Math. Program., Ser. A 93: 1–25 (2002)

Digital Object Identifier (DOI) 10.1007/s101070100281

Andreas Griewank · Christo Mitev

Detecting Jacobian sparsity patterns by Bayesian probing

Received: April 29, 2000 / Accepted: September 2001
Published online April 12, 2002 – © Springer-Verlag 2002

Abstract. In this paper we describe an automatic procedure for successively reducing the set of possible nonzeros in a Jacobian matrix until eventually the exact sparsity pattern is obtained. The dependence information needed in this probing process consist of "Boolean" Jacobian-vector products and possibly also vector-Jacobian products, which can be evaluated exactly by automatic differentiation or approximated by divided differences. The latter approach yields correct sparsity patterns, provided there is no exact cancellation at the current argument.

Starting from a user specified, or by default initialized, probability distribution the procedure suggests a sequence of probing vectors. The resulting information is then used to update the probabilities that certain elements are nonzero according to Bayes' law. The proposed probing procedure is found to require only $O(\log n)$ probing vectors on randomly generated matrices of dimension n, with a fixed number of nonzeros per row or column. This result has been proven for (block-) banded matrices, and for general sparsity pattern finite termination of the probing procedure can be guaranteed.

Key words. sparsity – automatic differentiation – difference quotients – conditional probability

1. Introduction and notation

Many numerical methods require the evaluation of Jacobians for vector functions given as evaluation procedures. If known, the sparsity pattern of these first derivative matrices can be used to evaluate [CM84], store, and manipulate them more efficiently. Especially on discretizations of differential equations and other large scale problems, detecting and specifying the pattern of potentially nonzero entries may be a laborious and error prone task. Therefore we investigate in this paper ways of determining the pattern automatically, without incurring the overhead of a dynamically sparse implementation of the chain rule like SparsLinC [BCK95].

1.1. Algorithmic differentiation background

Apart from computer algebra systems, which can be expected to work only for rather small problems, one may apply *automatic*, or *algorithmic* differentiation (AD) to conveniently and reliably determine sparsity pattern. AD is based on the simple observation that in the computational practice all functions are computed via a sequence of scalar assignments of the form

$$v = a(u, w) \equiv u \circ w \qquad \text{or} \qquad v = \varphi(u) \quad . \tag{1}$$

A. Griewank: Institute of Scientific Computing, Technical University Dresden, D-01062 Dresden,
e-mail: griewank@math.tu-dresden.de

C. Mitev: Treppenmeister GmbH, Emminger Straße 38, D-71131 Jettingen

Here $\circ \in \{+, -, *, /\}$ represents an arithmetic operation and φ a univariate intrinsic like the exponential or trigonometric functions. The partial derivatives a_u, a_w and φ_u of all such elemental functions can be easily generated symbolically and then evaluated at the current argument. The resulting floating point numbers can subsequently be combined using some version of the chain rule to yield the partial derivatives of all dependent variables, denoted here by y_i for $i = 1 \dots m$, with respect to all independent variables, denoted by x_j for $j = 1 \dots n$ (for details see [Gri00] and [CF$^+$01]).

In the so-called forward mode of algorithmic differentiation one propagates intermediate gradients $\dot{v} \equiv \nabla_x v \equiv \partial v/\partial x \in \mathbb{R}^n$ according to the simple rules

$$\dot{v} = a_u * \dot{u} + a_w * \dot{w} \qquad \text{and} \qquad \dot{v} = \phi_u * \dot{u} \quad . \tag{2}$$

The resulting computational effort for evaluating $y = F(x)$ and its Jacobian $F'(x) = (\nabla y_i^T)_{i=1\dots m} \in \mathbb{R}^{m\times n}$ with respect to x is bounded by a small multiple of n times the effort of evaluating y by itself. Instead of propagating the full gradient vectors $\nabla_x v$ at once, one may restrict the chain rule application to the partials with respect to one particular independent variable so that in the recursions above $\dot{v} \equiv \partial v/\partial x_j \in \mathbb{R}$ for some fixed j. The resulting partials $\dot{y}_i$ for $i = 1 \dots m$ then form the jth column of the Jacobian matrix. One may also propagate partials with respect to a subset of p independents and then obtains the corresponding p columns of the Jacobian at a time.

Since $\nabla_x x_j \equiv e_j \in \mathbb{R}^n$ is by definition a Cartesian basis vector, the gradients ∇v_x of intermediates v that occur early in the evaluation process tend to be rather sparse. Moreover, excluding incidental cancellations, sparsity must occur throughout the calculation if the Jacobian itself is sparse. The first AD tool that facilitates the automatic detection and exploitation of Jacobian sparsity was the SparsLinC [BCK95] module of ADIFOR. It keeps all $\nabla_x v$ in a sparse vector format, an approach that is very flexible but incurs a significant runtime overhead at each new evaluation point (for specific runtimes see [BK$^+$97]). Recently the packages ADOL-C [GJ$^+$96] and TAMC [GK00] have acquired the capability to propagate the sparsity pattern

$$\hat{v} \quad \equiv \quad \left(\text{sign}\left|\partial v/\partial x_j\right|\right)_{j=1\dots n} \in \{0, 1\}^n$$

as bit-vectors. This approach avoids the manipulation of index sets and requires roughly $1 + n/64$ times as much storage and run-time as the original function itself, if the whole Jacobian sparsity pattern is to be evaluated at once. In this paper we aim to avoid the proportional dependence on n altogether by successively collecting and incorporating information about the sparsity pattern. The factor 64 occurs assuming that the original function evaluation is carried out in double precision reals, each of which takes as much space as two unsigned integers representing 32 bits. The bit patterns can be propagated according to the rules

$$\hat{v} = \hat{u} \,|\, \hat{w} \quad \text{or} \quad \hat{v} = \hat{u} \tag{3}$$

which follow from (2) for $a_u \neq 0 \neq a_w$ or $\varphi_u \neq 0$ with $|$ representing a bitwise or. Again, the same recursions apply if the $\dot{v}$ and thus the $\hat{v}$ are restricted to a subset of components. In particular, if $\hat{v}$ is a pair of *Boolean* integers with 32 bits each, the bitwise or (and of course mere copying) requires on modern computer chips only a couple of machine cycles, just like floating point additions and multiplications.

In the study [GK01] it was found on the five problems FDC, FIC, IER, SFO, and SFI from the Minpack-2 test collection [AC$^+$92] that propagating integers was never slower and sometimes considerably faster than propagating doubles representing directional derivatives. Therefore we assume throughout the paper, that at the cost of evaluating one column of the Jacobian by forward automatic differentiation in double precision arithmetic, one may determine the sparsity pattern of at least 64 columns by bit-pattern propagation. In the reverse mode, one obtains analogous results for the transpose, so that one row of the Jacobian is about as expensive to obtain as the sparsity pattern of 64 of them. Of course one can also propagate just one bit representing the dependence on a single independent variable per intermediate value, either forward or backward. However, that won't be much cheaper than doing 32 of them simultaneously in either mode. In any case one should keep in mind, that either one of these efforts is a small multiple of the cost to evaluate the function $y = F(x)$ by itself.

These basic observations regarding computational costs are the only results from AD that are needed in the remainder of this paper. Moreover, our probing approach is also applicable to differencing, a simple technique that might be unavoidable, for example when source code for evaluating the function is not available.

1.2. Sparsity patterns and Boolean probes

Throughout the paper we assume that some vector function

$$F \equiv (F_i)_{i=1\ldots m} : \mathbb{R}^n \mapsto \mathbb{R}^m$$

is continuously differentiable on some neighbourhood $\mathcal{D}$ of a given vector argument $x \equiv (x_j)_{j=1\ldots n} \in \mathbb{R}^n$. Then we may define the *sparsity pattern* of the Jacobian

$$F'(x) \equiv \frac{\partial}{\partial x} F(x) \in \mathbb{R}^{m\times n}$$

near x as the Boolean matrix $S \equiv (s_{ij})_{j=1\ldots n}^{i=1\ldots m} \in \{0,1\}^{m\times n}$ with

$$s_{ij} = 0 \quad \Longleftrightarrow \quad \partial F_i(x)/\partial x_j \equiv 0 \quad \text{at all } x \in \mathcal{D}$$

The key assumption for the application of the techniques to be developed in this paper is the following. Given any vector $t \in \{0,1\}^n$ we must be able to evaluate the Boolean vector $r \equiv (r_i)_{i=1\ldots m} \equiv S\, t \in \{0,1\}^m$ with

$$r_i = 1 \quad \Longleftrightarrow \quad s_{ij} = 1 = t_j \quad \text{for at least one } j \le n \quad . \tag{4}$$

We will refer to the pair (t, r) or more generally to any other partial information that we can gather about the unknown matrix S as a "probe". A related but different concept of probing is used in domain decomposition methods to precondition matrices with a large number of very small entries (see [BSG96] and the original proposals [CP$^+$74] and [NR83]).

The "Boolean product" r might be evaluated directly by propagating suitable dependence information through a given evaluation procedure for F, as sketched in the previous Subsect. 1.1. If no AD software with this capability is available, one may base our probing method on the following consequence of the mean value theorem

$$F_i(x) \neq F_i(x + \varepsilon\, t) \quad \Longrightarrow \quad r_i = 1 \quad . \tag{5}$$

Here ε is assumed nonzero but small and t is interpreted as a real 0–1 vector in this context. Unfortunately, the converse implication is not strictly true as the inequality on the left may happen to be violated, yet r_i may still be equal to 1. However, for generic $x \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}$ such exact cancellations are extremely unlikely to occur, and determining the m components of r on the basis of the difference $F(x + \varepsilon t) - F(x)$ is probably as safe as any other calculation one performs in floating point arithmetic. In either case, we will be able to detect sparsity exclusively on the basis of the qualitative dependence information represented by r as a function of t and possibly corresponding transpose probes as defined below.

