Support Tree Preconditioning Lecture 1: Laplacians, Embeddings, and Support

Stephen Guattery
ICASE

March 28, 1997

Let A be symmetric, positive definite (s.p.d.). Consider the problem of solving

$$A\mathbf{x} = \mathbf{b}$$

using the conjugate gradient method. How fast does the approximate solution converge?

One upper bound on the number of iterations needed for convergence uses the *spectral condition number* κ. For a symmetric positive definite matrix A,

$$\kappa = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)},$$

the ratio of the largest and smallest eigenvalues.

The bound on convergence is

$$||\mathbf{e}_{(i)}||_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^i ||\mathbf{e}_{(0)}||_A,$$

where $||\mathbf{e}_{(i)}||_A$ is the energy norm of the error at the i^{th} iteration. This implies that, to reduce the initial error energy norm by a factor of ϵ, $i \leq \lceil \frac{1}{2}\sqrt{\kappa}\ln\left(\frac{2}{\epsilon}\right)\rceil$ iterations are required.

1 Preconditioned Conjugate Gradient

Idea: modify $A\mathbf{x} = \mathbf{b}$ to get faster convergence.

Let B be a symmetric positive definite matrix. Then any solution to

$$B^{-1}A\mathbf{x} = B^{-1}\mathbf{b} = \mathbf{b}'$$

is also a solution to the original problem. B is called a *preconditioning matrix* or *preconditioner*.

Useful preconditioners have the following properties:

- $\kappa(B^{-1}A)$ should be substantially smaller than $\kappa(A)$.
- B should be easy to solve. Note that B^{-1} could be dense even if B is sparse, so we don't want to form the inverse explicitly. However, we will need to compute $B^{-1}\mathbf{y}$, which can be done by solving for $\mathbf{z}$ in $B\mathbf{z} = \mathbf{y}$.

(Note that the product $B^{-1}A$ might not be symmetric. While there is not time today to cover the details, the conjugate gradient can be applied to the similar system $B^{-1/2}AB^{-1/2}$, which is symmetric. Further, the algorithm can be performed in such a way that the similar system need not be constructed, and only the preconditioner B need be used. See, e.g., Section 10.3 in Golub and Van Loan.

Note also that $B^{-1}A$ will be positive definite. You can prove this as an exercise, using the similar matrix given in the previous paragraph.)

But, how can we estimate $\kappa(B^{-1}A)$? Note that, since both A and B are s.p.d., we have

$$\lambda_{\min}(B^{-1}A) = \frac{1}{\lambda_{\max}((B^{-1}A)^{-1})} = \frac{1}{\lambda_{\max}(A^{-1}B)}. \tag{1}$$

Note that in this reciprocal form, an upper bound on $\lambda_{\max}(A^{-1}B)$ will give a lower bound on $\lambda_{\min}(B^{-1}A)$. Thus, if we can develop techniques for showing upper bounds on the largest eigenvalue of $B^{-1}A$, we can switch the role of A and B, and use the techniques to get the lower bound as well.

The problem of bounding $\lambda_{\max}(B^{-1}A)$ from above is the topic for rest of the lecture. We will first cover some general ideas, then examine a method that works for generalized Laplacian matrices.

2 The Generalized Eigenvalue Problem

The problem of finding λ's and $\mathbf{x}$'s that satisfy $A\mathbf{x} = \lambda B\mathbf{x}$ is called the *generalized eigenvalue problem.* For simplicity, today we will only consider the case where both A and B are s.p.d. Thus,

$$A\mathbf{x} = \lambda B\mathbf{x} \quad \leftrightarrow \quad B^{-1}A\mathbf{x} = \lambda\mathbf{x}. \tag{2}$$

3 The Support Lemma

Let A and B be symmetric positive definite matrices, and let τ be a real number.

