}^*\}$
13:     else
14:         break
15:     end if
16: end while
17: output: $\boldsymbol{x}^*_{M''} \leftarrow \boldsymbol{x}_E'^{\downarrow \boldsymbol{X}_M \setminus \boldsymbol{X}'_M}$
```

Numerical Experiments

For a first empirical validation of Algorithms 1 and 2, we consider a benchmark of seven publicly available Markov random fields with different characteristics. Details about these GMs are in Table 2, where n denotes the number of model variables, f the number of potentials, and ω the maximum cardinality of the variables. For each network we generate a random MMAP instance as follows: (i) we select k variables from $\boldsymbol{X}$; (ii) we generate a random observation of those variables. Given this evidence, we run Algorithm 2 with an entropy threshold level equal to ϵ and regard the variables explained by the algorithm after termination as $\boldsymbol{X}_M$.[1]

[1] Code available at https://github.com/Tioz90/MMAP2MAR.

For both MAR and MMAP queries, we use the state-of-the-art exact solver Merlin built on top of And/Or search as developed in (Marinescu, Dechter, and Ihler 2014) and later extended with stochastic search in (Marinescu, Dechter, and Ihler 2018).[2]

Id	Filename	n	f	ω
(a)	`GEOM30a_3.wcsp.uai`	30	81	3
(b)	`GEOM30a_4.wcsp.uai`	30	81	4
(c)	`rbm_ferro_22.uai`	44	528	2
(d)	`driverlog01ac.wcsp.uai`	71	618	4
(e)	`grid10x10.f10.uai`	100	280	2
(f)	`1502.wcsp.uai`	209	411	4

Table 2: Markov random fields benchmark

We denote as T_{MAR} the cumulative execution time used by the approximate algorithm when calling the MAR tasks, and T_{MMAP} the execution time required for the exact solution of the MMAP task. The two solutions are compared both in terms of *exact match*, i.e., how many times the exact and the approximate sequence of variables to be explained are equal, and *normalized Hamming similarity*, i.e., one minus the normalized Hamming distance between the two sequences. For each model the procedure is iterated q times and the average values are reported.

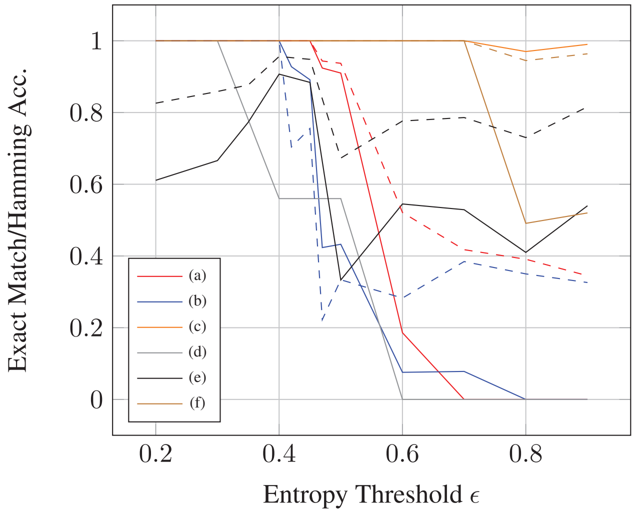

Figure 1: Accuracy trajectories for exact match (line) and Hamming (dashed) accuracies with $k = 5$ and $q = 1000$

In Figure 1, we report the exact-match accuracy trajectories (continuous lines) on the benchmark models for increasing values of the entropy threshold. If the exact-match accuracy is not one, also the Hamming accuracy (dashed line) accuracy is reported. As expected both accuracies decrease for increasing values of ϵ. Notably for low entropy thresholds the algorithm reach very high accuracy levels, while the smoother behaviour of the Hamming trajectories shows that accepting variables with higher entropies produces wrong explanations still including many variables in their right state. This basically proves that the quality of a MMAP solution as achieved by Algorithm 1 depends on the maximum entropy of the variables explained during the different iterations, and this value can be safely regarded as a confidence level of the quality of the resulting solution.

Regarding the execution times, the slowest exact MMAP inferences has been computed for network (d). Remarkably on those instances MMAP2MAR is two order of magnitude faster and we have the average value $T_{\text{MMAP}}/T_{\text{MAR}} \simeq 83$. On the other models, exact MMAP inference is fast and the two approaches have the same order of magnitude. Similar results have been obtained for different values of k.

Conclusions and Outlooks

We presented a heuristic approach to MMAP inference in probabilistic graphical models (both Bayesian networks and Markov random field). The algorithm reduces such a NP$^{\text{PP}}$-complete task to a polynomial number of marginalizations of single variables. Despite its simplicity, preliminary experiments show surprisingly accurate result in finding the most probable explanation when the reliability measure defined together with the algorithm is high. As a future work we intend to provide a deeper experimental validation and also evaluate the possibility of an application of this scheme as a general XAI tool for graphical models, this being in line with the ideas originally presented in (Tiotto 2019).

References

Butz, R.; Hommersom, A.; and van Eekelen, M. 2018. Explaining the Most Probable Explanation. In *Lecture Notes in Computer Science*, volume 11142 LNAI. Springer. 50–63.

Jiang, J.; Rai, P.; and Daume, H. 2011. Message-passing for approximate MAP inference with latent variables. In *Proceedings of NIPS 2011*, 1197–1205.

Koller, D., and Friedman, N. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

Liu, Q., and Ihler, A. 2013. Variational algorithms for marginal MAP. *The Journal of Machine Learning Research* 14(1):3165–3200.

Marinescu, R.; Dechter, R.; and Ihler, A. T. 2014. AND/OR search for marginal MAP. In *UAI*, 563–572.

Marinescu, R.; Dechter, R.; and Ihler, A. T. 2018. Stochastic anytime search for bounding marginal MAP. In *Proceedings of IJCAI 2018*, 5074–5081.

Mauá, D., and de Campos, C. 2012. Anytime marginal MAP inference. In *Proceedings of ICML 2012*, 1471–1478.

Park, J. D., and Darwiche, A. 2002. Solving MAP exactly using systematic search. In *Proceedings of UAI 2002*, 459–468.

Park, J. D., and Darwiche, A. 2004. Complexity results and approximation strategies for MAP explanations. *Journal of Artificial Intelligence Research* 21:101–133.

Tiotto, T. 2019. Explainable AI with probabilistic graphical models. Master's thesis, University of Lugano, Switzerland.

[2] https://github.com/radum2275/merlin