Apart from its greater reliability the key advantage of automatic differentiation for sparsity probing is that one may also evaluate Boolean products of the transpose Jacobian. Using the reverse mode one may obtain for a given probing vector $\bar{t} \in \{0, 1\}^m$ the result $\bar{r} = S^T \bar{t} \in \{0, 1\}^n$ with

$$\bar{r}_j = 1 \quad \Longleftrightarrow \quad s_{ij} = 1 = \bar{t}_i \quad \text{for at least one} \quad i \leq m \quad . \tag{6}$$

This kind of transposed dependence information is extremely useful when the Jacobian has dense rows, in which case one would need to let t range over all Cartesian basis vectors $e_j \in \{0, 1\}^n$ for $j = 1 \ldots n$, in order to completely determine the sparsity pattern S on the basis of direct products $r = S\, t$ alone. Conversely, when the pattern has a dense column, one would need transpose products with $\bar{t}$ ranging over all Cartesian basis vectors in $\{0, 1\}^m$, if direct Boolean products were not available for some reason. Of course, similar effects occur when a Jacobian has nearly dense rows or columns, so that in general only a combination of direct and transpose Boolean products allows the determination of the sparsity pattern S with a reasonable number of probes.

In [GR02] it is shown that minimizing the number of probes needed to find all zeros and all nonzeros in a given sparsity pattern leads to NP-complete optimization problems. These results justify the development and application of heuristic strategies in the remainder of the paper. It is also proven in [GR02] that, provided all m rows and n columns in a sparsity pattern are distinct, the minimal number of direct and transpose probes required for its determination is bounded below by $\log_2 [\max(m, n)]$. As it turns out, the heuristic procedure developed in this paper requires only a small multiple of this lower bound whenever the number of nonzeros per row or column is uniformly bounded. For banded matrices, such logarithmic probing complexities have been established theoretically in [GM01]. These results can be extended to standard discretization matrices and other Kronecker products of banded matrices.

1.3. Structure of the paper

The paper is organized as follows. In Sect. 2 we derive formulas for updating the conditional probabilities that a certain entry s_{ij} equals 1, on the basis of the information gained from one or several Boolean products. We also develop an uncertainty function U that represents the expected number of not yet identified zeros at the current stage of the probing procedure. Finally, we compute the reduction of U that can be expected for a certain test vector t or $\bar{t}$ on the basis of the current probabilities. In Sect. 3 we present a class of optimization procedure that is shown to always achieve $U = 0$ and thus to determine all zeros and nonzeros in a finite number of probes. This analysis does not rely on the stochastic derivations in Sect. 2, which may therefore be treated as mere heuristics for designing the proposed scheme. At the end of Sect. 3, we discuss various greedy strategies for computing suitable test vectors t and $\bar{t}$, either one by one or in bundles. In Sect. 4 we present experimental results obtained on some regular and some random sparsity patterns, as well as six matrices from the Harwell-Boeing collection [DGL]. The final Sect. 5 contains some tentative conclusions and ideas for further improvements of the probing approach.

2. Derivation as Bayesian update

The problem at hand is related to dynamic searches for randomly distributed objects in a given number of boxes (see e.g., [KS79] and [AZ85]). However, the results published in the literature are not directly applicable because we cannot simply look into any one of the boxes, i.e. matrix entries, but rather have to base our procedure on indirect information in the form of Boolean products. With $S = \left(s_{ij}\right)_{i=1\ldots m}^{j=1\ldots n}$ now a matrix of stochastic Boolean variables let

$$p_{ij}^0 \equiv P(s_{ij} = 1) \in [0, 1]$$

denote the probability that the ijth Jacobian element does not vanish identically. By suitable initialisation and appropriate updating, we must ensure that all probabilities p_{ij}^k computed satisfy the following consistency condition

$$p_{ij}^k = 0 \implies s_{ij} = 0 \quad \text{and} \quad p_{ij}^k = 1 \implies s_{ij} = 1 \quad . \tag{7}$$

Provided we consider the individual elements s_{ij} as stochastically independent, any initial matrix $P_0 \equiv (p_{ij}^0) \in [0, 1]^{m\times n}$ determines a probability distribution on the finite set of 0–1 matrices with the given dimensions m and n. This set Ω_0 has $2^{m\,n}$ elements, except when some elements s_{ij} are a priori known so that the corresponding p_{ij}^0 can be initialized to zero or one.

2.1. *Conditional probabilities*

Every subsequent probe restricts the set of possible sparsity patterns so that we generate in fact a descending chain of feasible sets

$$\Omega \equiv \{0, 1\}^{m\times n} \supset \Omega_0 \supset \Omega_1 \supset \cdots\cdots \supset \Omega_{k-1} \supset \Omega_k \supset \cdots \quad .$$

If there is enough time and interest, this process can be continued until Ω_k has been reduced to a singleton, at which point the sparsity pattern is completely determined.

Associated with each restriction of the feasible set Ω_k is a change in the conditional probabilities

$$p_{ij}^k \;\equiv\; P(\{s_{ij} = 1\} \big| S \in \Omega_k) \;\equiv\; P(\{s_{ij} = 1\} \cap S \in \Omega_k)/P(S \in \Omega_k) \quad . \tag{8}$$

Unfortunately, the restrictions of the Boolean random variables s_{ij} to the feasible sets $\Omega_k \neq \Omega$ for $k > 0$ are in general no longer stochastically independent.

2.2. *Bounding patterns and expected discrepancies*

Nevertheless, suppose for the moment we have some way of computing or approximating the matrices $P_k = (p_{ij}^k) \in [0, 1]^{m\times n}$. Then we may define at each stage the bounding sparsity pattern

$$S_k \;\equiv\; \left(s_{ij}^k\right) \;=\; \mathrm{sign}(P_k) \;=\; \left(\mathrm{sign}\left(p_{ij}^k\right)\right) \;\in \{0, 1\}^{m\times n} \quad .$$

In other words, we take the zeros that have been definitely verified and consider all other entries as nonzeros. This is a conservative estimate of the sparsity pattern and as more and more zeros are found, we obtain in the componentwise partial ordering of matrices the descending chain

$$\{1\}^{m\times n} \geq S_0 \geq S_1 \geq \cdots\cdots \geq S_{k-1} \geq S_k \geq \cdots \geq S \geq \{0\}^{m\times n} \quad .$$

Moreover, denoting by $\#(S_k - S)$ the number of of nonzero entries in the nonnegative discrepancy $S_k - S$, we can compute its expected value at each stage as follows.

Lemma 1. *With p_{ij}^k given by (8) we have the conditional expectation*

$$U(P_k) \;\equiv\; E(\#(S_k - S) \,\big|\, S \in \Omega_k) \;=\; \sum_{p_{ij}^k > 0} \left(1 - p_{ij}^k\right) \tag{9}$$

$$= \sum_{i,j} \left(1 - p_{ij}^k\right) - \left|\left\{ (i, j) : p_{ij}^k = 0 \right\}\right| \;\leq\; m\,n \tag{10}$$

provided the consistency condition (7) is satisfied.

□

Proof. The bound S_k is constant and S is restricted to the event set Ω_k, where its individual elements s_{ij} equal 1 with probability p_{ij}^k. If this conditional probability is positive, the bounding element s_{ij}^k must be 1 by (7). Hence, the corresponding element of the discrepancy $S_k - S$ is only nonzero (namely equal to 1) when $s_{ij} = 0$, which happens with probability $(1 - p_{ij}^k)$. If on the other hand $p_{ij}^k = 0$, it follows from the consistency condition (7) that s_{ij} must also vanish so that there is no contribution to the expected discrepancy. This completes the proof.

□

The function $U(P_k)$ represents the expected number of zeros in the actual sparsity pattern S that are still overlooked if we replace it with the current conservative bound S_k. We may use this objective function to gauge the remaining indeterminacy in P_k without knowing S itself. Of course, the value $U(P_k)$ does strongly depend on the a priori probability distribution P_0, for which we will develop a very simple default setting in Subsect. 2.7. As we will see, the measure of uncertainty $U(P_k)$ can be monotonically reduced by our probing procedure, even when the result vectors r or $\bar{r}$ do not contain a single zero. $U(P_k)$ is a loss function in the sense of the survey [Lin95], where a much more general framework for the design of sequential experiments is discussed.

2.3. Sequential updating

Strictly speaking, one should consider the totality of all probing results obtained up to a certain stage k as one composite probe, and compute the conditional probabilities $p_{ij}^k \equiv P(\{s_{ij} = 1\} \,|\, S \in \Omega_k)$ accordingly. However, the memory requirement and computational cost for keeping and updating a joint multivariate probability distribution is rather high. In this paper we use a sequential simplification where the elements of P_k are updated after each probe and then treated as the a priori probability distribution for the next stage of the probing process. Hence we will drop the index k and write instead p_{ij} and p'_{ij} for the a priori and a posteriori probabilities.