Lemma 3.1 [Axelsson] *If $\tau B - A$ is positive semidefinite, then $\lambda_{\max}(B^{-1}A) \leq \tau$.*

Proof: Let $\mathbf{u}$ be an eigenvector of $\lambda_{\max}(B^{-1}A)$. By (2), $A\,\mathbf{u} = \lambda B\,\mathbf{u}$. Starting from the assumption that $\tau B - A$ is positive semidefinite, we can deduce the following:

$$\begin{aligned} 0 &\leq \mathbf{u}^T(\tau B - A)\mathbf{u} \\ &= \tau\mathbf{u}^T B\,\mathbf{u} - \mathbf{u}^T A\,\mathbf{u} \\ &= \tau\mathbf{u}^T B\,\mathbf{u} - \lambda\mathbf{u}^T B\,\mathbf{u} \\ &= (\tau - \lambda)\mathbf{u}^T B\,\mathbf{u} \end{aligned}$$

Since $\mathbf{u}^T B\,\mathbf{u} > 0$ (B is positive definite), $\tau - \lambda \geq 0$.
□
Definition: The *support* $\sigma(A, B)$ of matrix B for matrix A is

$$\min\{\tau : \tau B - A \text{ is positive semidefinite}\}$$

3.1 A Way to Apply the Support Lemma

Note that, by the Support Lemma, $\lambda_{\max}(B^{-1}A) \leq \sigma(A, B)$. Note also that, by the Support Lemma and (1), $1/\lambda_{\min}(B^{-1}A) = \lambda_{\max}(A^{-1}B) \leq \sigma(B, A)$. Thus

$$\kappa(B^{-1}A) = \frac{\lambda_{\max}(B^{-1}A)}{\lambda_{\min}(B^{-1}A)} \leq \sigma(A, B) \cdot \sigma(B, A).$$

We can use the Support Lemma to find upper bounds on $\sigma(A, B)$ and $\sigma(B, A)$.

Linearity is the key to using the Support Lemma. Assume that we can decompose A into k positive semidefinite pieces $A_1, A_2, \ldots, A_k$ such that $\sum_{i=1}^k A_i = A$. Likewise, suppose B can be decomposed into k positive semidefinite pieces. Assume that we have a set $\{\tau_1, \tau_2, \ldots, \tau_k\}$ such that $\tau_i B_i - A_i$ is positive semidefinite for all i. Let $\tau^* = \max_i \tau_i$. Then $\tau^* B_i - A_i$ is positive semidefinite for all i, and

$$\sum_{i=1}^k (\tau^* B_i - A_i) = \tau^* \sum_{i=1}^k B_i - \sum_{i=1}^k A_i = \tau^* B - A.$$

By linearity, $\tau^* B - A$ is positive semidefinite.

One way to use the support lemma is to find such a decomposition of A and B such that the subproblems are easier to solve. We can then choose the maximum value from the τ_i's, and use that as our bound.

Can we always find such a decomposition? The answer is *yes* for certain classes of matrices.

4 Generalized Laplacian Matrices

Definition: L is a *generalized Laplacian matrix* if and only if:

- L is symmetric;
- all diagonal entries $l_{ii} > 0$;
- all off-diagonal entries $l_{ij} \leq 0$; and
- for every row i, $l_{ii} \geq \sum_{j \neq i} |l_{ij}|$.

To insure positive definiteness, we will assume that the matrix is irreducible, and that the last inequality is strict for at least one row.

4.1 Generalized Laplacians and Graphs

Generalized Laplacians correspond to graphs with positive edge weights according to the following rules:

- Each row (or column) corresponds to a vertex.
- Nonzero off-diagonal entries correspond to edges. That is, for $i \neq j$, if $l_{ij} \neq 0$, then there is an edge between vertices v_i and v_j. The weight of this edge is $|l_{ij}|$.
- Diagonal entry l_{ii} is the sum of the weights of the edges incident to vertex v_i.
- If a diagonal entry is greater than the sum of the incident edge weights, there is an additional edge from that vertex to an implicit zero-valued boundary vertex.

When necessary, we use the following notation to relate graphs and matrices: For a Laplacian L, the associated graph is $G(L)$. The generalized Laplacian of a graph G is denoted $L(G)$.

