Fisher Information in Gaussian Graphical Models

Jason K. Johnson

September 21, 2006

Abstract

This note summarizes various derivations, formulas and computational algorithms relevant to the Fisher information matrix of Gaussian graphical models with respect to either an exponential parameterization (related to the information form) or the corresponding moment parameterization.

1 Gauss-Markov Models

The probability density of a Gaussian random vector $x \in \mathbb{R}^n$ may be expressed in *information form*:

$$p(x) \propto \exp\{-\frac{1}{2}x^T J x + h^T x\}$$

with parameters $h \in \mathbb{R}^n$ and $J \in \mathbb{R}^{n \times n}$, where J is a symmetric positive-definite matrix. The mean and covariance of x are given by:

$$\begin{aligned} \mu &\triangleq p[x] = J^{-1}h \\ P &\triangleq p[(x-\mu)^T(x-\mu)] = J^{-1} \end{aligned}$$

where we denote expectation of $f(x)$ with respect to p by $p[f] \triangleq \int p(x)f(x)dx$. In this note, we focus on the zero-mean case $\mu = 0$. Thus, $h = 0$ as well.

A Gaussian distribution is *Markov* on a graph $\mathcal{G} = (V, E)$, with vertices $V = \{1, \ldots, n\}$, if and only if $J_{ij} = 0$ for all $\{i, j\} \notin E$. Thus, Markov models have a reduced information parameterization based on a sparse J matrix.

2 Exponential Family and Fisher Information

The Gaussian distributions can also be represented as an *exponential family*:

$$p(x) = \exp\{\theta^T \phi(x) - \Phi(\theta)\}$$

with parameters $\theta \in \mathbb{R}^d$ and sufficient statistics $\phi : \mathbb{R}^n \to \mathbb{R}^d$. To obtain the information form we define statistics:

$$\phi(x) = (x_i^2, i \in V) \cup (x_i x_j, \{i, j\} \in E)$$

The corresponding exponential parameters then correspond to elements of the information matrix:

$$\theta = (-\frac{1}{2}J_{ii}, i \in V) \cup (-J_{ij}, \{i, j\} \in E)$$

An implicit parameterization of the exponential family is given by the *moment parameters* defined by $\eta = p[\phi]$. In the Gauss-Markov family on $\mathcal{G}$, the moment parameters correspond to elements of the covariance matrix P:

$$\eta = (P_{ii}, i \in V) \cup (P_{ij}, \{i,j\} \in E)$$

The θ and η parameters are related by a bijective map: $\eta = \Lambda(\theta) \triangleq p_\theta[\phi]$.

The *cumulant generating function* of this exponential family is

$$\begin{aligned}\Phi(\theta) &\triangleq \log \int \exp\{\theta^T \phi(x)\} dx \\ &= \frac{1}{2}\{\log\det J^{-1}(\theta) + n \log 2\pi\}\end{aligned}$$

which serves to normalize the probability distribution. Derivatives of this function yield cumulants of the distribution. In particular, the gradient $\nabla\Phi(\theta) = (\frac{\partial\Phi(\theta)}{\partial\theta_i})$ is equal to the moment map:

$$\nabla\Phi(\theta)_i = \frac{\partial\Phi(\theta)}{\partial\theta_i} = \Lambda_i(\theta) = \eta_i$$

The Hessian matrix $\nabla^2\Phi(\theta) = (\frac{\partial^2\Phi(\theta)}{\partial\theta_i\partial\theta_j})$ is equal to the covariance of the sufficient statistics, which incidentally is also the *Fisher information* of the exponential family with respect to θ parameters:

$$G(\theta) \triangleq \nabla^2\Phi(\theta) = p_\theta[(\phi(x) - \Lambda(\theta))(\phi(x) - \Lambda(\theta))^T] = p_\theta[\nabla \log p_\theta(x) \nabla \log p_\theta(x)^T]$$

Note, since $\nabla\Phi(\theta) = \Lambda(\theta)$, the Fisher information matrix is also equal to the Jacobian $D\Lambda(\theta) = (\frac{\partial\eta_i}{\partial\theta_j})$ and therefore describes the first-order sensitivity of the moments to perturbations in θ parameters: $\delta\eta \approx G(\theta)\delta\theta$.

