

# A sum-product algorithm with polynomials for computing exact derivatives of the likelihood in Bayesian networks.

Alexandra Lefebvre\*

ALEXANDRA.LEFEBVRE@MATH.CNRS.FR

Grégory Nuel

GREGORY.NUEL@MATH.CNRS.FR

LPSM, CNRS 8001, Probability and Statistics, Sorbonne Université, 4 place Jussieu, 75005 Paris, France.

## Abstract

We consider a Bayesian network with a parameter  $\theta$ . It is well known that the probability of an *evidence* conditional on  $\theta$  (the likelihood) can be computed through a sum-product of potentials. In this work we propose a polynomial version of the sum-product algorithm based on generating functions for computing both the likelihood function and all its exact derivatives. For a unidimensional parameter we obtain the derivatives up to order  $d$  with a complexity  $\mathcal{O}(C \times d^2)$  where  $C$  is the complexity for computing the likelihood alone. For a parameter of  $p$  dimensions we obtain the likelihood, the gradient and the Hessian with a complexity  $\mathcal{O}(C \times p^2)$ . These complexities are similar to the numerical method with the main advantage that it computes exact derivatives instead of approximations.

**Keywords:** Bayesian network; sum-product algorithm; generating functions; derivatives of the likelihood.

## 1. Introduction

We consider a finite set of random variables  $X_{\mathcal{U}} = \{X_1, \dots, X_n\}$  over a parametric Bayesian network (BN) with parameter  $\theta \in \mathbb{R}^p$  (See [Koller and Friedman \(2009\)](#) or [Nielsen and Jensen \(2009\)](#) for a detailed introduction to Bayesian networks). Let  $\text{pa}_u$  be the subset (possibly empty) of  $\mathcal{U}$  associated with the parents of  $X_u$ , and for all  $u \in \mathcal{U}$ , let  $\mathcal{X}_u$  be the discrete or continuous set of values taken by  $X_u$ . It is well known that the probability of an *evidence*  $\text{ev} = \cap_{u=1}^n \{X_u \in \mathcal{X}_u^* \subset \mathcal{X}_u\}$  can be computed through the following sum-product expression:

$$\mathbb{P}(\text{ev}|\theta) = \sum_{X_1} \dots \sum_{X_n} \prod_{u=1}^n \phi_u(X_u, X_{\text{pa}_u}|\theta) \quad (1)$$

where  $\phi_u(X_u, X_{\text{pa}_u}|\theta) = \mathbf{1}_{X_u \in \mathcal{X}_u^*} \mathbb{P}(X_u|X_{\text{pa}_u};\theta)$  are the *potentials* of each variable. From a statistical point of view,  $L_n(\theta) = \mathbb{P}(\text{ev}|\theta)$  is the likelihood of  $\theta$  given the evidence. In statistics, computing the derivatives of the likelihood function is of great interest, especially the first and second order derivatives, from which one can derive the score and the observed Fisher information matrix. These quantities can not only help maximizing the likelihood function (*e.g.* through Newton-based algorithms) but also allow one to obtain confidence intervals on parameters as well as performing hypothesis testing ([Prum, 2010](#)).

---

\* LPSM, CNRS 8001, Probability and Statistics, Sorbonne Université, Paris, France

In probabilistic graphical models such as Bayesian networks, the sensitivity analysis allows one to express  $L(\theta)$  as a polynomial function of  $\theta$  under the hypothesis that all potentials can be expressed as polynomials in  $\theta$  (see [Nielsen and Jensen, 2009](#), pp 184–185). But when the same parameter appears in many potentials, the resulting polynomial is usually of high order, and its computational cost prohibitive. As an extension to the sensitivity analysis [Darwiche \(2003\)](#) proposed a method based on network polynomial with network parameters being the potentials. He uses a multilinear function containing two types of variables (evidence indicators and network parameters) and arithmetic circuits for an efficient computation. This interesting method has however the same limitation than the classical sensitivity analysis: it can only deal with simple parametrization (*e.g.* one parameter for each probability table entry). Alternatively, in the particular context of the hidden Markov models (HMMs), [Cappé and Moulines \(2005\)](#) suggest to compute the expectation of any additive functional through *smoothing recursions*. Taking advantage of the Fisher and Louis identities, they derive the first and second derivatives of the log-likelihood using these smoothing recursions. This approach is computationally effective, but its extension to higher order derivatives is not trivial. On the other hand, several authors suggested to introduce polynomial computations directly in the classical sum-product algorithms in order to compute various quantities of interest: order  $k$  moment of any additive functional in a Bayesian network ([Cowell, 1992](#); [Nilsson, 2001](#)), moment/probability generating functions of the count of a regular expression in a Markov sequence ([Nuel, 2008, 2010](#)).

In the present work, we want to follow the same idea by introducing a sum-product algorithm over polynomials for computing the derivatives of the likelihood of any BN. This new method is both an extension of the work of [Cappé and Moulines \(2005\)](#) from HMMs to BNs, and an extension of the work of [Cowell \(1992\)](#); [Nilsson \(2001\)](#); [Nuel \(2008, 2010\)](#) to the computation of derivatives.