The following property is a useful consequence of interpreting a Laplacian as a graph: Let L be a Laplacian with associated graph G. Let $w_{ij} = -l_{ij}$ be the weight of edge (v_i, v_j). Then for all $\mathbf{x}$,

$$\mathbf{x}^T L \mathbf{x} = \sum_{(v_i,v_j) \in E(G)} w_{ij}(x_i - x_j)^2. \tag{3}$$

5 Graph Embeddings

Let G be a connected graph, and let H be a graph such that $V(H) \subset V(G)$. To embed H into G, for each edge $(v_i, v_j) \in E(H)$ we specify a simple path in G from v_i to v_j. We apply the following definitions to embeddings:

- The *weight of a path* is the weight of the corresponding edge in H.
- The *congestion* of an edge $e \in G$ is the sum of the weights of the paths that include e divided by the weight of e. (In the unweighted case, this is just the number of paths that include e.) The congestion of the embedding is the maximum edge congestion taken over all edges in G.
- The *dilation* of an edge f in H is the length (the number of edges) in f's path in the embedding. The dilation of the embedding is the maximum dilation taken over all edges in H.

An example of an embedding is given in Figure 1 below.

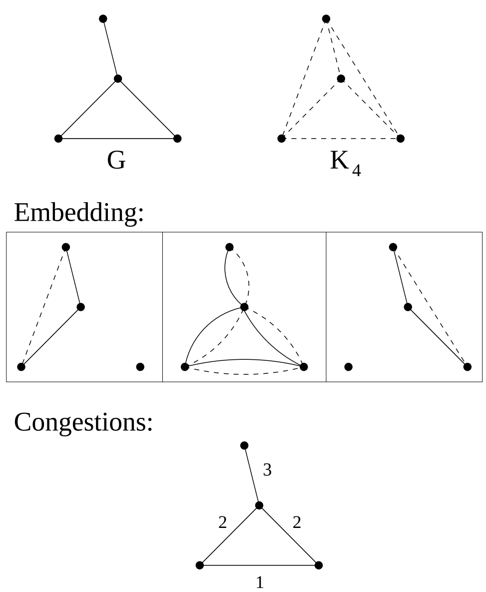

Figure 1: Embedding Example

5.1 Decompositions from Embeddings

These embeddings can be used to decompose the matrix $\tau B - A$. Let k be the number of edges in $G(A)$, the graph corresponding to the generalized Laplacian

A. We will decompose $\tau B - A$ into pieces $\tau B_i - A_i$, where A_i is the Laplacian of the graph on $V(G(A))$ consisting of only edge e_i of $G(A)$. B_i is the Laplacian of the (appropriately weighted) corresponding path in $G(B)$.

To determine the appropriate weighting of the paths, note that an edge e in $G(B)$ may show up in multiple paths in the decomposition. We must choose the weights of e on these paths such that they sum to w_e, the weight of e.

Let c_e be the congestion of edge e in B; let w_f be the weight of edge f in A. Assume that the path for f includes e. The amount of weight from w_e assigned to the path associated with f is

$$\frac{w_f}{c_e}.$$

(In the unweighted case, this will be just $\frac{1}{c_e}$.)

There is no problem if there are edges in $G(B)$ that do not occur in any path. These edges can be separated out into a generalized Laplacian, which is positive semidefinite.

An example of such a decomposition is given in Figure 2 below.