Consider the function $f(X) = \log\det X$. Boyd and Vandenberge derive first and second order differential analysis. The gradient is $\nabla f(X) = X^{-1}$, that is to say that $f(X + dX) \approx f(X) + \mathrm{tr}(X^{-1}dX)$ where $\mathrm{tr}(AB)$ is the natural inner product of two matrices A and B. The Hessian is naturally described by the action of a quadratic form on the vector space of symmetric matrices:

$$\nabla^2 f(X)(dX, dY) = -\mathrm{tr}(X^{-1}dXX^{-1}dY)$$

This is related to the fact that for the function $F(X) = X^{-1}$ we have $dF = -X^{-1}dXX^{-1}$, a matrix generalization of $d(1/x) = -dx/x^2$.

Using these relations, we can evaluate explicit formulas for the elements of the Fisher information matrix $G(\theta)$ in a Gauss-Markov model. We start by writing $J(\theta)$ as

$$J(\theta) = -2\sum_{i\in V}\theta_i e_i e_i^T - \sum_{\{i,j\}\in E}\theta_{ij}(e_i e_j^T + e_j e_i^T) \tag{1}$$

where $\{e_i\}$ are the standard basis vectors of $\mathbb{R}^n$. Note that $V \cup E$ serves as the index set of θ, which has dimension $d = |V| + |E|$. Thus, $\frac{\partial J(\theta)}{\partial\theta_i} = -2e_i e_i^T$ and $\frac{\partial J(\theta)}{\partial\theta_{ij}} = -(e_i e_j^T + e_j e_i^T)$. Now, we evaluate the matrix elements of $G(\theta)$ using the chain rule and the quadratic form for the Hessian of the log-determinant function. For example, the element corresponding to edges $\{i,j\}$ and $\{k,l\}$ is obtained as:

$$\begin{aligned}G_{ij,kl} &= \tfrac{1}{2}\mathrm{tr}(J^{-1}\frac{\partial J}{\partial\theta_{ij}}J^{-1}\frac{\partial J}{\partial\theta_{kl}}) \\ &= \tfrac{1}{2}\mathrm{tr}(P(e_i e_j^T + e_j e_i^T)P(e_k e_l^T + e_l e_k^T)) \\ &= P_{il}P_{jk} + P_{ik}P_{jl}\end{aligned}$$

Here, we have used $\mathrm{tr}(Pe_ie_j^TPe_ke_l^T) = \mathrm{tr}(e_l^TPe_ie_j^TPe_k) = P_{li}P_{jk}$. Similarly, we obtain $G_{ij,k} = 2P_{ik}P_{jk}$ for edge-to-vertex terms and $G_{i,k} = 2P_{ik}^2$ for vertex-to-vertex terms.

These formulas for the elements of G are also valid for the Fisher information in any Markov sub-family of the full Gaussian model, because the Fisher information in the Markov model is a submatrix of the Fisher information in the full model. This nesting relation follows from the fact that the Markov sub-family is obtained by constraining some θ parameters to zero, so the Hessian of the resulting cumulant-generating function is a submatrix of the Hessian in the unconstrained model.

In practice, we may not need to explicitly evaluate the Fisher information matrix. For instance, we may only wish to apply the Fisher information matrix to a given vector $\Delta\theta$ in order to compute the first-order change in moments $\Delta\eta = G(\theta)\Delta\theta$. Often, this can be done more efficiently than explicitly computing $G(\theta)$ and evaluating the matrix-vector product. For instance, in the fully connected model, the explicit computation requires $\mathcal{O}(d^2) = \mathcal{O}(n^4)$ computations to compute $G(\theta)$ and evaluate the matrix-vector product. But, using the fact that $G(\theta) = \frac{\partial\eta}{\partial\theta}$ and that $d(X^{-1}) = -X^{-1}dXX^{-1}$ we can more efficiently evaluate $\Delta\eta$ as follows:

1. Let $\Delta J = J(\theta + \Delta\theta) - J(\theta)$.
2. Compute $P = J^{-1}$ and $\Delta P = -P\Delta JP$.
3. Let $\Delta\eta_i = \Delta P_{ii}$ for all $i \in V$ and $\Delta\eta_{ij} = \Delta P_{ij}$ for all $\{i, j\}$.

This approach only requires $\mathcal{O}(n^3)$, which is about n times faster than the explicit computation. In a similar manner, we can more efficiently apply the Fisher information matrix in models with thin graphical structure that allow efficient computations of the moment parameters. We discuss this further in Section 4.