This paper is organized as follows: in Section 2, we will first introduce the definitions, notations and necessary tools to implement our method before explaining in detail its computation. In Section 3, we illustrate our method with two practical examples. The first one is a toy example with a simple binary BN. The second example is taken from the two-point linkage analysis model used in genetics for locating a targeted gene on the genome. Finally, in Section 4, we propose a discussion to expose further perspectives.

## 2. Method

### 2.1. Definitions and notations

**Definition 1 (derivative generating function)** *Let  $f$  be a function of class  $\mathcal{C}^d$  ( $d \in \mathbb{N}$ ) of  $\theta \in \mathbb{R}$ , we define the derivative generating function of  $f$  the generating function associated with the sequence of its derivatives:*

$$D^d f(\theta) = \sum_{k=0}^d f^{(k)}(\theta) z^k$$

where  $z$  is a dummy variable.

**Remark:** We generalize the derivative generating function to a multidimensional parameter with the sequence of partial derivatives. Let  $f$  be a function of class  $\mathcal{C}^d$  of  $\theta = \{\theta_1, \dots, \theta_p\} \in \mathbb{R}^p$ ,

$$D^d f(\theta) = \sum_{k_1 + \dots + k_p \leq d} \frac{\partial^{(k_1 + \dots + k_p)} f(\theta)}{\partial \theta_1^{k_1} \dots \partial \theta_p^{k_p}} z_1^{k_1} \dots z_p^{k_p}$$

where  $z_1, \dots, z_p$  are  $p$  dummy variables.

**Our aim:** We propose in this work a sum-product algorithm based on polynomials to compute  $D^d L(\theta) = \sum_{k=0}^d L^{(k)}(\theta) z^k$  up to an arbitrary order  $d$ .

For the sake of simplicity, we will focus on the unidimensional case and briefly extend the notions to the multidimensional case at the end of the section.

**Definition 2 (Leibniz's product)** Let  $P = \sum_{k=0}^d a_k z^k$  and  $Q = \sum_{k=0}^d b_k z^k$  be two polynomials in  $z$ , we define the Leibniz product of  $P$  and  $Q$  as

$$P \star Q = \sum_{k=0}^d \sum_{i=0}^k \binom{k}{i} a_{k-i} b_i z^k$$

where  $\binom{k}{i}$  is the binomial coefficient. Note that we deliberately drop all coefficients of degree greater than  $d$ .

## 2.2. Computation

In this section we detail the tools of the computation.

**Proposition 3** Let  $f$  and  $g$  be two functions of class  $\mathcal{C}^d$  of  $\theta \in \mathbb{R}$ ,

$$D^d f(\theta) \star D^d g(\theta) = D^d (fg)(\theta).$$

**Proof** Let  $f$  and  $g$  be two functions of class  $\mathcal{C}^d$  of  $\theta \in \mathbb{R}$ . Let  $P = D^d f(\theta) = \sum_{k=0}^d f^{(k)}(\theta) z^k$  and  $Q = D^d g(\theta) = \sum_{k=0}^d g^{(k)}(\theta) z^k$ , then

$$P \star Q = \sum_{k=0}^d \sum_{i=0}^k \binom{k}{i} f^{(k-i)}(\theta) g^{(i)}(\theta) z^k.$$

We recognize the Leibniz's rule for computing the derivatives of the product of two functions which concludes the proof. ■

It is well known that a BN is decomposable into a factorized graphical structure called a *Junction tree* (JT) composed of a set of cliques and a set of edges and following the *Junction tree properties* (see Chapter 4 in [Cowell et al. \(1999\)](#) or [Koller and Friedman \(2009\)](#) for details). Let  $I$  be the number of cliques and edges of the JT. For  $i \in 1, \dots, I$ , we denote by  $\mathcal{C}_i \subset \mathcal{U}$  (resp.  $\mathcal{S}_i \subset \mathcal{U}$ ) the set of labels of the variables in the  $i$ -th clique (resp.  $i$ -th separator) of the JT. Let  $C_i = \{X_u, u \in \mathcal{C}_i\}$  and  $S_i = \{X_u, u \in \mathcal{S}_i\}$ . We denote by

of<sub>u</sub> the choice of a unique  $i \in \{1, \dots, I\}$  such that  $\{X_u, X_{\text{pa}_u}\} \subset C_i$ . Then the polynomial potentials of the cliques for an arbitrary order  $d$  are defined as:

$$\Phi_i^d(C_i|\theta) = \star_{u \in C_i^*} D^d \phi_u(X_u|X_{\text{pa}_u}; \theta) \quad (2)$$

with  $C_i^* = \{u \in \mathcal{U}, \text{of}_u = i\}$ . For all  $i \in \{1, \dots, I\}$ , let  $\text{to}_i \in \{i+1, \dots, I\}$  be the label of the subsequent clique of  $C_i$ . We also define  $\text{from}_i = \{j, \text{to}_j = i\}$ . For all  $i \in \{1, \dots, I\}$ , we recursively define  $\mathcal{V}_i = \{i\} \cup_{j \in \text{from}_i} \mathcal{V}_j$  and  $V_i = \{C_j\}_{j \in \mathcal{V}_i}$ . Then the polynomial *forward messages* for all  $i \in \{1, \dots, I\}$  are the quantities recursively defined as follows:

