SPAI – SParse Approximate Inverse

Thomas Huckle
Institut für Informatik
TU München
85748 Garching
Germany

Sparse Approximate Inverse Matrix

For a given sparse matrix $A$ a sparse matrix $M \approx A^{-1}$ is computed by minimizing $\|AM - I\|_F$ in the Frobenius norm over all matrices with a certain sparsity pattern. In the SPAI algorithm the pattern of $M$ is updated dynamically to improve the approximation until a certain stopping criterion is reached.

Introduction

For applying an iterative solution method like the conjugate gradient method (CG), GMRES, BiCGStab, QMR, or similar algorithms, to a system of linear equations $Ax = b$ with sparse matrix $A$, it is often crucial to include an efficient preconditioner. Here, the original problem $Ax = b$ is replaced by the preconditioned system $M Ax = Mb$ or $Ax = A(My) = b$. In a parallel environment a preconditioner should satisfy the following conditions:

- $M$ can be computed efficiently in parallel.
- $Mc$ can be computed efficiently in parallel for any given vector $c$.
- The iterative solver applied on $AMx = b$ or $MAX = Mb$ converges much faster than for $Ax = b$ (e.g. it holds $\text{cond}(MA) \ll \text{cond}(A)$).

The first two conditions can be easily satisfied by using a sparse matrix $M$ as approximation to $A^{-1}$. Note, that the inverse of a sparse $A$ is nearly dense, but in many cases the entries of $A^{-1}$ are rapidly decaying, so most of the entries are very small [11].
Benson and Frederickson [4] were the first to propose a sparse approximate inverse preconditioner in a static way by computing
\begin{equation}
\min_M \|AM - I\|_F
\end{equation}
for a prescribed a priori chosen sparsity pattern for \( M \). The computation of \( M \) can be split into \( n \) independent subproblems \( \min_{M_k} \|AM_k - e_k\|_2 \), \( k = 1, ..., n \) with \( M_k \) the columns of \( M \) and \( e_k \) the \( k \)-th column of the identity matrix \( I \).

In view of the sparsity of these Least Squares (LS) problems, each subproblem is related to a small matrix \( \hat{A}_k := A(I_k, J_k) \) with index set \( J_k \) which is given by the allowed pattern for \( M_k \) and the so-called shadow \( I_k \) of \( J_k \), that is, the indices of nonzero rows in \( A(:, J_k) \). These \( n \) small LS problems can be solved independently, for example, based on QR decompositions of the matrices \( \hat{A}_k \) by using the Householder method or the modified Gram-Schmidt algorithm.

The SPAI Algorithm

The SPAI algorithm is an additional feature in this Frobenius norm minimization that introduces different strategies for choosing new profitable indices in \( M_k \) that lead to an improved approximation. Assume that, by solving (1) for a given index set \( J \), an optimal solution \( M_k(J_k) \) has been already determined resulting in the sparse vector \( M_k \) with residual \( r_k \). Dynamically there will be defined new entries in \( M_k \). Therefore, (1) has to be solved for this enlarged index set \( \tilde{J}_k \) such that a reduction in the norm of the new residual \( \tilde{r}_k = A(I_k, \tilde{J}_k)M_k(\tilde{J}_k) - e_k(I_k) \) is achieved.

Following Cosgrove, Griewank, Díaz [10], and Grote, Huckle [13], one possible new index \( j \in J_{new} \) out of a given set of possible new indices \( J_{new} \) is tested to improve \( M_k \). Therefore, the reduced 1D problem
\begin{equation}
\min_{\lambda_j} \|A(M_k + \lambda_j e_j) - e_k\| = \min_{\lambda_j} \|\lambda_j A_j + r_k\|
\end{equation}
has to be considered. The solution of this problem is given by
\[
\lambda_j = -\frac{r_k^T A e_j}{\|A e_j\|^2}
\]
which leads to an improved squared residual norm
\[
\rho_j^2 = \|r_k\|^2 - \frac{(r_k^T A e_j)^2}{\|A e_j\|^2}.
\]
Obviously, for improving \( M_k \) one has to consider only indices \( j \) in rows of \( A \) that are related to the nonzero entries in the old residual \( r_k \); otherwise they do not lead to a reduction in the residual norm. Thus, the column indices \( j \) have to be determined that satisfy \( r_k^T A e_j \neq 0 \) with the old residual \( r_k \). Let the index set of nonzero entries in \( r_k \) be denoted by \( L \). Furthermore, let \( \tilde{J}_i \) denote the set of new indices that are related to the nonzero elements in the \( i \)-th row of \( A \),
and let $J_{\text{new}} = \bigcup_{i \in \tilde{J}} J_i$ denote the set of all possible new indices that can lead to a reduction of the residual norm. Then, one or more indices $J_c$ are chosen as a subset of $J_{\text{new}}$ that corresponds to a large reduction in $r_k$. For this enlarged index set $J_k \cup J_c$ the QR decomposition of the related LS submatrix has to be updated and solved for the new column $M_k$.