3 Moments, Entropy and Fisher Information

Entropy is defined $h(p) = -p[\log p]$ and provides a measure of randomness or uncertainty of a random variable with distribution p. A Gaussian density with covariance P has entropy:

$$h(P) = \frac{1}{2}(\log\det P + n\log 2\pi e)$$

In the maximum-entropy approach to modeling, one seeks the probability distribution that maximizes entropy subject to linear moment constraints:

$$\begin{array}{rl} \max & h(p) \\ \text{s.t.} & p[\phi] = \eta \end{array}$$

where ϕ are a set of selected statistics and η are the prescribed expected values of those statistics (for example, the empirical averages of those statistics taken from some data set). A well-known maximum-entropy principle asserts that the solution to this problem (when it exists) will be of the form of an exponential family based on ϕ where the θ are chosen to realize the prescribed moments. Let p_η denote the maximum-entropy model with moments η and $h(\eta)$ denote it's entropy.

There is an important duality principle underlying the maximum-entropy principle. The negative entropy $\Psi(\eta) \triangleq -h(\eta)$ and the cumulant generating function $\Phi(\theta)$ are convex-conjugate functions:

$$\begin{aligned} \Phi(\theta) &= \sup_\eta\{\theta^T\eta - \Psi(\eta)\} \\ \Psi(\eta) &= \sup_\theta\{\eta^T\theta - \Phi(\theta)\} \end{aligned}$$

Moreover, the optimal value of η and θ in these two variational problems are respectively $\eta = \Lambda(\theta)$ and $\theta = \Lambda^{-1}(\eta)$.

As a consequence of this duality, it holds that the gradient of negative entropy function evaluated at η is equal to the corresponding θ parameters:

$$\nabla\Psi(\eta) = \Lambda^{-1}(\eta) = \theta$$

Also, the Hessian of the negative entropy function $\Psi(\eta)$ is equal to the *inverse* of the Hessian of $\Phi(\theta)$ evaluated for the corresponding $\theta = \Lambda^{-1}(\eta)$:

$$\nabla^2\Psi(\eta) = \nabla^2\Phi(\Lambda^{-1}(\eta))^{-1}$$

Recall that $G(\theta) = \nabla^2\Phi(\theta)$ is the Fisher information associated with the θ parameters. Similarly, $G^*(\eta) \triangleq \nabla^2\Psi(\eta)$ is the Fisher information in the η parameterization:

$$G^*(\eta) = p_\eta \left[\nabla \log \Psi(\eta) \nabla \log \Psi(\eta)^T\right]$$

We can also use the Hessian of the log-determinant function to compute the Fisher information $G^*(\eta)$ in the full Gaussian model (not imposing any Markov constraints). Then, P is fully parameterized by η:

$$P(\eta) = \sum_i \eta_i e_i e_i^T + \sum_{\{i,j\}} \eta_{ij}(e_i e_j^T + e_j e_i^T)$$

Thus, $\frac{\partial P}{\partial \eta_i} = e_i e_i^T$ and $\frac{\partial P}{\partial \eta_{ij}} = (e_i e_j^T + e_j e_i^T)$. For example, the edge-to-edge terms are given by:

$$\begin{aligned} G^*_{ij,kl} &= \frac{1}{2}\mathrm{tr}(P^{-1}\frac{\partial P}{\partial \eta_{ij}} P^{-1}\frac{\partial P}{\partial \eta_{kl}}) \\ &= \frac{1}{2}\mathrm{tr}(J(e_i e_j^T + e_j e_i^T) J(e_k e_l^T + e_l e_k^T)) \\ &= J_{ik}J_{jl} + J_{il}J_{jk} \end{aligned}$$

Similarly, we obtain $G^*_{ij,k} = J_{ik}J_{jk}$ and $G^*_{i,k} = \frac{1}{2}J^2_{ik}$. Note, these formulas differ slightly from the analogous formulas for $G(\theta)$. Again, it is worth noting that $\Delta\theta = G^*\Delta\eta$ can be more efficiently computed using: $\Delta\eta \to \Delta P \to \Delta J = -J\Delta P J \to \Delta\theta$. This is an $\mathcal{O}(n^3)$ computation whereas explicitly building and multiplying by G^* is $\mathcal{O}(n^4)$. Moreover, if J and ΔP are sparse with respect to a graph, then these computations can be implemented very efficiently, requiring $\mathcal{O}(n)$ computations in graphs with bounded degree.