$$F_i^d(S_i|\theta) = \sum_{V_i \setminus S_i} \star_{u \in \mathcal{V}_i^*} D^d \phi_u(X_u|X_{\text{pa}_u}; \theta)$$

where  $\mathcal{V}_i^* = \{u \in \mathcal{U}, \exists j \in \mathcal{V}_i, \text{of}_u = j\}$ . Note that  $V_I = X_{\mathcal{V}_I^*} = X_{\mathcal{U}}$  and  $S_I = \emptyset$  and we get in particular:

$$F_I^d(\emptyset|\theta) = \sum_{X_{\mathcal{U}}} \star_{\mathcal{U}} D^d \phi_u(X_u|X_{\text{pa}_u}; \theta) = D^d \left( \sum_{X_{\mathcal{U}}} \prod_{\mathcal{U}} \phi_u(X_u|X_{\text{pa}_u}; \theta) \right) = D^d L(\theta).$$

Applying the message passing algorithm with our polynomial potentials results in the following proposition for computing recursively the *forward messages*:

**Proposition 4**  $\forall i \in \{1, \dots, I\}$ ,

$$F_i^d(S_i|\theta) = \sum_{C_i \setminus S_i} \left( \star_{j \in \text{from}_i} F_j^d(S_j|\theta) \right) \star \Phi_i^d(C_i|\theta).$$

**Proof** The proof is straightforward with the belief propagation in JT over which potentials loaded are polynomial potentials, the additive law is the conventional additive law (+) and the multiplicative law is the Leibniz's product ( $\star$ ). Some details of the proof are given below: For all  $i \in \{1, \dots, I\}$ ,

$$F_i^d(S_i|\theta) = \sum_{V_i \setminus S_i} \star_{u \in \mathcal{V}_i^*} D^d \phi_u(X_u|X_{\text{pa}_u}; \theta) = \sum_{\substack{V_j \setminus S_j \\ j \in \text{from}_i}} \sum_{C_i \setminus S_i} \star_{u \in \mathcal{V}_i^*} D^d \phi_u(X_u|X_{\text{pa}_u}; \theta).$$

Recalling the JT properties we have, for all  $i \in \{1, \dots, I\}$ ,  $V_i \setminus S_i = \sqcup_{j \in \text{from}_i} V_j \setminus S_j \sqcup C_i \setminus S_i$  and  $\mathcal{V}_i^* = \sqcup_{j \in \text{from}_i} \mathcal{V}_j^* \sqcup C_i^*$  where  $\sqcup$  is the disjoint union, and therefore

$$F_i^d(S_i|\theta) = \sum_{C_i \setminus S_i} \left( \star_{j \in \text{from}_i} \underbrace{\sum_{V_j \setminus S_j} \left( \star_{u \in \mathcal{V}_j^*} D^d \phi_u(X_u|X_{\text{pa}_u}; \theta) \right)}_{F_j^d(S_j|\theta)} \right) \star_{u \in C_i^*} D^d \phi_u(X_u|X_{\text{pa}_u}; \theta)$$

which concludes the proof by induction. ■

**Remark:** We make the choice here to focus on *forward messages* as we are only interested in the likelihood and its derivatives but the extension to *backward messages* to compute joint probabilities and their derivatives is straightforward.

The extension of our method to a BN with a multidimensional parameter  $\theta \in \mathbb{R}^p$ ,  $\theta = \theta_1, \dots, \theta_p$ , implies multivariate polynomials with as many dummy variables as dimensions of the parameter. The “Leibniz’s product” of two multivariate polynomials is defined as:

$$P \star Q = \sum_{k_1 + \dots + k_p \leq d} \sum_{i_1=0}^{k_1} \dots \sum_{i_p=0}^{k_p} \binom{k_1}{i_1} \dots \binom{k_p}{i_p} a_{k_1-i_1, \dots, k_p-i_p} b_{i_1, \dots, i_p} z_1^{k_1} \dots z_p^{k_p}$$

where  $f$  is a function of class  $\mathcal{C}^d$  of  $\theta \in \mathbb{R}^p$  and  $P$  and  $Q$  are two polynomials of degree at most  $d$  in  $p$  dummy variables. The generalization of Proposition 4 to a multidimensional parameter gives:

$$F_I^d(\emptyset|\theta) = \sum_{k_1 + \dots + k_p \leq d} \frac{\partial^{(k_1 + \dots + k_p)} L(\theta)}{\partial \theta_1^{k_1} \dots \partial \theta_p^{k_p}} z_1^{k_1} \dots z_p^{k_p} = D^d L(\theta).$$

### 3. Results

In this section we illustrate the implementation and results with our method with two examples. We first present an illustrative toy-example with a BN over binary variables and in the second part we will illustrate our method with the two-point linkage analysis which aims to locate a targeted gene on the genome.

For the sake of simplicity, we will only consider here a degree  $d = 2$  and unidimensional parameters but our method is valid for any arbitrary degree and multidimensional parameters.

#### 3.1. Toy-example: A binary BN

Let us consider the BN over  $n$  variables  $X_{\mathcal{U}} = \{X_u\}_{u=1, \dots, n}$  represented in the Directed Acyclic Graph (DAG) Figure 1 where  $n = 7$  and  $X_{\mathcal{U}} \in \{0, 1\}^n$ . For all  $u \in \{1, \dots, n\}$  we assume that

$$\mathbb{P}(X_u = 1 | X_{\text{pa}_u}; \theta) = \frac{\exp\left(\mu + \theta \sum_{v \in \text{pa}_u} X_v\right)}{1 + \exp\left(\mu + \theta \sum_{v \in \text{pa}_u} X_v\right)}$$

where  $\mu = -0.5$  is assumed to be known.