To illustrate the effects of this sequential approach, let us consider the case $m = 1$ with $n = 3$ and uniform initial probabilities $p_j \equiv p_{1j} = \frac{1}{2}$ for $j = 1, 2, 3$. Suppose the probing directions $t_1 = (1, 1, 0)$ and $t_2 = (0, 1, 1)$ yield the results $r_1 = 1$ and $r_2 = 0$, respectively. Updating the vector $P = (1/2, 1/2, 1/2)$ according to Bayes' formula (see equation (15) below), we obtain first $P' = (2/3, 2/3, 1/2)$ and subsequently $P'' = (2/3, 0, 0)$. If however, we incorporate the probing results in the opposite order we obtain the results $P' = (1/2, 0, 0)$ and $P'' = (1, 0, 0)$. Thus we see that in the second case the sparsity patter has already been completely determined, whereas in the first case, one more probe is needed to identify the first element as a nonzero. If the result of the probe in the direction t_2 was also $r_2 = 1$, the final probability distribution P'' would be either $(2/3, 4/5, 3/5)$ or the reversal $(3/5, 4/5, 2/3)$, again depending on the order in which the results were incorporated. The correct conditional probabilities given the two probes with results $r_1 = 1 = r_2$, form the symmetric distribution $(3/5, 4/5, 3/5)$.

In general, one finds that updating the probabilities sequentially may perturb nonzero probabilities, but all zeros are properly determined. Since the latter are the most important pieces of information, the sequential approach seems to be a sensible idea. However, one must be aware that the sequential approach may still yield a positive value of the remaining uncertainty $U(P)$ when a simultaneous analysis might already imply $U(P) = 0$, in which case the process could finish even under this most stringent stopping criterion.

2.4. Expected reductions for general probes

Suppose we perform on S some arbitrary probe that has certain mutually exclusive result events $E^{(q)}$ for $q = 1 \ldots Q$. Let $e^{(q)} > 0$ denote their a priori probabilities, which must sum to one. Then we have by Bayes' formula for each s_{ij} the conditional probability

$$p_{ij}^{(q)} \equiv P\big(\{s_{ij} = 1\} \cap E^{(q)}\big)/e^{(q)} \quad . \tag{11}$$

Denoting the Q possible new probability matrices by $P^{(q)}$ and weighting them by the a priori probability of event $E^{(q)}$, we obtain the following results.

Lemma 2. *The expected reduction of U is given by*

$$U(P) - \sum_q e^{(q)}\, U(P^{(q)}) \quad = \quad \sum_q e^{(q)} \,\big|\, \big\{(i, j) : p_{ij}^{(q)} = 0 \neq p_{ij}\, \big\} \,\big| \quad .$$

□

Proof. The expected value of the uncertainty after the update is by Lemma 1

$$\begin{aligned} \sum_q e^{(q)} U(P^{(q)}) \; &= \; \sum_{i,j} \sum_q e^{(q)}\big(1 - p_{ij}^{(q)}\big) \; - \; \sum_q e^{(q)} \,\big|\, (i, j) : p_{ij}^{(q)} = 0 \,\big| \\ = \sum_{i,j} \sum_q \big[\, e^{(q)} - P\big(\{s_{ij} = 1\} \cap E^{(q)}\big) \,\big] \; &- \; \sum_q e^{(q)} \,\big|\, (i, j) : p_{ij}^{(q)} = 0 \,\big| \\ = \sum_{i,j} (1 - p_{ij}) \; - \; \sum_q e^{(q)} \,\big|\, (i, j) &: p_{ij}^{(q)} = 0 \,\big| \\ = U(P) \; - \; \sum_q e^{(q)} \,\big|\, (i, j) &: p_{ij}^{(q)} = 0 \neq p_{ij} \,\big| \end{aligned}$$

where we have used in the last equation that the $E^{(q)}$ form a fundamental system of events so that

$$\sum_q P\big(\{s_{ij} = 1\} \cap E^{(q)}\big) \; = \; P(\{s_{ij} = 1\}) \; = \; p_{ij} \quad .$$

Subtracting this new value from the previous uncertainty $U(P)$ and using again $\sum_q e^{(q)} = 1$ we obtain the assertion.

□

On the right hand side of the identity asserted by Lemma 1, we find the sum over the number of new zeros revealed in the event $E^{(q)}$ multiplied by its probability $e^{(q)}$. Hence we may draw the not very surprising conclusion that the more zeros a probe is likely to reveal, the better it is for the reduction of our uncertainty measure $U(P)$.

2.5. *Expected reduction for Boolean probes*

For a single probe with $t \in \{0, 1\}^n$, there are $Q = 2^m$ events $E^{(q)}$ that correspond to all possible outcomes $r = S\,t \in \{0, 1\}^m$. Assuming that the Boolean entries in the sparsity pattern S viewed as stochastic variables are mutually independent, we may break U down into a sum of the row-wise contributions

$$U_i(P) = \sum_j (1 - p_{ij}) - |\{j : p_{ij} = 0\}| \quad .$$

Concerning the ith row, there is only one event that possibly introduces any new zeros, namely the result $r_i = 0$, whose prior probability is given by

$$w_i(\mathcal{J}) \;\equiv\; P(r_i = 0) \;\equiv\; \prod_{j \in \mathcal{J}} (1 - p_{ij}) \quad , \tag{12}$$

where $\mathrm{supp}(t) = \mathcal{J}$. Since already known zeros can be left completely out of the picture, only the index set

$$\mathcal{J}^i \equiv \{j \in \mathcal{J} : p_{ij} > 0\} \quad \text{of size} \quad l_i \equiv l_i(\mathcal{J}) \equiv |\mathcal{J}^i| \tag{13}$$

is of interest in the ith row. Thus the expected reduction of U_i is $w_i\, l_i$ and we sum over i to obtain by Lemma 2 the total expected reduction

$$\Delta U(P, \mathcal{J}) \;\equiv\; \sum_i w_i(\mathcal{J})\, l_i(\mathcal{J}) \quad . \tag{14}$$

After the probe defined by $\mathcal{J}$ the probabilities are updated according to the Bayesian formula (11), which takes the specific form

$$p'_{ij} = \begin{cases} p_{ij} & \text{if } j \notin \mathcal{J} \\ 0 & \text{if } j \in \mathcal{J} \quad \text{and} \quad r_i = 0 \\ p_{ij}/(1 - w_i) & \text{if } j \in \mathcal{J} \quad \text{and} \quad r_i = 1 \end{cases} \quad . \tag{15}$$

In other words, the probabilities in columns that do not belong to $\mathcal{J}$ stay unchanged. Where $r_i = 0$, all s_{ij} with $j \in \mathcal{J}$ are determined as zeros, and we have naturally $p'_{ij} = 0$. Where $r_i = 1$, all nonzero probabilities p_{ij} with $j \in \mathcal{J}$ move some way toward 1, which is reached exactly only in the rather special situation $l_i(\mathcal{J}) = 1$, where a single s_{ij} is identified as 1. In Subsect. 2.8 and Subsect. 3 we will discuss greedy methods for finding index sets $\mathcal{J}$ with a large expected reduction $\Delta U(P, \mathcal{J})$. In any case, our probing method will be of the following form

0) Initialize P in agreement with (7).
1) Select a $\mathcal{J} \subseteq [1 \ldots n]$ maximizing $\Delta U(P, \mathcal{J})$.
2) Perform the probe $r = S\,t$ with $\mathrm{supp}(t) = \mathcal{J}$.
3) Update P according to (15).
4) Continue with **1)** until $U(P) = 0$.

For notational simplicity, the boxed procedure is formulated exclusively in terms of direct probes. For a transpose probe $\bar{t}$ with $\bar{\mathcal{J}} \equiv \mathrm{supp}(\bar{t})$, we obtain the expected reduction corresponding to (14)

$$\Delta U(P, \bar{\mathcal{J}}) \;=\; \sum_j \bar{w}_j(\bar{\mathcal{J}})\, \bar{l}_j(\bar{\mathcal{J}}) \quad ,$$

where naturally

$$\bar{w}_j(\bar{\mathcal{J}}) \;=\; \prod_{i \in \bar{\mathcal{J}}} \big(1 - p_{ij}\big)$$

and

$$\bar{l}_j(\bar{\mathcal{J}}) \equiv \left|\bar{\mathcal{J}}^j\right| \quad \text{with} \quad \bar{\mathcal{J}}^j \;\equiv\; \left\{i \in \bar{\mathcal{J}} : p_{ij} > 0\right\} \quad .$$

In the methods labeled *combined* later on, we perform Step 1) in each iteration twice, once to find an $\mathcal{J}$ and once to find an $\bar{\mathcal{J}}$. Depending on which index set promises a larger reduction, we perform either a direct or a transpose probe in Step 2). As we will see in Sect. 4, this combination may or may not significantly reduce the total number of probes.

2.6. *How things might go wrong*

The only contingency that may arise in the update equation (15) for the p_{ij}, is that $r_i = 1$ but $w_i = 1$ and thus $p_{ij} = 0$ for all $j \in \mathcal{J}$. This would mean that our a priori information, that all $s_{ij} = 0$ with $j \in \mathcal{J}$ vanish, is contradicted by the current probe. This could happen either because the p_{ij} were initialised so that the first part of the consistency condition (7) is violated or possibly because a previous probe based on divided difference approximations generated a faulty $r_i = 0$ due to incidental cancellations. In view of the second, remote possibility one might reinitialise all the p_{ij} with $j \in \mathcal{J}$ to some positive probability less than 1, as replacement for the indeterminate ratio $0/0$ in the update formula (15) above. A related but a little more rational approach is to initialise the probabilities p_{ij} of all elements that one expects to be zero without being absolutely certain, to some rather small value ε. Then the contingency discussed above turns into a regular update, where all the ε probabilities grow to a common value $1/l_i(\mathcal{J}) + O(\varepsilon)$, which seems sensible under the circumstances. If the second part of the consistency condition (7) is violated, no contingency in the Baysian update formulas can arise but some sparsity may go undetected forever.

