SAND2013-0505C

Exceptional service in the national interest

Sandia National Laboratories

Efficient Computation of Eigenpairs for Large Scale-free Graphs

SIAM CS&E, Feb. 26, 2013

Erik Boman, Karen Devine,

Richard Lehoucq, Nicole Slattengren,

Kevin Deweese

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Scale-Free Graphs

- Large Graphs (Networks) are often "scale-free"
  - Power-law degree distribution
  - Very sparse (low average degree)
  - Low diameter (small-world)
- Different from traditional physics-based simulations
  - Does this change choice of algorithms?
  - Do our current computational tools (software) work?

Our Focus: Graph Laplacians

- The combinatorial Laplacian of a graph G is the sparse matrix L(G) = D-A, where
  - A is the adjacency matrix of G
  - D= diag(d) contains the degrees of the vertices in G

- We wish to compute the extreme eigenpairs of L
  - Used in network analysis
  - Also used in spectral partitioning and ordering

Eigensolvers

- We use the Anasazi package from Trilinos, which has a wide variety of methods:
  - Block Krylov Schur (BKS)
  - LOBPCG
  - Block Davidson (BD)
  - Implicit Riemannian Trust-Region (IRTR)

Preconditioners

- Some eigensolvers can be accelerated using a preconditioner: LOBPCG, IRTR
- Any preconditioner designed for linear systems can be used
- We compare “black-box” algebraic preconditioners from Ifpack:
  - Jacobi (diagonal)
  - Symmetric Gauss-Seidel (SGS)
  - Incomplete Cholesky (IC)

Support-Graph Preconditioners

Main Idea: Construct a sparser subgraph that is a good spectral approximation, use this as preconditioner.

- First proposed by Vaidya ('90) but not published
- First described publicly in [Bern et al.] and implemented by [Chen and Toledo]
- Important recent progress in theoretical CS community; near optimal solvers by Spielman et al. and also by Miller et al.
  - However, no software available
- We have implemented some simple versions (Deweese'12):
  - MSF: Maximum-weight spanning forest
  - MSF(k): Union of k spanning forests (higher quality, but expensive)

Test Graphs

- Real-world graphs/matrices from UF and SNAP collections
- Symmetrized (if needed), largest component

Matrix/graph	Rows	#nonzeros
Enron	68 K	507 K
P2p-Gnutella	63 K	296 K
Dblp-2010	226 K	1433 K

Comparison of Eigensolvers

- Test graph: enron
- Nev = blocksize = 5 (BKS: blocksize=1)
- Tol = 1e-5 (absolute)

Method	Lapl.	Prec.	Iter.	Matvec.	Total time
BKS	Comb.		>50000	--	--
BD	Comb.	Jac.	>50000	--	--
LOBPCG	Comb.	Jac.	466	2335	46.7
IRTR	Comb.	Jac.	14	492	27.1
BD	Norm.		4397	43980	200.6
BKS	Norm.		865	865	22.6
LOBPCG	Norm.		250	1255	20.4
IRTR	Norm.		12	380	13.6

Comparison of Eigensolvers

- Test graph: p2p_Gnutella
- Nev = blocksize = 5 (BKS: blocksize=1)
- Tol = 1e-5 (absolute)

Method	Lapl.	Prec.	Iter.	Matvec.	Total time
BKS	Comb.		13615	13615	144.7
BD	Comb.	Jac.	1510	15125	81.4
LOBPCG	Comb.	Jac.	147	770	12.5
IRTR	Comb.	Jac.	14	1035	9.7
BD	Norm.		1309	13120	62.8
IRTR	Norm.		14	1515	9.9
LOBPCG	Norm.		121	650	9.4
BKS	Norm.		705	705	7.5

Comparison of Eigensolvers

- Test graph: Dblp-2010
- Nev = blocksize = 5 (BKS: blocksize=1)
- Tol = 1e-5 (absolute)

Method	Lapl.	Prec.	Iter.	Matvec.	Total time
BKS	Comb.		>20000	>20000	--
BD	Comb.	Jac.	>2000	>10000	--
LOBPCG	Comb.	Jac.	613	3110	202.6
IRTR	Comb.	Jac.	19	4550	185.7
BD	Norm.		>2000	>10000	--
BKS	Norm.		5235	5235	232.7
IRTR	Norm.		16	990	138.4
LOBPCG	Norm.		422	2155	127.6

Preconditioning Results: Combinatorial Laplacian

- Test graph: enron_bsl (67K rows, 507K nonzeros, 1 connected component)
- Solver=LOBPCG
- Nev = blocksize = 5
- Tol = 1e-5 (absolute)

Precon dition	Lapl.	Iter.	Matvec.	Setup time	Iterate time	Total time
None	Comb.	6896	34505		564.8	564.8
Jacobi	Comb.	466	2335	0.0	46.9	46.9
IC(0)	Comb.	476	2385	0.2	43.7	43.9
SGS	Comb.	155	780	0.0	21.4	21.4
MSF(1)	Comb.	125	630	1.9	13.2	15.1
MSF(4)	Comb.	44	245	8.2	6.5	14.7
MSF(2)	Comb.	53	270	3.3	6.2	9.5

Preconditioning Results: Normalized Laplacian

- Test graph: enron_bsl (67K rows, 507K nonzeros, 1 connected component)
- Solver=LOBPCG
- Nev = blocksize = 5
- Tol = 1e-5 (absolute)

Precond.	Laplacian	Iter.	Matvec.	Setup time	Iterate time	Total time
MSF(4)	Norm.	34	195	18.9	6.5	25.4
IC(0)	Norm.	270	1355	0.2	22.8	23.0
None	Norm.	250	1255	0	20.0	20.0
MSF(2)	Norm.	42	215	7.9	6.2	14.1
MSF(1)	Norm.	102	515	2.0	10.7	12.7
SGS	Norm.	87	440	0.0	11.8	11.8

Conclusions

- We can compute eigenvalues of graph Laplacians of order 10^6 on a desktop using Trilinos/Anasazi
- No clear winner among {BKS, LOBPCG, IRTR}, but LOBPCG appears most consistent
- Normalized Laplacians are better computationally
- Preconditioning is essential for combinatorial Laplacian, also helps for normalized.
- MSF(k) is a simple but effective preconditioner
- Future work: Larger problems on parallel computers
  - Need to revisit preconditioners (domain decomposition?)