The *junction tree*  $JT_1$  of Figure 1 has been obtained from this DAG. For all  $i \in \{1, \dots, I = 4\}$ ,  $\mathcal{C}_i^*$  is the set of labels of the starred variables in  $C_i$ .

Let

$$P_k = f_k(\theta) + f'_k(\theta)z + f''_k(\theta)z^2 \quad \text{with} \quad f_k(\theta) = \frac{e^{\mu + k\theta}}{1 + e^{\mu + k\theta}}.$$

Note that the properties of the derivatives give  $1 - f_k(\theta) + (1 - f_k)'(\theta)z + (1 - f_k)''(\theta)z^2 = 1 - P_k$  and, for all  $u \in \{1, \dots, n\}$ , the polynomial potential of  $X_u$  is given by the expression:

$$D^2 \phi_u \left( X_u \left| \sum_{v \in \text{pa}_u} X_v = k; \theta \right. \right) = \begin{cases} P_k & \text{if } X_u = 1 \\ 1 - P_k & \text{if } X_u = 0 \end{cases}.$$

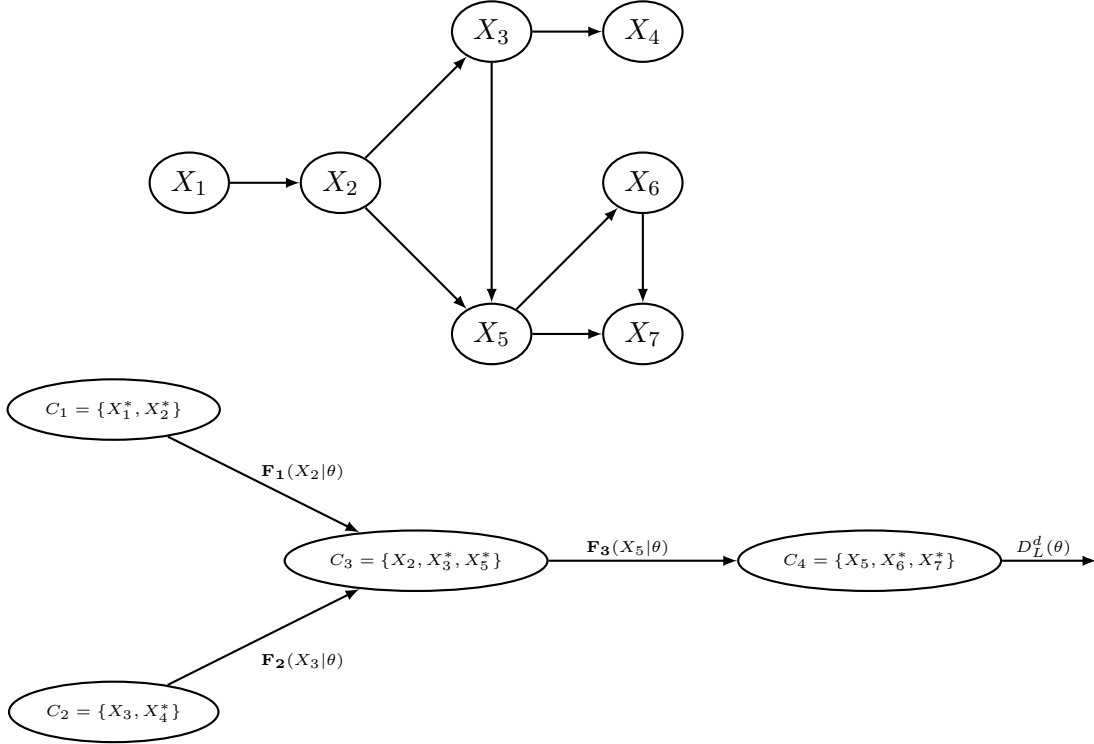


Figure 1: DAG of the toy-example and a corresponding JT containing four cliques. For all  $i \in \{1, \dots, I = 4\}$ ,  $C_i^*$  is the set of labels of the starred variables in  $C_i$ .

We now consider the *evidence*  $\text{ev} = \{X_1 = 0, X_7 = 1\}$  which leads to  $D^2\phi_1(X_1 = 1) = D^2\phi_7(X_7 = 0|X_5 + X_6; \theta) = 0$ , the null polynomial. For all  $i \in \{1, \dots, 4\}$ , the polynomial potentials  $\Phi_i^2(C_i|\theta)$  are computed with Equation (2) and the polynomial *forward messages*  $F_i^2(S_i|\theta)$  recursively with Proposition 4. For example, if we drop  $\theta$  in the notations and we assume that  $\Phi_1^2(X_1, X_2)$ ,  $\Phi_2^2(X_3, X_4)$ ,  $F_1^2(X_2)$  and  $F_2^2(X_3)$  are computed, the quantities  $\Phi_3^2(X_2, X_3, X_5)$  and  $F_3^2(X_5)$  are given by the equations:

$$\Phi_3^2(X_2, X_3, X_5) = D^2\phi_3(X_3|X_2) \star D^2\phi_5(X_5|X_2, X_3)$$

and

$$F_3^2(X_5) = \sum_{X_2, X_3} F_1^2(X_2) \star F_2^2(X_3) \star \Phi_3^2(X_2, X_3, X_5).$$

Note that building  $\text{JT}_1$  and computing all polynomial clique potentials and *forward messages* is strictly equivalent to performing a *variable elimination* with the following elimination order:  $X_1, X_4, \{X_3, X_2\}, \{X_5, X_6, X_7\}$  (see Koller and Friedman, 2009).