2.7. *The neutral initialisation*

Normally, nothing specific is a priori known about the sparsity pattern, but we may still assume that the pattern is quite sparse and initialize all p^0_{ij} to a rather small $\varepsilon > 0$. Then $U(P_0) = n\,m(1-\varepsilon)$ and the trivial choice $\mathcal{J}_0 \equiv \{1, \dots, n\}$ or $\bar{\mathcal{J}}_0 \equiv \{1, \dots, m\}$ yield according to (14), the expected reductions

$$\begin{aligned} \Delta U(P_0, \mathcal{J}_0) &= \big(1-\varepsilon\big)^n\, n\,m = \big(1 - \; n\,\varepsilon + O(\varepsilon^2)\big)\, U(P_0) \\ \Delta U(P_0, \bar{\mathcal{J}}_0) &= \big(1-\varepsilon\big)^m\, n\,m = \big(1 - \; m\,\varepsilon + O(\varepsilon^2)\big)\, U(P_0) \quad . \end{aligned}$$

When ε is sufficiently small one can easily see that all proper subsets $\mathcal{J} \subset \mathcal{J}_0$ or $\bar{\mathcal{J}} \subset \bar{\mathcal{J}}_0$ yield a smaller expected reduction. Hence one should then take the direct probe $\mathcal{J}_0$ if $n \le m$ and otherwise the transpose probe $\bar{\mathcal{J}}_0$. More generally one may see that whenever p_{ij} is uniform, direct probing is more promising than transpose probing if $m > n$ and vise versa. This relative performance reflects the fact that a forward probe yields m pieces of information r_i, whereas a transpose probe yields n pieces of information $\bar{r}_j$. It also nicely conforms to the usual rule of thumb in automatic differentiation, namely, that the forward mode is preferable to the reverse mode if the number of independent variables is smaller than the number of dependents and vise versa. In reality, a transpose probe is somewhat more expensive, at least in terms of memory requirement, so that one may wish to bias the decision whether to take a direct or transpose probe a little in favour of the former.

Unless the sparsity pattern restricts some rows or columns to vanish identically, we must obtain for $\mathcal{J}_0$ or $\bar{\mathcal{J}}_0$ the results $r = 1 \in \{0, 1\}^m$ or $\bar{r} = 1 \in \{0, 1\}^n$ and the first Bayesian update yields correspondingly

$$p_{ij} = 1/\min(m, n) + O(\varepsilon) \quad \text{for all} \quad i, j \quad . \tag{16}$$

Hence we will refer to $p_{ij} = 1/\min(m, n)$ as the *neutral* initialization and use it as default throughout.

2.8. *Maximizing expected reductions*

To obtain a probe that promises a significant reduction in our uncertainty function $U(P)$ we wish to find a subset $\mathcal{J} \subset [1..n]$ of column indices such that the reduction $\Delta U(P, \mathcal{J})$ defined by (14) is as large as possible. This seems to be a hard combinatorial problem even though we have the simple algebraic relations

$$w_i(\tilde{\mathcal{J}} \cup \mathcal{J}) = w_i(\tilde{\mathcal{J}}) \cdot w_i(\mathcal{J}) \tag{17}$$

and

$$l_i(\tilde{\mathcal{J}} \cup \mathcal{J}) = l_i(\tilde{\mathcal{J}}) + l_i(\mathcal{J}) \tag{18}$$

provided the two column index sets $\tilde{\mathcal{J}}$ and $\mathcal{J}$ are disjoint.

Also it follows for the singletons $\mathcal{J} = \{j\}$ for $j = 1 \ldots n$ by comparison of Lemma 1 and (14) with $w_i = (1 - p_{ij})$ that

$$U(P) = \sum_{j=1}^{n} \Delta U(P, \{j\}) \quad . \tag{19}$$

Thus we conclude that by using a suitable singleton $\mathcal{J} = \{j\}$ one can always achieve an expected reduction no smaller than $U(P)$ divided by n or m for direct and transpose probing, respectively. This observation allows us to ensure finite termination of our probing scheme as shown below.

3. A finite optimization procedure

Whatever the merit of the stochastic modelling effort in the previous section, we may directly attack the successive reduction of the uncertainty functional $U(P)$ defined in Lemma 1 as a finite dimensional optimization problem. The choice of the index sets $\mathcal{J}$ then resembles the problem of finding a descent direction, but though the actual reduction is not exactly predictable, there is no need for a line-search. Moreover, for the practical reason discussed in Subsect. 1.1, we will wish to select a whole bundle of at least 32 index sets and by making them disjoint we can easily ensure that the resulting reductions are additive. It is not yet clear whether this simple strategy can be significantly improved by allowing overlaps between the index sets.

3.1. Guaranteed reduction

The following lower bound on the actual reduction immediately implies that even relatively simple choices of the $\mathcal{J}$ force the uncertainty measure U to converge towards zero.

Lemma 3. *For any possible outcome* $r = St \in \{0,1\}^m$ *with* $\mathrm{supp}(t) = \mathcal{J}$ *the matrix* P' *obtained from* P *according to* (15) *satisfies*

$$U(P) - U(P') \;\geq\; \sum_i w_i(\mathcal{J})\,\mathrm{sign}(l_i(\mathcal{J})) \;\geq\; \Delta U(P,\mathcal{J})/|\mathcal{J}| \quad ,$$

where $|\mathcal{J}| \leq n$ *denotes the cardinality of* $\mathcal{J}$ *and* $w_i(\mathcal{J})$, $l_i(\mathcal{J})$ *as in (12, 13).*

□

Proof. In the ith row we have the reduction

$$\sum_{j\in\mathcal{J}^i}(1-p_{ij}) \;\text{ if }\; r_i = 0 \quad\text{or}\quad \sum_{j\in\mathcal{J}^i} w_i\,p_{ij}/(1-w_i) \;\text{ if }\; r_i = 1 \quad ,$$

where we have used that in the second case according to (15) for each $j \in \mathcal{J}$

$$(1-p_{ij}) - (1-p'_{ij}) = p'_{ij} - p_{ij} = p_{ij}/(1-w_i) - p_{ij} = w_i\,p_{ij}/(1-w_i)\ .$$

Dividing by the expected reduction $\Delta U_i = l_i\,w_i$ we obtain in the first case by (12) and the inequality of the means

$$\frac{1}{l_i}\sum_{j\in\mathcal{J}^i}(1-p_{ij}) \Big/ \prod_{j\in\mathcal{J}^i}(1-p_{ij}) \;\geq\; \Big[\prod_{j\in\mathcal{J}^i}(1-p_{ij})\Big]^{\frac{1}{2}-1} \;\geq\; 1 \quad .$$

Hence, we have shown that in the first case $r_i = 0$ the actual reduction is bounded below by $l_i w_i$ under the tacit assumption that $\mathcal{J}^i$ is nonempty so that $l_i > 0$. If $l_i = 0$, there is no reduction in the ith row possible at all. This situation cannot arise in the second case $r_i = 1$ where the actual reduction divided by the expected reduction $l_i w_i$ is given by

$$\frac{1}{l_i}\sum_{j\in\mathcal{J}^i} p_{ij} \Big/ \Big[1 - \prod_{j\in\mathcal{J}^i}(1-p_{ij})\Big] \;\geq\; \frac{1}{l_i}$$

since for $p_{ij} \leq 1$ always

$$\sum_{j\in\mathcal{J}^i} p_{ij} \geq 1 - \prod_{j\in\mathcal{J}^i}(1-p_{ij}) \quad ,$$

as one can easily check by induction on the number of factors $1 - p_{ij}$. Finally, by summing over i and using again $\Delta U_i = l_i\,w_i$, we obtain the assertion, which completes the proof.

□

3.2. Significant descent implies finite termination

Now suppose that starting from some initial probability distribution P_0 we successively select a sequence of index sets $\mathcal{J}_k$ and obtain the updated probabilities P_k. Then it follows from the Lemma 3 that

$$\frac{1}{n}\sum_k \Delta U(P_k, \mathcal{J}_k) \;\leq\; \sum_k \Delta U(P_k, \mathcal{J}_k)/|\mathcal{J}_k| \;\leq\; U(P_0) \;\leq\; n\,m\ .$$

Consequently, the expected reductions $\Delta U(P_k, \mathcal{J}_k)$ must tend to zero if k grows unbounded. A natural *significant descent* condition on the $\mathcal{J}_k$ is that their expected reduction is no smaller than the maximal one achievable by a singleton $\mathcal{J} = \{j\}$ so that by (19)

$$\Delta U(P_k, \mathcal{J}_k) \;\geq\; \max_{1\leq j\leq n} \Delta U(P_k, \{j\}) \;\geq\; U(P_k)/n \quad . \tag{20}$$

Then it follows immediately from $\Delta U(P_k, \mathcal{J}_k) \rightarrow 0$ that the uncertainty measures $U(P_k)$ must also converge to zero. This means that all element probabilities p_{ij} are set to 0 or 1 at some particular update or they gradually converge toward 1. In fact the last possibility cannot occur as established in the following result.