However, it must be noted that the preceding specifications for G^* are only valid in the full Gaussian family. In particular, the Fisher information $\hat{G}^*$ of the Gauss-Markov family on $\mathcal{G}$ is *not* simply a sub-matrix of the full G^*, but rather is the corresponding Schur complement:

$$\hat{G}^* = G^*_{\mathcal{G},\mathcal{G}} - G^*_{\mathcal{G},\backslash\mathcal{G}}(G^*_{\backslash\mathcal{G},\backslash\mathcal{G}})^{-1}G^*_{\backslash\mathcal{G},\mathcal{G}}$$

For a sparse J matrix, we see that the full G^* becomes sparse. However, due to fill in the Schur complement, the matrix $\hat{G}^*$ will typically become a full matrix.[1] Nonetheless, the fast algorithm for G^* may still be useful as an approximate preconditioner in iterative methods (by approximating $\hat{G}^*$ by the submatrix $G^*_{\mathcal{G},\mathcal{G}}$, which neglects the "fill" term in the Schur complement).

[1]Although, we show an important exception to the rule is shown in the following section.

4 Chordal Graphs and Junction Trees

In this section we consider some specialized algorithms which implement Fisher information operations efficiently in Gauss-Markov models defined on chordal graphs.

We recall that a graph is *chordal* if for each cycle of length four or more there exists a corresponding *chord*, an edge linking two non-consecutive vertices of the cycle. One equivalent characterization of a chordal graph is that is has a junction tree. Let $\mathcal{C}$ denote the set of cliques of the graph where a *clique* is any completely connected subsets of vertices. A tree $T = (\mathcal{C}, E_T)$, formed by linking cliques of the graph, is called a *junction tree* if for any two cliques C_α and C_β, every clique along the unique path in T from C_α to C_β is contained in the intersection of the endpoints $C_\alpha \cap C_\beta$.

Given a junction tree of the graph, we also define a corresponding set of *separators* $\mathcal{S}$, one for each edge (C_α, C_β) of the junction tree, defined as the intersection of the endpoints of the edge $S_{\alpha,\beta} = C_\alpha \cap C_\beta$. Incidentally, these are also separators in the original chordal graph. In fact, the set of conditional independence relations implied by the chordal graph $\mathcal{G}$ is exactly equivalent to those implied by the the junction tree (viewed as a Markov-tree model).

4.1 Sparsity of $G^*(\eta)$ on Chordal Graphs

Given a probability distribution $p(x)$ which is Markov on a chordal graph $\mathcal{G}$, we may express the joint probability distribution as the product of clique marginals divided by the product of separator marginals:

$$p(x) = \frac{\prod_{C \in \mathcal{C}} p_C(x_C)}{\prod_{S \in \mathcal{S}} p_S(x_S)}$$

The entropy submits to a similar decomposition in terms of the marginal entropies on the cliques and separators of the graph.

$$h(\eta) = \sum_C h_C(\eta_C) - \sum_S h_S(\eta_S)$$

Differentiating with respect to moment parameters and negating the result we obtain:

$$\Lambda^{-1}(\eta) = \sum_C \Lambda_C^{-1}(\eta_C) - \sum_S \Lambda_S^{-1}(\eta_S)$$

Here, we have used the property that $\nabla h(\eta) = \Lambda^{-1}(\eta)$, both for the global entropy and each of the marginal entropies. This relates the global map Λ^{-1} to local mappings which have a closed form solution. In the Gaussian model, this is equivalent to the computation:

$$J = \sum_C (P_C)^{-1} - \sum_S (P_S)^{-1}$$

where the terms on the right are appropriately zero-padded to be consistent with J. Thus, evaluation of $\theta = \Lambda^{-1}(\eta)$ has a simple closed form solution in chordal graphs.

If we differentiate again, we obtain a corresponding decomposition of the Fisher information matrix $G^*(\eta) = D\Lambda^{-1}(\eta) = -\nabla^2 h(\eta)$ in terms of marginal Fisher information terms over the cliques and separators of the graph:

$$G^*(\eta) = \sum_C G_C^*(\eta_C) - \sum_S G_S^*(\eta_S)$$

Using the fact that the marginal Fisher informations in moment parameters are inverses of the marginal Fisher information in exponential parameters, and those are submatrices of $G(\theta)$, we obtain:

$$G^* = \sum_C (G_C)^{-1} - \sum_S (G_S)^{-1}$$

Note, the relationship between G^* and G is analogous to that between J and P. This relation arises because the matrix G^* is itself a symmetric, positive-definite matrix defined over a chordal graph (an augmented version of $\mathcal{G}$ with vertex set $V \cup E$, corresponding to statistics ϕ, and with edges linking any two statistics that have support inside a common clique of $\mathcal{G}$).