Table 1 gives the expression of a few chosen polynomial clique potentials and polynomial *forward messages* for  $\theta = 1$  and  $\text{ev} = \{X_1 = 0, X_7 = 1\}$ .

Table 1: A sample of chosen clique potentials and *forward messages* in JT<sub>1</sub>.

$\Phi_2^2(X_3 = 0, X_4 = 0)$	0.622459331201855
$\Phi_2^2(X_3 = 0, X_4 = 1)$	0.377540668798145
$\Phi_2^2(X_3 = 1, X_4 = 0)$	$0.3775407 - 0.2350037z + 0.05755679z^2$
$\Phi_2^2(X_3 = 1, X_4 = 1)$	$0.6224593 + 0.2350037z - 0.05755679z^2$
$F_2^2(X_3 = 0)$	1
$F_2^2(X_3 = 1)$	1
$F_3^2(X_5 = 0)$	$0.2655306 - 0.1096367z + 0.06460333z^2$
$F_3^2(X_5 = 1)$	$0.3569288 + 0.1096367z - 0.06460333z^2$
$F_4^2(\emptyset) = D^2L(\theta)$	$0.3903244 + 0.1678956z - 0.05803046z^2$

Note that,  $X_3$  having no parents and  $X_4$  having only  $X_3$  as a parent, the potentials  $\Phi_2^2(X_3 = 0, X_4)$  for  $X_4 \in \{0, 1\}$  are of degree 0. We also can notice that  $F_2^2(X_3 = 0) = F_2^2(X_3 = 1) = 1$ . Indeed  $F_2^2(X_3 = 0) = (1 - P_0) + P_0$  and  $F_2^2(X_3 = 1) = (1 - P_1) + P_1$ . All other potentials in this table are of degree greater than 0. In particular  $F_4^2(\emptyset)$  which has been verified numerically (data not shown) is a polynomial in  $z$  containing  $L(\theta)$  and its derivatives up to order 2 in its coefficients.

Table 2 gives the values obtained for the log-likelihood of  $\theta = 1$  and its derivatives up to order 2 for different simulations of  $N$  values for  $\{X_1, \dots, X_7\}$  with  $\theta = 1$ . Let  $ev_{ab}$  denote the observation  $\{X_1 = a, X_7 = b\}$  and  $N_{ab}$  the number of observations of the *evidence*  $ev_{ab}$  among  $N$ . Let us recall that the true  $\theta$  under simulation is  $\theta^* = 1$ . We can see as expected that  $\ell'(1)/\ell(1)$  tends towards 0 with an increasing  $N$  as  $L'(\theta^* = 1)$  tends toward 0. We can also see that  $-\ell''(1)$  is small for  $N=1$  leading to a large variance which was expected in the case of the observation of a unique couple  $\{X_1, X_7\}$ . On the other hand,  $-\ell''(1)$  increases linearly with an increasing  $N$  as expected as  $1/(-\ell''(\theta))$  (the Cramer-Rao bound for the variance of  $\beta$ ) decreases linearly with  $N$ .

Table 2: Log-likelihood and its derivatives up to order 2 computed with our algorithm for different simulations of  $N$  values for  $\{X_1, \dots, X_7\}$  leading to  $N_{ab}$  observed couples  $\{X_1 = a, X_7 = b\}$ .

$N$	$N_{00}$	$N_{01}$	$N_{10}$	$N_{11}$	$\ell(1)$	$\ell'(1)$	$\ell'(1)/\ell(1)$	$\ell''(1)$
1	1	0	0	0	-1.460	$-7.233 \times 10^{-1}$	$4.952 \times 10^{-1}$	$-2.731 \times 10^{-1}$
1	0	1	0	0	$-9.408 \times 10^{-1}$	$4.301 \times 10^{-1}$	$-4.572 \times 10^{-1}$	$-3.337 \times 10^{-1}$
1	0	0	1	0	-2.012	$-8.981 \times 10^{-1}$	$4.466 \times 10^{-1}$	$-6.914 \times 10^{-1}$
1	0	0	0	1	-1.412	$4.933 \times 10^{-1}$	$-3.494 \times 10^{-1}$	$-3.067 \times 10^{-1}$
50	4	20	8	18	$-6.616 \times 10^1$	7.405	$-1.119 \times 10^{-1}$	$-1.882 \times 10^1$
500	136	187	71	106	$-6.670 \times 10^2$	$-2.940 \times 10^1$	$4.408 \times 10^{-2}$	$-1.811 \times 10^2$
5000	1198	1868	753	1181	$-6.689 \times 10^3$	$-1.566 \times 10^2$	$2.341 \times 10^{-2}$	$-1.833 \times 10^3$
50,000	11592	19470	6681	12257	$-6.599 \times 10^4$	$3.754 \times 10^1$	$-5.690 \times 10^{-4}$	$-1.804 \times 10^4$