Proposition 1. *Provided the initial P_0 satisfies the consistency condition* (7) *and the index sets $\mathcal{J}_k$ are chosen such that the condition* (20) *is satisfied the process stops at some finite k with $U(P_k) = 0$ and thus $P_k = S_k = S$.*

□

Proof. Suppose the assertion was wrong. Then the number of $p_{ij} \in \{0, 1\}$ must be stable after a while with all other p_{ij} converging to but never reaching 1 exactly. Let δ_k denote the maximum over all such remaining discrepancies $1 - p^k_{ij} > 0$ at some late probing iteration. Because $U(P_k) \rightarrow 0$ we must have $\delta_k \rightarrow 0$. The singleton probe $\mathcal{J} = \{j\}$ would yield an expected reduction $\Delta U(P_k, \{j\}) \geq \delta_k$. However, by our assumption it may not be taken since it would settle the issue of whether s_{ij} is 0 or 1 and thus reduce the number of remaining undetermined entries further. The probe actually taken must therefore satisfy $\Delta U(P_k, \mathcal{J}_k) \geq \delta_k$ by (20). Moreover we must have $l_i(\mathcal{J}_k) > 1$ for all i with $w_i(\mathcal{J}_k) > 0$ since otherwise the elements with $j \in \mathcal{J}^i_k$ would also be determined by the next probe. Hence we must have in fact $w_i(\mathcal{J}_k) \leq \delta^2_k$ for all $i = 1 \ldots m$, which clearly contradicts the lower bound on $\Delta U(P_k, \{j\})$ for all large k as δ_k tends to zero. This completes the proof by contradiction.

□

Clearly, the proposition also applies to combined schemes, since direct or transpose probing would have to be performed infinitely often if the result was not true. As shown in the proof, the condition (20) on $\mathcal{J}_k$ eventually forces the probing scheme to determine the sparsity column by column using singletons $\mathcal{J} = \{j\}$. This happens once the current probability that the remaining undetermined elements s_{ij} equal 1 is high enough. Obviously this reverting to the simple-minded column by column approach is highly undesirable because it means that some $\min(m, n)$ probes need to be taken, whereas we want to get away with much less. Fortunately, in our computational experience this happens only rarely, for example, in the complementary diagonal case.

3.3. Single probe selection

At each stage we may select $\mathcal{J}$ as one of 2^n possible subsets with the aim of maximising $\Delta U(P, \mathcal{J})$ at least approximately. This task appears to be quite difficult since there may be several local minima. Here we consider two sets as immediate neighbours if they can be transformed into each other by including or excluding just one index $j \in \{1 \ldots n\}$. This effect can already be seen in the following tiny example.

In Fig. 1 the nodes of the diamond represent the four possible choices for subsets $\mathcal{J} \subset \{1, 2\}$. To reach the globally maximal value $\Delta U(P, \{2\})$ from the locally maximal value $\Delta U(P, \{1\})$ one has to go through either one of the local minimizer $\mathcal{J} = \emptyset$ or $\mathcal{J} = \{1, 2\}$. The inequality

$$-\Delta U(P, \{1\}) - \Delta U(P, \{2\}) = -0.7 < -0.24 = \Delta U(P, \{1, 2, 3\})$$

also shows that $-\Delta U(P, \cdot)$ is not a submodular function [Fle00], whose minimization would be a comparatively simple task.

In our current implementation we have adopted the following ascent strategy in two variants. In order to compute at least a local maximum of $\Delta U(P, \mathcal{J})$, we start with the empty set $\mathcal{J} = \emptyset$ and then successively include and exclude column indices while monotonically increasing our objective until no more gain can be achieved in this way. To this end we cycle through all n column indices $j = 1 \ldots n$ and compare $\Delta U(P, \mathcal{J} \cup \{j\})$ or $\Delta U(P, \mathcal{J} \setminus \{j\})$ with the current value $\Delta U(P, \mathcal{J})$. Depending on whether we accept any increases right away or modify $\mathcal{J}$ only after finding the largest variation over a full cycle $j = 1 \ldots n$, we will refer to our method as *first ascent* or *steepest ascent*, respectively. The second strategy appears a lot more thorough than the first.

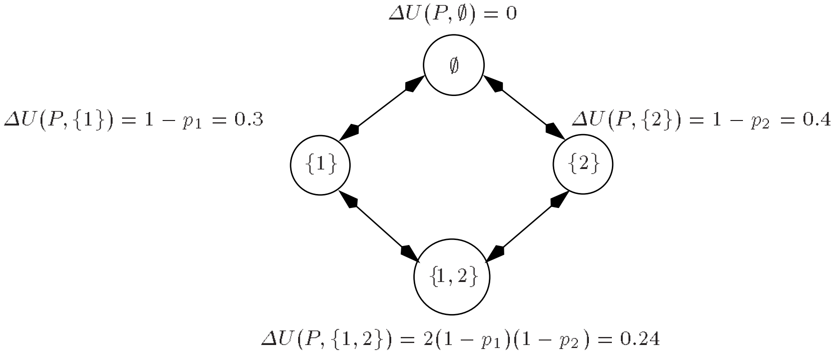
$\Delta U(P, \emptyset) = 0$
$\emptyset$
$\Delta U(P, \{1\}) = 1 - p_1 = 0.3$
$\Delta U(P, \{2\}) = 1 - p_2 = 0.4$
$\{1\}$
$\{2\}$
$\{1, 2\}$
$\Delta U(P, \{1, 2\}) = 2(1 - p_1)(1 - p_2) = 0.24$

Fig. 1. Multimodiality of ΔU on subsets of $\{1, 2\}$ with $P = (p_1, p_2) = (0.7, 0.6)$

Whereas *first ascent* depends on variable and equation numbering, *steepest ascent* applied column- and/or row-wise is invariant with respect to renumberings except when the maximal increase of ΔU is attained for several incoming or outgoing rows or columns during the computation of $\mathcal{J}$ or $\bar{\mathcal{J}}$. However, in our empirical experience the much larger expense for computing the sets $\mathcal{J}_k$ according to the steepest ascent strategy was not justified by a significant reduction in the overall number of probing steps k compared to the numbers achieved by the first ascent strategy. To ensure that the *significant descent* condition (20) is satisfied the very first column j entering the initially empty set $\mathcal{J}$ is always chosen in the steepest ascent fashion as a maximizer of $\Delta U(P, \{j\})$ for $j = 1 \ldots n$. Note that due to the special form (14) of the objective function $\Delta U(P, \mathcal{J})$ the inclusion or exclusion of a single element in $\mathcal{J}$ can be accounted for quite economically.

3.4. Bundle probe selection

As we noted in the introductions it is usually barely more expensive to conduct 32 probes simultaneously than to conduct just one. Hence it is natural to look for a family of sets $\mathcal{J}_k$ with $k = 1 \ldots K$ that maximises the expected reduction of $U(P)$. While the formula given in Lemma 2 is then still valid for one such bundle probe, its actual evaluation seems quite complicated except when we make the assumption that the $\mathcal{J}_k$ are mutually disjoint. Then the k corresponding result events are stochastically independent and we have the total expected reduction

$$\Delta U(P, \mathcal{J}_1, \ldots, \mathcal{J}_k) \equiv \sum_{k=1}^{K} \Delta U(P, \mathcal{J}_k) \quad . \tag{21}$$

Even after imposing disjointness we still have $(K + 1)^n$ possible families of K index sets $\mathcal{J}_k \subset [0 \ldots n]$. To find one with a reasonable effort, we have tried two strategies, so far. In the first *bottom up* approach we simply pick one $\mathcal{J}_k$ at a time by the single probe optimisation specified above but limited to the column indices that have not been included in any one of the predecessors $\mathcal{J}_{\tilde{k}}$ with $\tilde{k} < k$. Here we may find that all indices are assigned before k has reached its upper bound K, meaning that probing capacity would be wasted. To fix this problem we have limited the number of elements in any $\mathcal{J}_k$ to n/K, arriving at what we will refer to as the *bottom up* method based on either first or steepest ascent in each single set selection.

3.5. Top down bundling

Clearly the simple-minded bottom up approach described above is somewhat unsatisfactory, as the various sets $\mathcal{J}_k$ are chosen without any real consideration of their structural interdependence. Therefore we have also implemented the following *top down* approach, which worked better for sizable K even though the overhead is significantly larger.

Our point of departure is the family of singletons $\mathcal{J}_k = \{k\}$ for $k = 1 \ldots n$ for which the maximal expected reduction

$$\Delta U(P, \mathcal{J}_1, \ldots, \mathcal{J}_n) \equiv U(P)$$

would actually be achieved in one bundle probe. Except when the problem is so small that $n \leq K$, we cannot perform n probes simultaneously and therefore have to reduce the test family, which we chose to do by repeatedly merging them in pairs. In merging two disjoint sets $\mathcal{J}$ and $\tilde{\mathcal{J}}$, the objective function contribution

$$\begin{aligned}\Delta U(P, \mathcal{J}, \tilde{\mathcal{J}}) &= \Delta U(P, \mathcal{J}) + \Delta U(P, \tilde{\mathcal{J}}) \\ &= \sum_{i=1}^{m} w_i(\mathcal{J}) l_i(\mathcal{J}) + \sum_{i=1}^{m} w_i(\tilde{\mathcal{J}}) l_i(\tilde{\mathcal{J}})\end{aligned}$$

is because of (17) and (18) reduced to

$$\Delta U(P, \mathcal{J} \cup \tilde{\mathcal{J}}) = \sum_{i=1}^{m} w_i(\mathcal{J}) w_i(\tilde{\mathcal{J}}) \, [l_i(\mathcal{J}) + l_i(\tilde{\mathcal{J}})] \quad .$$

Taking the differences we obtain after some elementary manipulation the objective function variation

$$\sigma(\mathcal{J}, \tilde{\mathcal{J}}) = \sum_{i=1}^{m} \Big[w_i(\mathcal{J}) l_i(\mathcal{J}) [1 - w_i(\tilde{\mathcal{J}})] + w_i(\tilde{\mathcal{J}}) l_i(\tilde{\mathcal{J}}) [1 - w_i(\mathcal{J})] \Big] \; . \tag{22}$$

This function $\sigma(\mathcal{J}, \tilde{\mathcal{J}})$ is always nonnegative and may be interpreted as a similarity measure between the index sets $\mathcal{J}$ and $\tilde{\mathcal{J}}$. The more similar two sets $\mathcal{J}$ and $\tilde{\mathcal{J}}$ are in this sense, the more information is lost when they are merged.