Based on our earlier analysis of the Fisher information in the full Gaussian family, we can provide explicit formula for those marginal terms. Moreover, the resulting matrix is sparse, reflecting the same sparsity structure as in the underlying graph. Perhaps more importantly, we can now efficiently compute $\Delta\eta = G^*(\eta)\Delta\theta$ which, in matrix form, can be expressed as:

$$\Delta J = -\sum_C P_C^{-1} \Delta P_C P_C^{-1} + \sum_S P_S^{-1} \Delta P_S P_S^{-1}$$

This is an $\mathcal{O}(nw^3)$ computations where w is the width of the graph. This is more efficient than explicitly computing the sparse matrix $G^*(\eta)$ and performing the matrix-vector product $G^*(\eta)\Delta\eta$, which would require $\mathcal{O}(nw^4)$ operations (although, in graphs with very low width, the latter method may still be acceptable).

Similar as before, given a non-chordal graph $\mathcal{G}_{nc}$, we can express its Fisher information matrix (of the moment parameterization) as the Schur complement of the Fisher information of any chordal graph $\mathcal{G}_c$ that contains $\mathcal{G}_{nc}$ as a subgraph. This suggests that the preceding fast implementations of Fisher information for chordal graphs may serve as a useful approximation to the Fisher information of its embedded subgraphs.

4.2 Recursive Inference on Junction Trees

Let $T = (\mathcal{C}, E_T)$ be a junction tree of a chordal graph. We obtain a directed version of T by selecting an arbitrary clique as the root of the tree and orienting the edges away from this root node. For a given node α of the junction tree, let $\pi(\alpha)$ denote the parent of α. At each non-root node α, we split the corresponding clique C_α into disjoint subsets $S_\alpha = C_\alpha \cap C_{\pi(\alpha)}$ and $R_\alpha = C_\alpha \setminus C_{\pi(\alpha)}$. At the root node we define these as $S_\alpha = \emptyset$ and $R_\alpha = C_\alpha$.

We specify our recursive inference procedure in two passes: an “upward” leaf-to-root pass and a “downward” root-to-leaf pass. The input to this procedure is the sparse J matrix defined over a chordal graph. The output is a sparse P matrix, which contains the corresponding subset of elements in the covariance matrix.

Upward Pass The upward pass begins at the leaves of the tree and works its way up the tree performing the following computations:

$$\begin{aligned} Q_\alpha &= (J_{R_\alpha,R_\alpha})^{-1} \\ A_\alpha &= -Q_\alpha J_{R_\alpha,S_\alpha} \\ J_{S_\alpha,S_\alpha} &\leftarrow J_{S_\alpha,S_\alpha} + J_{S_\alpha,R_\alpha} A_\alpha \end{aligned}$$

In the last step, the principle submatrix of J indexed by S_α is overwritten. This serves to propagate information to the parent of node α in the junction tree. These computations are clearly equivalent to Gaussian elimination in the J matrix:

$$J_{S_\alpha,S_\alpha} \leftarrow J_{S_\alpha,S_\alpha} - J_{S_\alpha,R_\alpha} J_{R_\alpha,R_\alpha}^{-1} J_{R_\alpha,S_\alpha}$$

Thus, when we compute Q_α at non-leaf nodes, the value of J_{R_α,R_α} used above is the result of already having eliminating the descendents of node α.

We store the intermediate computations A_α and Q_α at each node of the tree as these can be reused in the following downward pass. In fact, these parameters may be interpreted as specifying the equivalent directed forward model:

$$x_{R_\alpha} = A_\alpha x_{S_\alpha} + w_\alpha$$

where w_α is zero-mean Gaussian with covariance Q_α. Essentially, the upward pass may be viewed as reparameterizing the joint distribution in the form:

$$\begin{aligned} p(x) &= \prod_\alpha p(x_{R_\alpha}|x_{S_\alpha}) \\ p(x_{R_\alpha}|x_{S_\alpha}) &\propto \exp[-\tfrac{1}{2}(x_{R_\alpha} - A_\alpha x_{S_\alpha})^T Q_\alpha^{-1}(x_{R_\alpha} - A_\alpha x_{S_\alpha})] \end{aligned}$$

It is a simple exercise to verify that this yields an equivalent information form.