### 3.2. Two-point linkage in genetics

#### 3.2.1. INTRODUCTION TO THE TWO-POINT LINKAGE IN GENETICS

The two-point linkage analysis is a statistical analysis in genetic epidemiology aiming at locating a targeted gene on the genome. A detailed explanation of basics in genetics and genetic linkage is given in [Lauritzen and Sheehan \(2003\)](#) Section 2.2 and we recommend those not familiar with genetic data to read this reference. We give below some brief recalls:

Most of the human cells are diploid (containing pairs of chromosomes) with one paternal chromosome and one maternal chromosome. In the gonads, a diploid cell with double-stranded chromosomes split into four haploid cells with single-stranded chromosomes called gametes, dedicated to be transmitted to the offspring. During meiosis, two chromosomes of the same pair can exchange genetic material and produce recombinant gametes. This phenomenon is called a crossover. Figure 2 represents a simplified meiosis with only one pair of chromosomes. The closest two genes are on the chromosome the less chances their alleles to be separated during meiosis. The two-point linkage analysis uses this phenomenon, the results of genetic tests for a marker whose location on the genome is known, the penetrance of the targeted gene (probability of the trait (or phenotype) conditional on the genotype) and the allele frequencies previously estimated with segregation analysis, in order to estimate the distance between the targeted gene (whose alleles are here denoted  $X$ ) and the marker (whose alleles are here denoted  $M$ ) as a function of the fraction of recombinant gametes  $\theta = \#R/(\#R + \#NR)$ . Figure 3 is a DAG of the variables commonly used in genetic linkage where the names of the variables are the same as in Figure 2 with the addition of the traits  $Y$  coded by the targeted gene and the genetic tests for the marker  $G$ . Each allele  $M$  and  $X$  are labeled pat or mat according to their origin.  $SM \in \{\text{pat}, \text{mat}\}$  (resp.  $SX \in \{\text{pat}, \text{mat}\}$ ) denotes the selector (the origin) of the marker (resp. the targeted gene) for the offspring.

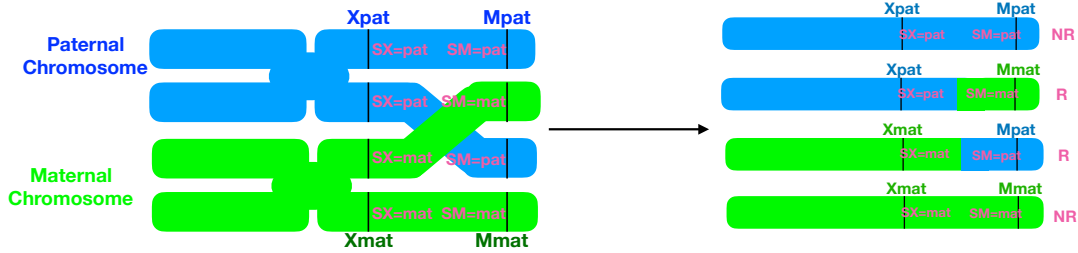


Figure 2: Simplified meiosis with only one pair of chromosomes between which a crossover happens.  $M_{\text{pat}}$  (resp.  $M_{\text{mat}}$ ) stands for the paternal (resp. maternal) allele of the marker,  $X_{\text{pat}}$  (resp.  $X_{\text{mat}}$ ) stands for the paternal (resp. maternal) allele of the targeted gene and  $SX \in \{\text{pat}, \text{mat}\}$  (resp.  $SM \in \{\text{pat}, \text{mat}\}$ ) stands for the selector for  $X$  (resp.  $M$ ).  $R$  (resp.  $NR$ ) denotes a recombinant (resp. non-recombinant) gamete.



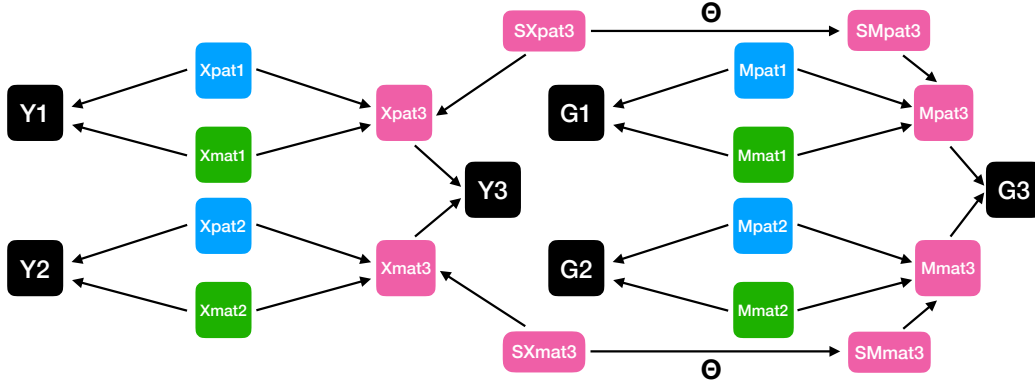


Figure 3: DAG of the variables commonly used in genetic linkage analysis in a simple trio with one father labeled 1, one mother labeled 2 and one child labeled 3.  $Y$  stands for the trait and  $G$  for the genetic test for the marker.