Following again a greedy philosophy we always merge a pair of sets $\mathcal{J}_k$ and $\mathcal{J}_{\tilde{k}}$ for which $\sigma(\mathcal{J}_k, \mathcal{J}_{\tilde{k}})$ is currently minimal. If this minimal loss is still bigger than one of the individual reductions $\Delta U(P, \mathcal{J}_k)$, we decide that it is better to reduce the number of index sets by removing that particular $\mathcal{J}_k$ completely from the family. We may interpret this action as merging $\mathcal{J}_k$ with a fictitious set $\mathcal{J}_0$ for which all $w_i(\mathcal{J}_0)$ vanish. Rather than maintaining a full table of similarity coefficients $\sigma(\mathcal{J}_k, \mathcal{J}_{\tilde{k}})$, which would require $O(n^2)$ storage initially, we keep for each of the remaining sets $\mathcal{J}_k$ only the minimal value $\sigma(\mathcal{J}_k, \mathcal{J}_{\tilde{k}})$ and the corresponding index $\tilde{k}$. This information can be updated with little effort after each set merger or removal. The *top down* strategy just described has been applied with various values of K between 1 and 32. In the special case $K = 1$ we obtain an unbundled probing scheme that works acceptably well but seems a little less efficient than the *bottom up* approach described earlier.

3.6. Combination with transpose

Identifying all nonzeros by direct probing always requires n test vectors if the Jacobian has a dense row. As already discussed at the end of Subsect. 2.5, for the single probe case we have therefore also included a row-wise implementation of our methods for selecting [bundle] probes. In all methods labeled as *combined* in Sect. 4, we have then compared the expected reduction for direct [bundle] probing with that for transpose [bundle] probing and selected the more promising mode in each step. In case of ties, which occur especially in the beginning for square matrices, direct probing is applied. The total number of test vectors is then the sum of the column and row probes.

3.7. Compressed storage scheme

In all schemes for finding a suitable probe one needs to access the current probabilities p_{ij} in order to evaluate $U(P)$ and $\Delta U(P, \mathcal{J})$ for prospective index sets $\mathcal{J}$. Storing and manipulating P as a dense array of $m\,n$ entries seems rather inappropriate on truly large problems, where the actual number of nonzeros in S can expected to be a small multiple of $m+n$. Instead one may store for each direct probe only the column indices $\mathcal{J}_k = \text{supp}(t^{(k)})$, the row indices $\bar{\mathcal{J}}_k \equiv \text{supp}(S\,t^{(k)})$ and the corresponding nonzero values $\hat{w}_i^k \equiv 1/(1-w_i)$ with w_i computed as in (12) at the kth probe. Setting formally $\hat{w}_i^k = 0$ for all $i \notin \bar{\mathcal{J}}_k$, we obtain the product representation

$$p_{ij}^k \;=\; p_{ij}^0 \prod_{\tilde{k}\,:\,j\,\in\,\mathcal{J}_{\tilde{k}}} \hat{w}_i^{\tilde{k}}. \tag{23}$$

We may assume that the initial probability p_{ij}^0 is a simple function of its indices. For example it may be constant or depend only on the index distance $|i-j|$ from the diagonal. Then the total storage and the effort for computing one current probability component p_{ij}^k grow linearly as functions of the number k of probes so far performed. In case of disjoint bundle probes the effort grows only with the number of probes since the condition $j \in \mathcal{J}_{\tilde{k}}$ in (23) can only be met for one set $\mathcal{J}_{\tilde{k}}$ in each bundle. In case of combined probing, the right hand side of (23) must be multiplied by the corresponding product over all transposed probes that effect the ith row.

4. Experimental results

Throughout this section we display along the vertical axis the total number k of Boolean products needed to completely detect the sparsity pattern. When probes are taken in bundles of size K the total number k is always a multiple of K. In the histograms Figs. 6 and 8 the height of the shaded parts represents the number of bundles k/K, with the total height still equalling k. Usually k grows when K is increased as less information is available in choosing the probing vectors. All results were obtained from the neutral initialization $p_{ij} = 1/\max(m,n)$ *motivated* in Subsect. 2.7. All calculation were stopped only when $U(P)$ had been reduced exactly to zero so that $P = S$, the actual sparsity pattern.

4.1. Specially structured matrices

As we have mentioned in the introduction, diagonal matrices of dimension n can be detected using $2\log_2 n$ probes. This theoretical result assumes that this very special sparsity pattern is given and merely needs to be verified.

The dotted line in Fig. 2 displays the number of probes needed by our *bottom up* first ascent scheme starting from the uniform distribution $p_{ij} = 1/n$ and thus without any specific information. Only direct products were allowed and exactly the same results would have been obtained on any row or column permutation of the identity matrix.

As one can see the method detects the diagonal structure in roughly $2\log_2 n$ probes, which is only twice the lower bound for any sparsity pattern with n distinct columns. The dashed line above gives a similar result for a tridiagonal matrix and the solid one on top was computed for the sparsity structure obtained from the 5-Point stencil on the unit square with $\sqrt{n}$ grid points in each direction. As one can see the dependence of the number of probes required on the matrix dimension appears to be logarithmic in all cases. However, the number of probes in the tridiagonal case is more than twice as many as the estimate $2\log_2 n$ established in [GM01] for that problem.

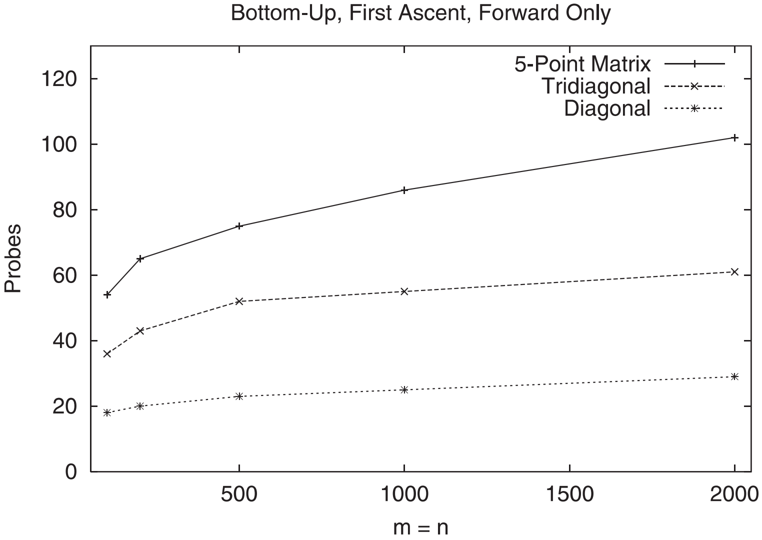

Fig. 2. Forward bottom-up first ascent on diagonal, tridiagonal and 5-point

Apparently due to the symmetry of the three matrix structures examined above, the use of transpose Boolean products was found not to reduce the number of probes needed for complete determination. The situation is different for matrices with dense rows and columns, like the simple arrowhead structure obtained by adding a dense last row and column to a diagonal matrix. Then complete determination based on either direct or transpose products alone requires $k = n$ probing steps. In contrast, the number of probes taken by our combined bottom-up first ascent scheme again grows only logarithmically with the number of transpose products being roughly 20 seemingly irrespective of n. This result is displayed in Fig. 3, where the solid line represents the bottom up first ascent and the dashed one the steepest ascent variant. As one can see the differences are not consistent and certainly not very large.

The results obtained by our probing scheme on the four special sparsity structures considered in this subsection are very satisfactory, but they certainly do not demonstrate its suitability for general sparse problems. For this purpose we consider in the remainder randomly sparse matrices and a subset from the Harwell-Boeing test collection. We also examine the effects of bundling, which makes the individual Boolean product much

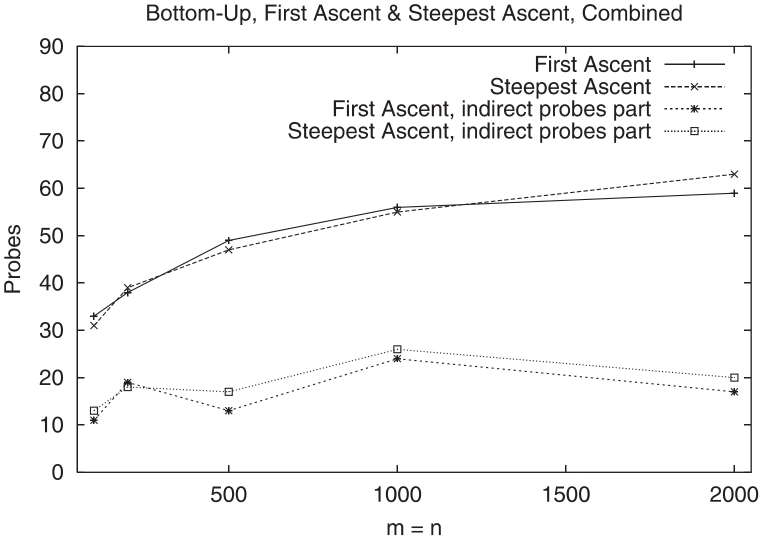

Fig. 3. Combined bottom-up first ascent and steepest ascend on arrowhead

cheaper. The dependence on the initialization is in general quite weak as shown in the following section.

