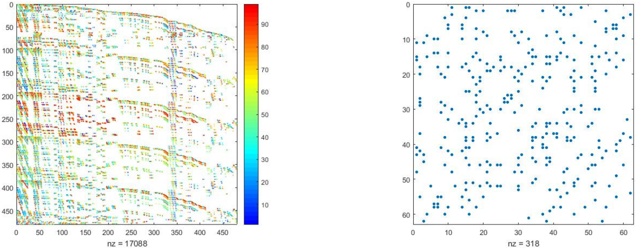
Balancing Size and Structure to Reveal Useful Matrix Properties

Philip A. Knight[1]
with I. Duff, S. Mouysset, D. Ruiz, B. Uçar

[1]Department of Mathematics and Statistics
University of Strathclyde

Toulouse, July 2015

Motivation



- How do we balance size and structure of nonzeros in a matrix to draw out key features?
- Can we make effective use of spectral information?
- $\kappa_\infty(DAE)$ minimised over all diagonal scalings if $D|A|E$ and $E^{-1}|A^{-1}|D^{-1}$ have equal row sums.

Balancing a Matrix

$$DA, \quad AD, \quad DAD^{-1}, \quad D^{1/2}AD^{1/2}, \quad D_1AD_2$$

- Choice of where to apply balance, measure of balance, method of balance.
- We choose to find D_1 and D_2 so that D_1AD_2 has equal row/column sums.
- We use a fast Newton iteration to ensure rapid computation.
- Elements that link together almost disjoint blocks are highlighted.

Spectral Properties

- $P = D_1|A|D_2$.
- We can extend Fiedler/Perron–Frobenius theory.
- If P has k disjoint components principal singular value is repeated k times.
- $\sigma_1 = 1$.
- Typical singular vector: permutation of $\begin{bmatrix} 1 & \ldots & 1 & 0 & \ldots & 0 \end{bmatrix}^T$.
- We can infer block entire block structure from a single singular vector.
- For symmetric matrices we can ensure $D_1 = D_2$ and work with eigenvectors.

Algorithm

1. Preprocess.
2. Balance.
3. Calculate singular vector(s).
4. Split vector(s) to identify blocks.

Preprocessing

- We want to avoid scaling a matrix if it is not fully indecomposable.
- Initialise by looking for BTF and work on biggest block.
- After roughly balancing matrix try and remove strongly diagonally dominant parts of diagonal.

Fast Balancing

- Suppose A is symmetric and $DAD\mathbf{e} = \mathbf{e}$ where $D = \mathbf{diag}(\mathbf{x})$.
- Rewrite: $A\mathbf{x} - \mathbf{diag}(\mathbf{x})^{-1}\mathbf{e} = \mathbf{0}$.
- Solve using Newton method. Newton step solved approximately with CG.
- Easily adapted to nonsymmetric A.
- Typically requires a small number of matrix-vector products using A and A^T.

Computing Singular Vectors

- We compute $p = 1, 2, 3, 4, 5, \dots$ singular vectors with `eigs`.
- Convergence can be slow.
- Output dependent on p and initial guess.
- We project out contribution of $\mathbf{e}$.
- We can use information from pth vector to further project to enhance $(p+1)$th.

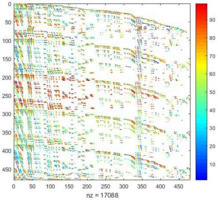
Splitting A Vector, I

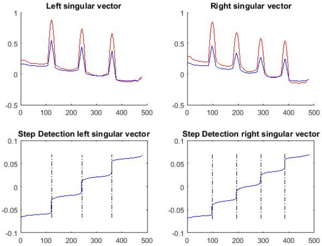
- Reorder components of vector by size.
- Identify jumps with an edge detecting algorithm (Canny filter).
- Jumps resolved at multiple levels.
- Parameter free determination of k blocks.

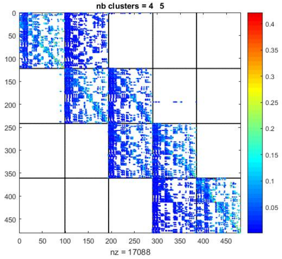
Splitting A Vector, II

- We can use $(p+1)$th vector to refine blocks determined by first p.
- Currently we split all blocks according to information supplied by latest vector.
- To avoid countless tiny blocks we may be better refining existing blocks.
- At the end we can reconstruct matrix based on all blocks uncovered.
- For example, we can attempt to pack the diagonal with large elements.

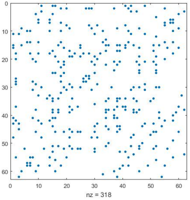
Example: Matrix Blocks

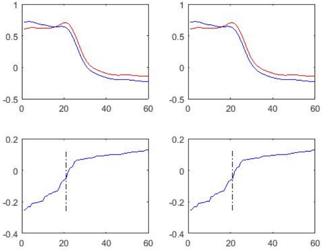


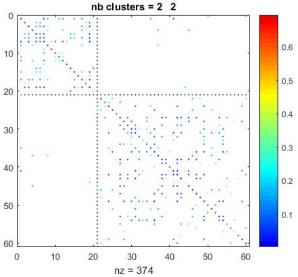




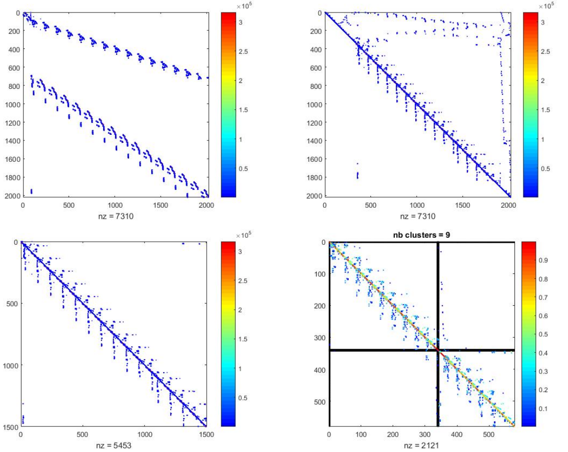
Example: Network Clusters







Example: The Need for Preprocessing



Future Work, I

- We aim to provide a block structure amenable to factorisation and/or preconditioning.
- Preprocessing: there is no point in trying to scale a matrix if it is not fully indecomposable. We guard against this but would like to do better.
- Need to fully understand role of diagonal dominance in the substructures.
- Bi-clustering: algorithm can work with rectangular input.
- To what extent can we reveal useful information simply by using adjacency matrix of A?

Future Work, II

- We want to fill some theoretical gaps.
- Perturbation theory for singular vectors of nearly block matrices is missing.
- How much can we using existing theory on Laplacians?
- We use a measure of cluster quality in reconstructing blocks. . . are we using the right one?