### 3.2.2. POLYNOMIALS ASSOCIATED WITH THE VARIABLES

In this section and for the sake of simplicity we will expose our method with a very simple trio composed of one father, one mother and one child and we will again be interested in computing derivatives up the degree  $d = 2$ . Figure 3 represents a DAG of the variables involved where the labels pat and mat for the selectors denote the origin (paternal or maternal) of the transmitted allele  $M$  and  $X$ .

Let  $W_{\mathcal{U}} = \{W_u\}_{u=1,\dots,n}$  the set of variables in Figure 3. We are now interested in the implementation of the functions  $D^2\mathbb{P}(W_u|W_{\text{pa}_u};\theta)$ . Note that, for all  $W_u \in W_{\mathcal{U}} \setminus \{SM_{\text{pat}_3}, SM_{\text{mat}_3}\}$  we have  $D^2\mathbb{P}(W_u|W_{\text{pa}_u};\theta) = \mathbb{P}(W_u|W_{\text{pa}_u})$  are all 0 degree polynomials.

As an example we will consider the `contrao12a` example given in Mendel package [Lange et al. \(2013\)](#) using PGM1 as the marker and RADIN as the targeted gene. PGM1 has 4 alleles ( $\{1+, 1-, 2+, 2-\}$ ) with given allele frequencies and given conditional probabilities of the results of the genetic test conditional on the genotype. RADIN is biallelic ( $\{+, -\}$ ) with given allele frequencies and given penetrance.

**Constant polynomials.** We assume a Mendelian transmission of the alleles such that  $D^d\mathbb{P}(SX_{\text{pat}_3} = s) = D^d\mathbb{P}(SX_{\text{mat}_3} = s) = 0.5$  with  $s \in \{\text{pat}, \text{mat}\}$ . For all  $W_u \in W_{\mathcal{U}} \setminus \{SM_{\text{pat}_3}, SM_{\text{mat}_3}, SX_{\text{pat}_3}, SX_{\text{mat}_3}\}$ ,  $D^d\mathbb{P}(W_u|W_{\text{pa}_u};\theta) = \mathbb{P}(W_u|W_{\text{pa}_u})$  are straightforward assuming Hardy-Weinberg equilibrium for the founders (constant allele and genotype frequencies from generation to generation) and using allele frequencies, conditional probabilities of the genetic tests and penetrance of the targeted gene.

**Polynomials of strictly positive degree.** As  $\mathbb{P}(SM_{\text{pat}_3}|SX_{\text{pat}_3};\theta)$  and  $\mathbb{P}(SM_{\text{mat}_3}|SX_{\text{mat}_3};\theta)$  depend on  $\theta$ , their polynomials are of degree strictly positive and we have

$$D^2\mathbb{P}(SM_{\text{pat}_3} = SX_{\text{pat}_3}|SX_{\text{pat}_3};\theta) = D^2\mathbb{P}(SM_{\text{mat}_3} = SX_{\text{mat}_3}|SX_{\text{mat}_3};\theta) = (1 - \theta) - z$$

and

$$D^2\mathbb{P}(SM_{\text{pat}_3} \neq SX_{\text{pat}_3}|SX_{\text{pat}_3};\theta) = D^2\mathbb{P}(SM_{\text{mat}_3} \neq SX_{\text{mat}_3}|SX_{\text{mat}_3};\theta) = \theta + z$$

In practice, as  $\theta$  is constrained, we use the logit transformation  $\theta = e^\beta/(1+e^\beta)$  and therefore:

$$D^2\mathbb{P}(SM_{\text{pat}_3} = SX_{\text{pat}_3}|SX_{\text{pat}_3};\beta) = \frac{1}{1+e^\beta} - \frac{e^\beta}{(1+e^\beta)^2}z - \frac{e^\beta(1-e^\beta)}{(1+e^\beta)^3}z^2$$

and

$$D^2\mathbb{P}(SM_{\text{pat}_3} \neq SX_{\text{pat}_3} | SX_{\text{pat}_3}; \beta) = \frac{e^\beta}{1 + e^\beta} + \frac{e^\beta}{(1 + e^\beta)^2} z + \frac{e^\beta (1 - e^\beta)}{(1 + e^\beta)^3} z^2$$

and respectively for the maternal selector  $SM_{\text{mat}_3}$ . Note that preprocessing first the polynomials related to the marker and the polynomials related to the targeted gene before moralizing the graph is crucial in the context of genetic analysis as explained in [Lauritzen and Sheehan \(2003\)](#) Section 3.3 in order to remove unnecessary links. The following results have been obtained after preprocessing potentials.

### 3.2.3. RESULTS

In this section we give some results of the implementation of our method over the two-point linkage analysis for the KUS family (22 individuals) and the whole `control2a` (93 individuals) set of pedigrees in Mendel package ([Lange et al., 2013](#)). We suggest the readers not familiar with genetic linkage notions the following reference for the basics in quantities used such as the LOD score: section 4.1 [Lauritzen and Sheehan \(2003\)](#).