4.2. *Randomly sparse matrices*

Figure 4 displays the results for square matrices of dimension $n = 500$ and 1000 with $10n$ randomly distributed nonzeros and various constant initializations of the p_{ij}. As one can see the results are the same for all sufficiently small initial probabilities, which agrees with our derivation of the neutral initialization in Subsect. 2.7.

Figure 5 displays the results for square matrices of dimension n with $n^2/100$ or $10n$ randomly distributed nonzero entries, respectively. In the first case represented by the solid line, the number of probes appears to be growing only slightly faster than linear as a function of n. The number of probes required is roughly 17 times the average number of nonzero elements per row, irrespective of whether n equals 200, 500, 1000, or 2000. In the second case represented by the dashed line the average number of nonzeros per row was kept constant at 10 and the resulting number of probes appears to grow again logarithmically.

Hence we might formulate the daring conjecture that for randomly generated square matrices

$$\boxed{\#\text{probes} \sim \text{length} * \log(\min(m, n))} \quad . \tag{24}$$

Here length is the average number of nonzeros in any row or column of the matrix. Irrespective of whether or not this conjecture is true, our results indicate that the probing scheme can reveal even random sparsity patterns at a fraction of the cost that would be incurred by the basic column by column approach.

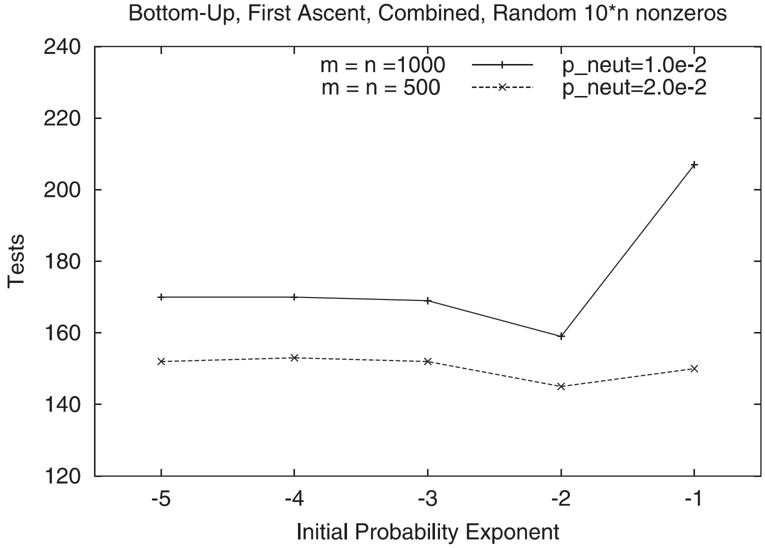

Fig. 4. Dependence of probes on initial probability $p_{ij} = 10^k$ for $k = -5 \ldots -1$

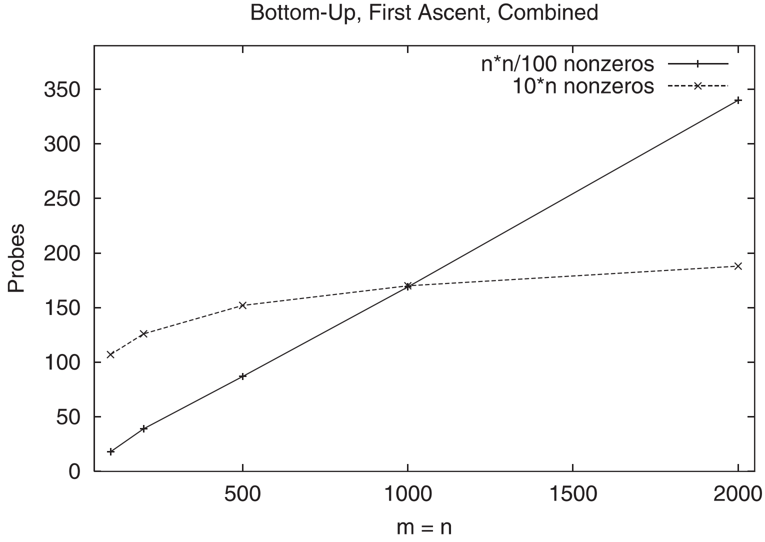

Fig. 5. Number of probes on random matrices with linear row or constant length

Things look even better when we utilize the bundling procedure described in Subsects. 3.4, 3.5 and 3.6. Contrary to what one might have expected, bundling into groups of 2, 4, 8, 16, and 32 probes at a time does not increase the total number of direct or transpose probes very much at all. Unfortunately, the overhead of computing the test vector bundles by our *top-down* approach is still quite significant, which explains why we show the results only up to $n = 500$ in Fig. 6. As one can see, a random

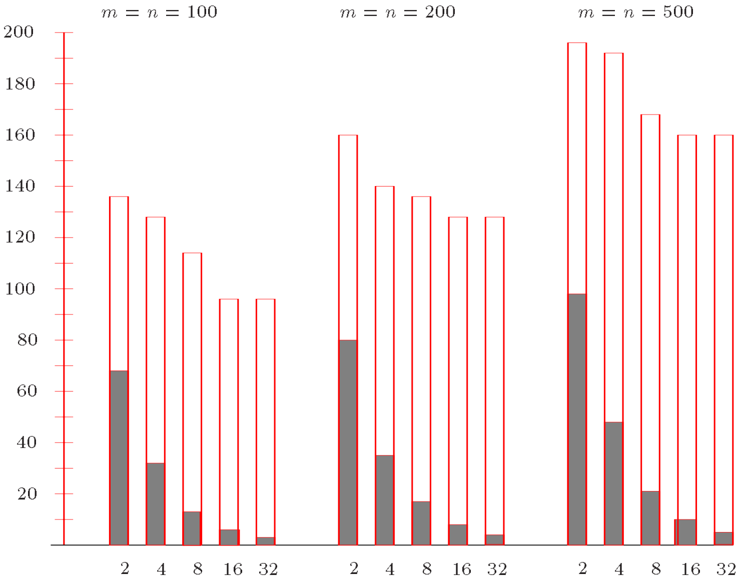

Fig. 6. Combined top-down with bundle-size $K = 2,\ 4,\ 8,\ 16,\ 32$ on random matrices with 10 nonzeros per row on average

sparsity structure of size $n = 500$ can be detected using just 5 bundles of size 32, which corresponds to a cost of about 5 directional derivatives and thus roughly 15 function evaluations.

4.3. Harwell-Boeing matrices

Finally we consider six rectangular matrices from the Harwell Boeing collection [DGL]. Table 1 lists the problem acronyms, the dimensions m and n, the minimal and maximal number of nonzeros per row mnr and mxr, the corresponding values mnc and mxc per column, and finally the percentage of nonzero elements overall.

Table 1. Characteristics of Harwell-Boeing selection

Matrix	m	n	mnr	mxr	mnc	mxc	nz(%)
ABB313	313	176	1	6	2	26	2.83
ASH219	219	85	2	2	2	9	2.35
ASH331	331	104	2	2	3	12	1.92
ASH608	608	188	2	2	2	12	1.06
ASH958	958	292	2	2	3	12	0.68
WIL199	199	199	1	6	2	9	1.77

None of these matrices has more than 6 nonzeros per row, which explains why the combined first ascent scheme was not dramatically faster than the corresponding direct only versions, as displayed in Fig. 7. The black partitions of the columns represent the number of transpose probes. The probe counts for the two corresponding steepest ascent variants are not shown as they were, just slightly lower, when they differed all.

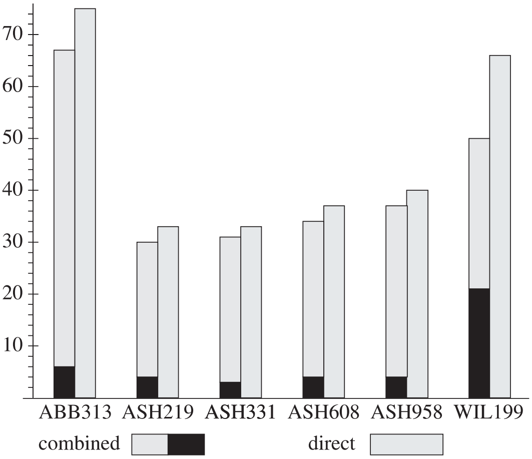

Fig. 7. First ascent schemes on Harwell-Boeing selection

Compared to the row and column dimensions of the problems, the number of probes required seems acceptable, though computing some 30-60 Boolean products in sequence is certainly not a negligible effort. As we noted in the introduction this is in practice at least 30 times as expensive as propagating one or two integer vectors that correspond to bundles of 32 test vectors.

Fortunately, bundling by our *top-down* scheme does not increase the overall number of probing vectors very much, as was already observed on the random matrices and can now be seen in Fig. 8. The entries in Fig. 8 are slightly reminiscent of speed-up tables in parallel computing, where the wall-clock times scale almost inversely with the number of processes. Here the number of bundle probes k/K represented by the shaded column parts are almost reciprocal to the size K of each bundle. For the natural choice $K = 32$, we never need more than 3 bundle probes, of which the first one is computed almost completely blind, since all entries of the initial P are again chosen equal to the neutral value $1/\min(m, n)$.