Inside SPAI there are different parameters that steer the computation of the preconditioner $M$:

- How many entries are added in one step
- How many steps of adding new entries are allowed
- Start pattern
- Maximum allowed pattern
- What residual $\|r_k\|$ should be reached
- How to solve the LS problems

**Modifications of SPAI**

A different and more expensive way to determine a new profitable index $j$ with $\tilde{J}_k := J_k \cup \{j\}$ considers the more accurate problem

$$\min_{M_k(J_k)} \|A(:, \tilde{J}_k)M_k(\tilde{J}_k) - e_k\|$$

introduced by Gould and Scott [12]. For $\tilde{J}_k$ the optimal reduction of the residual is determined for the full minimization problem instead of the 1D minimization in SPAI.

Chow [9] showed ways to prescribe an efficient static pattern a priori and developed the software package PARASAILS.

Holland, Shaw, and Wathen [17] have generalized this ansatz allowing a sparse target matrix on the right side in the form $\min_M \|AM - B\|_F$. This approach is useful in connection with some kind of two-level preconditioning: First compute a standard sparse preconditioner $B$ for $A$ and then improve this preconditioner by an additional Frobenius norm minimization with target $B$. From the algorithmic point of view the minimization with target matrix $B$ instead of $I$ introduces no additional difficulties. Only the pattern of $M$ should be chosen more carefully with respect to $A$ and $B$.

Zhang [23] introduced an iterative form of SPAI where in each step a thin $M$ is derived starting with $\min_{M_1} \|AM_1 - I\|_F$. In the second step the sparse matrix $AM_1$ is used and $\min_{M_2} \|(AM_1)M_2 - I\|_F$ is solved, and so on. The advantage is, that because of the very sparse patterns in $M_i$ the Least Squares problems are very cheap.

Chan and Tang [8] applied SPAI not to the original matrix but first used a Wavelet transform $W$ and computed the sparse approximate inverse preconditioner for $WAW^T$ that is assumed to be more diagonal dominant.
Yeremin, Kolotilina, Nikishin, and Kaporin [19, 20] introduced factorized
sparse approximate inverses of the form $A^{-1} \approx LU$. Huckle generalized the
factorized preconditioners adding new entries dynamically like in SPAI [14].

Grote and Barnard [2] developed a software package for SPAI and also intro-
duced a block version of SPAI.

combined SPAI with the probing method [7] in the form

$$\min_{M} \left( \|AM - I\|_F^2 + \rho^2 \|e^T AM - e^T\|_2^2 \right)$$

for probing vectors $e$ on which the preconditioner should be especially improved.
Furthermore, they developed a software package for MSPAI.

**Properties and Applications**

**Advantages of SPAI:**

- Good parallel scalability.
- SPAI allows modifications like factorized approximation or including prob-
ing conditions to improve the preconditioner relative to certain subspaces,
  for example, as smoother in Multigrid or for regularization [16].
- It is especially efficient for preconditioning dense problems (see Benzi [1]
et al.).

**Disadvantages of SPAI:**

- SPAI is sequentially more expensive, especially for denser patterns of $M$.
- Sometimes it shows poor approximation of $A^{-1}$ and slow convergence as
  preconditioner.

**RELATED ENTRIES**

- Iterative solution of linear systems
- Krylov subspace methods like CG, GMRES, BiCGStab, QMR
- Preconditioners like Jacobi, Gauss-Seidel, ILU
- Approximate inverse preconditioners like SAI, FROB, SFAI, FSPAI, MSPAI, AINV
BIBLIOGRAPHIC NOTES AND FURTHER READING

Books:


Software:

1. Chow, E., Parasails, 
   https://computation.llnl.gov/casc/parasails/parasails.html
2. Barnard, S., Bröker, O., Grote, M., and Hagemann, M., SPAI and Block SPAI, http://www.computational.unibas.ch/software/spai,
3. Huckle, T., Kallischko, A., and Sedlacek, M. MSPAI, 
   http://www5.in.tum.de/wiki/index.php/MSPAI.

BIBLIOGRAPHY