In genetic linkage analysis, the null hypothesis  $\hat{\theta} = \theta_0 = 0.5$  is tested against  $\hat{\theta} < 0.5$  with the estimation of a quantity named the LOD score,  $\text{LOD}(\theta) = \log_{10}(L(\theta)/L(0.5))$ . Defining  $Z(\beta) = \log_{10}(L(e^\beta/(1 + e^\beta))/L(0.5))$ , we computed values of  $Z(\beta)$  for various  $\beta$  and obtained the same values as computed with the Mendel package for the corresponding  $\text{LOD}(\theta)$ . The computed derivatives of  $\tilde{L}(\beta) = L(e^\beta/(1 + e^\beta))$  allow to calculate confidence intervals on  $\theta$  and to perform likelihood ratio test, Wald test and score test whose results are compared in Table 3 for both the KUS family and the whole set of families in `control2a`. As expected, the confidence intervals are shrinking with an increasing number of individuals. The likelihood ratio test is the one commonly done in genetic linkage through the LOD score. We can see Table 3 that the three tests are not equivalent though all p-values are significant. One further extension of this work could be a comparison of the power of these tests in genetic linkage in different pedigrees.

Table 3: Confidence intervals on  $\theta$  and statistics of the likelihood ration test (LR), Wald test (W) and Score test (S) along with the p-values.

	$\hat{\theta}$	IC 95%	LR (p-value)	W (p-value)	S (p-value)
KUS(n=22)	0.059	[0.008, 0.320]	14.574 ( $1.3 \times 10^{-4}$ )	7.264 ( $7.0 \times 10^{-3}$ )	32.010 ( $1.5 \times 10^{-8}$ )
ALL (n=93)	0.193	[0.106, 0.326]	17.010 ( $3.7 \times 10^{-5}$ )	15.821 ( $7 \times 10^{-5}$ )	12.900 ( $3.3 \times 10^{-4}$ )

## 4. Discussion

We proposed in this work an algorithm on polynomials to compute  $L(\theta)$  and its exact derivatives in a single belief propagation up to an arbitrary order  $d$  with a complexity of  $\mathcal{O}(C \times d^2)$  for a unidimensional parameter and  $\mathcal{O}(C \times p^2)$  if  $d = 2$  for a parameter of dimension  $p$  where  $C$  is the complexity to compute  $L(\theta)$  alone with a classical belief propagation algorithm. These complexities are similar to the one needed in the classical

empirical method (numerical derivation) and to the approach of Cappé and Moulines (2005) with some advantages. Its main advantage over the empirical method is that it provides exact value in a single computation with no need to iteratively converge to the solution. Its main advantage over Cappé and Moulines (2005) is its ability to deal with any BN rather than only HMMs as well as its straightforward formulation and implementation for any degree.

We have considered an application in the two-point linkage analysis and we obtained confidence intervals on the parameter. The derivatives of the likelihood also allowed us to perform other tests than the LR test commonly used in linkage analysis to compare their performances. Our method is extendable to any parametric BN and more generally in sum-product computations in probabilistic graphical models.

As we focused in this work on the likelihood and its derivatives only, we restricted our explanation to the *forward* quantities but of course *backward* quantities can be computed if needed. It could be interesting to study the quantities  $D^d \mathbb{P}(X_v, \text{ev} | X_{\text{pa}_v}; \theta)$  with  $v \in \mathcal{U}$ . These quantities could be used to obtain  $D^d \mathbb{P}(X_v | \text{ev}; \theta)$  for an arbitrary  $d$  and therefore help studies about variations of the derivatives of marginal probabilities.

## Acknowledgments

This work was funded by the *Ligue Nationale Contre le Cancer* (PhD grant).

## References

- O. Cappé and E. Moulines. Recursive computation of the score and observed information matrix in hidden Markov models. In *13th IEEE Workshop on Statistical Signal Processing, Bordeaux, France*, 2005.
- R. G. Cowel. Calculating moments of decomposable functions in Bayesian networks. *Research Report 109, Department of Statistical Sciences, University College London, London.*, 1992.
- R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic networks and expert systems*. Springer-Verlag, 1999.
- A. Darwiche. A differential approach to inference in Bayesian networks. *Journal of the ACM (JACM)*, 50(3):280–305, 2003.
- D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- K. Lange, J. C. Papp, J. S. Sinsheimer, R. Sripracha, H. Zhou, and E. M. Sobel. Mendel: the Swiss army knife of genetic analysis programs. *Bioinformatics*, 29(12):1568–1570, 2013.
- S. L. Lauritzen and N. A. Sheehan. Graphical models for genetic analyses. *Statistical Science*, pages 489–514, 2003.

- T. D. Nielsen and F. V. Jensen. *Bayesian networks and decision graphs*. Springer Science & Business Media, 2009.
- D. Nilsson. The computation of moments of decomposable functions in probabilistic expert systems. In *Proceedings of the Third International Symposium on Adaptive Systems*, pages 116–21, 2001.
- G. Nuel. Pattern Markov chains: optimal Markov chain embedding through deterministic finite automata. *Journal of Applied Probability*, 45(1):226–243, 2008.
- G. Nuel. On the first k moments of the random count of a pattern in a multistate sequence generated by a Markov source. *Journal of Applied Probability*, 47(4):1105–1123, 2010.
- B. Prum. *La démarche statistique*. Cépaduès, 2010.