As one can see in Fig. 9 the sparsity patterns of the Harwell-Boeing matrix `ASH219` has a certain degree of regularity, but it is by no means trivial. To be able to figure out that pattern in just two bundle probes must be considered a very satisfactory achievement of the proposed approach.

5. Summary and discussion

In this paper we have developed a sequential probing scheme for determining the sparsity pattern of Jacobian matrices for vector functions defined by computer pro-

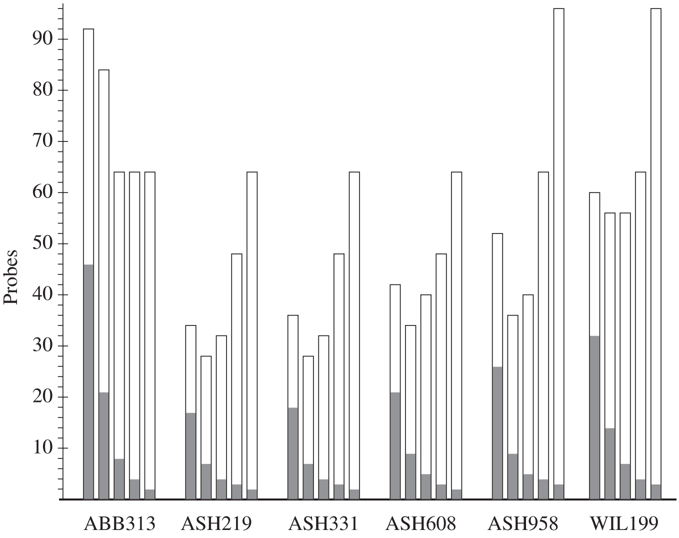

Fig. 8. Combined top-down with bundle sizes 2, 4, 8, 16, 32

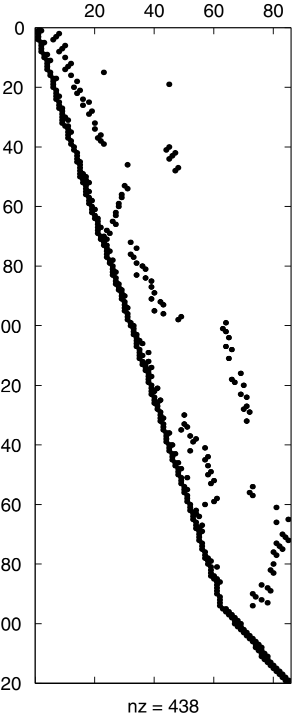

Fig. 9. Sparsity pattern of ASH219 with m=219 and n=85

grams. The problem specific input for this Bayesian estimation procedure consists of Boolean Jacobian-vector products, which can be generated either by differencing or suitably adapted automatic differentiation software. In the former case there is a remote possibility that false zeros are detected due to exact cancellations at the current argument.

The dependence information generated by automatic or algorithmic differentiation is more reliable, and in its reverse mode one can also evaluate Boolean products of the transpose Jacobian at a similar cost. As demonstrated on the arrowhead example in Fig. 3, this capability is very useful if the Jacobian in question happens to have some (nearly) dense rows. For this and other highly structured square matrices, we found that the total number of probing vectors required grew only logarithmically as a function of the dimension n. The same observation applies to randomly generated matrices with a uniformly bounded number of nonzeros per row. Moreover, if test vectors were bundled in groups of up to 32 the total number of test vectors grew only by a factor less than two in all cases. Especially gratifying and still a little surprising is that the sparsity patterns of several sizable matrices from the Boeing test collection could be identified in just two or three bundle probes, with the first one being selected without any structural information.

There are several aspects that warrant further examination and improvement. First, it seems likely that the number of probes can be bounded on certain classes of matrices. Possibly a probabilistic analysis could also verify that the expected number of probes on randomly sparse matrices satisfies a relation similar to our conjecture (24). The sequential updating approach may lead to some probabilities p_{ij}'s remaining below 1 even though a simultaneous analysis of all probing results would reveals them already as being 1. While this information may be a little hard to come by, it implies that all probes involving column j or row i provide no information regarding other elements in row i or column j, respectively. Therfore, an implementation of the simultaneous approach with careful attention to the data structures used to store the probing results already gathered may well be worthwhile.

Even when the updated probabilities can be (re)computed very efficiently, it would appear that at least our *top-down* procedure is much too costly. Some closer analysis of the similarity measure (22) and other optimization heuristics should lead to a faster bundle calculation. Possibly, one might even be more general and allow overlapping index sets $\mathcal{J}_k$ within each bundle probe.

Also, some more thought should be given to the initialization of the probability distribution, which was simply uniform equal to $1/\min(n, m)$ in all our test calculations. In many applications it would make sense to initialize p_{ij} as a function of some "distance measure" between the jth independent and the ith dependent variable. The closer they are the more likely they are to interact, as is certainly the case in discretizations of differential equations with possibly irregular grids. Then using the Euclidean distance between the associated grid points would seem rather natural.

Acknowledgements. The authors gratefully acknowledge the advice of Prof. D. Ferger and Dr. J. Franz of the Institute of Stochastics in the Department of Mathematics at the Technical University of Dresden. They are also thankful for the referee's comments, which helped them to significantly improve the readability of the paper.

References

[AZ85] Assaf, D., Zamir, S. (1985): Optimal Sequential Search: A Bayesian Approach. Ann. Statist. **13**, 1213–1221

[AC+92] Averick, B.M., Carter, R.G., Moré, J.J., Xue, G.-L. (1992): The MINPACK-2 test problem collection. Preprint MCS–P153–0692, ANL/MCS–TM–150, Rev. 1, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL

[BB+96] Berz, M., Bischof, C.H., Corliss, G., Griewank, A., eds. (1996): Computational differentiation–techniques, applications, and tools. SIAM, Philadelphia

[BCK95] Bischof, C.H., Carle, A., Khademi, P.M. (1995): Fortran 77 Interface Specification to the SparsLinC 1.0 Library. ANL/MCS-TM-196, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL

[BK+97] Bischof, C.H., Khademi, P.M., Bouaricha, A., Carle, A. (1997): Efficient computation of gradients and Jacobians by dynamic exploitation of sparsity in automatic differentiation. Optim. Methods Softw. **7**, 1–39

[BSG96] Bjørstad, P., Smith, B., Gropp, W. (1996): Domain decomposition: Parallel multilevel methods for elliptic partial differential equations. Cambridge University Press

[CF+01] Corliss, G., Faure, C., Griewank, A., Hascoët, L., Naumann, U., eds. (2001): Automatic Differentiation: From Simulation to Optimization. Springer

[CM84] Coleman, T.F., Moré, J.J. (1984): Estimation of sparse Jacobian matrices and graph coloring problems. SIAM J. Numer. Anal. **20**), 187–209

[CP+74] Curtis, A.R., Powell, M.J.D., Reid, J.K. (1974): On the estimation of sparse Jacobian matrices. J. Inst. Math. Appl. **13**, 117–119

[CV95] Coleman, T.F., Verma, A. (1998): The efficient computation of sparse Jacobian matrices using Automatic Differentiation. SISC, **19**(4), 1210–1233

[DGL] Duff, I.S., Grimes, R.G., Lewis, J.G.: Users' Guide for the Harwell-Boeing Sparse Matrix Collection (Release I). Research and Technology Division, Boeing Computer Services, Mail Stop 7L-2, P.O. Box 24236, Seatle, WA98124-0346, USA

[Fle00] Fleischer, L. (2000): Recent progress in Submodular Minimization. OPTIMA 64, Math. Prog. Soc. Newslett., Univ. Florida

[GK00] Giering, R., Kaminski, T.: Generating recomputations in reverse mode AD. In: Corliss et al. [CF+01], chap. 32

[GK01] Giering, R., Kaminski, T. (2001): Automatic Sparsity Detection. Draft, March 2001

[GM01] Griewank, A., Mitev, C.: Verifying Jacobian Sparsity. In: Corliss et al. [CF+01], chap. 31

[GR02] Griewank, A., Riehme, J.: Covering Zeros and Isolating Nonzeros in Sparse Jacobians. In preparation

[Gri00] Griewank, A. (2000): Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia

[GGU96] Geitner, U., Griewank, A., Utke, J.: Sparse Jacobians by Newsam/Ramsdell. In [BB+96], pp. 161–172

[GJ+96] Griewank, A., Juedes, D., Utke, J. (1996): ADOL–C, a package for the automatic differentiation of algorithms written in C/C++. ACM Trans. Math. Software **22**, 131–167, `http://www.math.tu-dresden.de/wir/project/adolc/`

[Hoc97] Hochbaum, D.S., ed. (1997): Approximation Algorithms for NP-Hard Problems. PWS Publishing Company, 20 Park Plaza, Boston

[ST98] Shahadat Hossain, A.K.M., Steihaug, T. (1998): Computing a sparse Jacobian matrix by rows and columns. Optimization Methods and Software **10**, 33–48

[NR83] Newsam, G.N., Ramsdell, J.D. (1983): Estimation of sparse Jacobian matrices. SIAM J. Alg. Disc. Meth. **4**(3), 404–417

[KS79] Kimeldorf, G., Smith, F.H. (1979): Binomial Searching for a Random Number of Multinomially Hidden Objects. Management Sci. **23**, 1115–1126

[Lin95] Lindley, D.V. (1995): Bayesian Statistics, A Review. CBMS-NSF Regional Conference Series in Appl. Math. SIAM, University College London