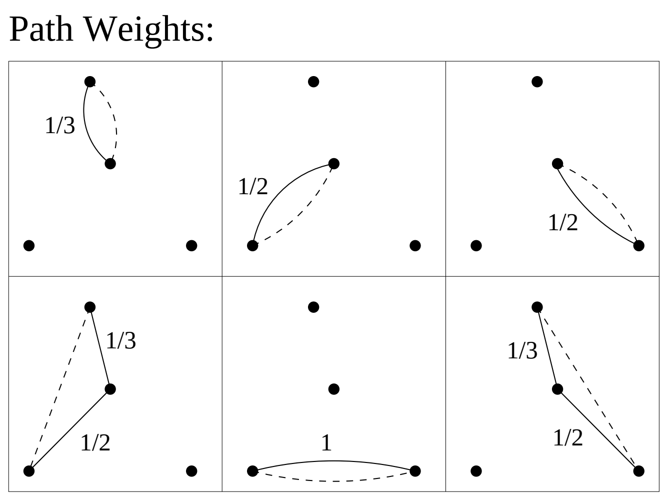

Figure 2: Decomposition Example

6 The Path Problem

We've reduced the problem of finding a τ that is an upper bound of $\sigma(A, B)$ to the problem of computing τ_i's for a number of problems that consist of supporting an edge with a path, then setting τ to the maximum over the τ_i's. The problems involving paths are much easier to solve than the original.

Now consider the problem of a path p supporting an edge e. Assume the dilation for edge e is j, and let w_e be the weight of e in $G(A)$. For simplicity, we will reindex the vertices from 1 to $j+1$ according to their order along the path. Let c_i be the congestion on edge i of the path. The weight of edge i in the path is w_e/c_i.

Using (3), we can state the path problem as follows: Choose τ_p such that

$$\tau_p \sum_{i=1}^{j} \frac{w_e}{c_i}(x_i - x_{i+1})^2 \geq w_e(x_1 - x_{j+1})^2,$$

or, cancelling the common factor w_e,

$$\tau_p \sum_{i=1}^{j} \frac{1}{c_i}(x_i - x_{i+1})^2 \geq (x_1 - x_{j+1})^2.$$

6.1 The Path Problem and Electrical Circuits

CLAIM: The path problem looks like the power problem in a series resistive circuit. In particular, the entries in $\mathbf{x}$ correspond to voltages at path nodes. The question is, given voltages at the ends of the circuit, what voltages at the internal nodes produce the minimum power dissipation?

Laplacians can be used to represent resistive circuits, where the off-diagonal entries represent *conductances* between nodes in the circuit (conductances are the reciprocals of resistances). We construct a series resistive circuit corresponding to path p as follows: For edge i on the path, assign a resistor with resistance $r_i = c_i$ (i.e., resistance corresponds to congestion). Let

$$r_p = \sum_{i=1}^{j} r_i = \sum_{i=1}^{j} c_i.$$

The following theorem is well-known:

Theorem 6.1 *For any* $\mathbf{x}$*,*

$$r_p \sum_{i=1}^{j} \frac{1}{c_i}(x_i - x_{i+1})^2 \geq (x_1 - x_{j+1})^2$$

Proof: We can rewrite the left-hand side of the theorem statement in terms of congestions as follows:

$$\sum_{i=1}^{j} c_i \sum_{i=1}^{j} \frac{1}{c_i}(x_i - x_{i+1})^2.$$

Rewriting slightly and applying Cauchy-Schwarz gives the following:

$$\sum_{i=1}^{j} \left(\sqrt{c_i}\right)^2 \sum_{i=1}^{j} \frac{1}{\left(\sqrt{c_i}\right)^2} (x_i - x_{i+1})^2 \geq \left(\sum_{i=1}^{j} \sqrt{c_i} \frac{1}{\sqrt{c_i}} (x_i - x_{i+1}) \right)^2 = (x_1 - x_{j+1})^2.$$

The last inequality follows because the sum telescopes. This proves the theorem. □

Thus, for any path problem, it is sufficient to set $\tau_p = r_p$, the sum of the congestions along the path.

6.2 A Simplification

To compute τ in a simpler way, we note the following two facts:

- For any path p in the embedding, setting all congestions equal to the maximum congestion on the path only increases τ_p, which remains an upper bound.
- If all congestions on the path have value c, then the value we compute for τ_p is the product of the path length times c.

Thus, the τ_p we compute for any path p is less than the product of the path's length times the maximum congestion on any path edge. This product is bounded above for any path by the product of the embedding's congestion times its dilation.

Though the product of the congestion of the embedding times dilation can be greater than the value of τ derived by looking at all paths, in many cases the difference is not more than a constant factor. Since this product is often much easier to compute, it is a valuable simplification.