Downward Pass The downward pass begins at the root of the tree and works it way back down the tree to the leaves performing the following computations in the order shown:

$$\begin{aligned} P_{R_\alpha,S_\alpha} &\leftarrow A_\alpha P_{S_\alpha,S_\alpha} \\ P_{S_\alpha,R_\alpha} &\leftarrow (P_{S_\alpha,R_\alpha})^T \\ P_{R_\alpha,R_\alpha} &\leftarrow P_{R_\alpha,S_\alpha} A_\alpha^T + Q_\alpha \end{aligned}$$

If a non-redundant symmetric storage format is used for P (e.g., only the "upper" triangular part of P is actually stored), the second step above can be omitted. This iteration computes the subset of elements of $P = J^{-1}$ corresponding to the vertices and edges of the chordal graph. The downward pass simply propagates covariances in a causal fashion according to the forward model:

$$P_{R_\alpha,R_\alpha} = p[x_{R_\alpha} x_{R_\alpha}^T] = p[(A_\alpha x_{S_\alpha} + w_\alpha)(A_\alpha x_{S_\alpha} + w_\alpha)^T] = A_\alpha P_{S_\alpha,S_\alpha} A_\alpha^T + Q_\alpha$$

$$P_{R_\alpha,S_\alpha} = p[x_{R_\alpha} x_{S_\alpha}^T] = p[(A_\alpha x_{S_\alpha} + w_\alpha) x_{S_\alpha}^T] = A_\alpha P_{S_\alpha,S_\alpha}$$

Note that this form of recursive inference only requires one matrix inverse per node in the junction tree. We also comment that the entire computation can be performed "in place" in the sparse J matrix (by over-writing the elements of J by the corresponding values of P in the downward pass) so that we do not have to simultaneously provide storage for two sparse matrices.

Differential Propagation There also is a linearized version of this algorithm which computes the first-order perturbation ΔP as a result of a change of ΔJ in the information matrix. The input to the procedure is a pair of sparse matrices J and ΔJ both defined over a chordal graph. In the differential version of the algorithm, we perform the following computations (in addition to those computations already listed above).

Upward Pass:

$$\begin{aligned} \Delta Q_\alpha &= -Q_\alpha \Delta J_{R_\alpha,R_\alpha} Q_\alpha \\ \Delta A_\alpha &= -(Q_\alpha \Delta J_{R_\alpha,R_\alpha} + \Delta Q_\alpha J_\alpha) \\ \Delta J_{S_\alpha,S_\alpha} &\leftarrow \Delta J_{S_\alpha,S_\alpha} + \Delta J_{S_\alpha,R_\alpha} A_\alpha + J_{S_\alpha,R_\alpha} \Delta A_\alpha \end{aligned}$$

Downward Pass:

$$\begin{aligned} \Delta P_{R_\alpha,S_\alpha} &\leftarrow \Delta A_\alpha P_{S_\alpha,S_\alpha} + A_\alpha \Delta P_{S_\alpha,S_\alpha} \\ \Delta P_{R_\alpha,R_\alpha} &\leftarrow \Delta P_{R_\alpha,S_\alpha} A_\alpha^T + P_{R_\alpha,S_\alpha} \Delta A_\alpha^T + \Delta Q_\alpha \end{aligned}$$

Upon completion, this computes the perturbations in the moment parameters defined on the chordal graph.

This linearized computation of ΔP over a chordal graph given J and ΔJ is equivalent to computing $\Delta\eta = G(\theta)\Delta\theta$ given θ and $\Delta\theta$ in the corresponding exponential family representation of the Gauss-Markov family defined on that chordal graph. Thus, these recursive algorithms may be viewed as an efficient method to multiply by the Fisher information matrix $G(\theta)$. The complexity of this algorithm is $\mathcal{O}(nw^3)$, where w is the width of the chordal graph (the size of the largest clique). Note that $G(\theta)$ is typically a full $d \times d$ matrix (where $d = |V| + |E|$), hence explicit evaluation of $G(\theta)\Delta\eta$ would require $\mathcal{O}(d^2)$ computations.