Graph coloring using eigenvalue decomposition

Ashish Sangwan and Arvind Batra

December 3, 2007

Spectral techniques have been applied to come up with efficient solutions to combinatorial problems like graph partitioning, min-cut, coloring. We are investigating further the applications of spectral techniques to graph coloring, a well-known NP-complete problem.

Negative eigenvalues of adjacency matrix give useful information about the connectivity of the original graph. Most negative eigenvalue has been used in [1], [2] to devise algorithms for graph coloring. Alon et al [2] correctly color 3-colorable random graphs with high probability by using eigenvetors corresponding to two smallest eigenvalues. We extend the theorem 4 in [1] to regular graphs of any degree in section 1. [1] gives an approximate 2-coloring algorithm using most negative eigenvalue. We would like to obtain an upper bound on number of intra-partition edges obtained in 2-partitions. When we 2-color a graph and try to maximize the number of correctly color edges, we note that essentially we are finding a maximum cut of an unweighted graph. Moreover, when we remove the wrongly colored edges (coloring done using 2-coloring algorithm) from the graph, we get a bipartite subgraph of the graph and we want to find a bipartite subgraph with maximum number of edges. Erdos et al [3] reproves the edwards result on the minimum number of edges in a bipartite subgraph of a graph. We compare result obtained by our 2-coloring algorithm with this lower bound in section 3.

Karger et al. [5] use semi-definite programming (SDP) to color arbitrary graphs. [4] applies SDP with triangle inequality constraints to devise approximation algorithms for sparsest cut, graph conductance and related problems. Another approach could be to assign coordinates to vertices of graph in R^n (as done in [4]) and then project the vertices to lower dimension.

We note that problem of graph coloring is closely related to well-known problem of clustering. Particularly, if we relax the condition of correct coloring and say that we want a coloring of vertices such that large number of edges are correctly colored and upper bound the number of wrongly colored edges[1]. In this setting, we say that two nodes are similar if they are not adjacent. If we cluster vertices using the similarity defined just now, we expect to get clusters of vertices with very low number of edges inside the cluster. [6] gives a recursive algorithm for clustering using second eigenvector of similarity matrix. In coloring case, similarity graph is essentially the complement of the given graph. Therefore, insted of using the second eigenvalue, we use largest eigenvalue of laplacian matrix of given graph. We discuss this recursive algorithm in detail in section 2. In section 3, the proposed algorithm is compared with some existing algorithms and some caveat graphs are identified where proposed algorithm performs "poorly". We conclude the report in section 4.

1 Theorem 4 of [1] is extended for k-regular graphs

Theorem 1. *Suppose G is a connected k-regular graph, and let its adjacency matrix be A having eigenvalues $\lambda_1 \geq ... \geq \lambda_n$ and corresponding eigenvectors $u_1, ..., u_n$. Let u_{jk} be the j-th component of eigenvector u_k. If $|\lambda_n| \geq \lambda_2$ and there is a constant η, such that $1 \leq \eta \leq \sqrt{\frac{2-\rho}{1-\rho}}$ where $\rho = k/n$, and $|\frac{u_{in}}{u_{jn}}| \leq \eta$ for all i and j, then the number of edges satisfying the coloring condition are more than $[\frac{2-\eta^2}{1+\eta^2} + \frac{\rho(\eta^2-1)}{1+\eta^2}]$ times the total number of edges in G.*

[1] An edge is wrongly colored if it's end-point get same color.

In the last theorem, we assumed an upper bound on ratio (η) of magnitude of maximum and minimum component of eigenvector corresponding to most eigenvalue. That is not a good assumption as theorem can not be applied to all regular graphs. Ideally we would like to get a generic result which puts no upper bound on η and indeed in the following theorem, we give a generic bound on the number of correctly colored edges which does not impose any restriction on η. It is easy to see that lower bound given in following theorem is better than what we got in previous theorem.

Theorem 2. *Suppose G is a connected k-regular graph, and let its adjacency matrix be A having eigenvalues $\lambda_1 \geq ... \geq \lambda_n$ and corresponding eigenvectors $u_1, ..., u_n$. Let u_{jk} be the j-th component of eigenvector u_k. If $|\lambda_n| \geq \lambda_2$ and there is a constant η, such that $|\frac{u_{in}}{u_{jn}}| \leq \eta$ for all i and j, then twice the number of edges satisfying the coloring condition is more than $[\frac{(a+b)}{(1+\eta^2)} + \frac{(-\lambda_n)}{|u_{min}|^2(1+\eta^2)}]$ where u_{min} denotes the component of minimum magnitude of u_n.*

2 A recursive algorithm for coloring

Throughout this section, we use laplacian matrix representation L of given graph G. First, we describe the $MostNeg$ algorithm proposed in [1]. Algorithm $MostNeg$ for approximately 2-coloring of graph.

1. Calculate the eigenvector v corresponding to highest eigenvalue of L.
2. Map each vertex to corresponding component in v.
3. Partition the vertices into two groups according to the sign of mapped value.

$Mostneg$ gives two partition of vertices. Each partition gets a same color. We get an approximate coloring of graph with edges in same partition wrongly colored. Now we suggest that, apply $MostNeg$ recursively on each partition till a correct coloring is obtained. Algorithm $RecCut$ is as follows:

1. Calculate the eigenvector v corresponding to the highest eigenvalue of L.
2. Partition the vertices based on the signs of the eigenvector v.
3. Recurse on the pieces induced by the partition.

3 Simulation results

We calculate the performance of four coloring algorithms for two class of graphs. Four coloring algorithms are as follows:

- MostNeg: Approximately colors graph with 2 colors using the eigenvector corresponding to the most negative eigenvalue of adjacency matrix.
- RecCut: Correct coloring of graph using algorithm mentioned in section 2.
- MaxDeg: Trivial upper bound on the number of colors required to color the graph correctly.
- Alon: [2] first uses a spectral algorithm to get an approximate coloring and then refines it in subsequent phases. We analyze the coloring obtained by the output of spectal algorithm only.

Two classes of graphs considered:

- Random 3-colorable graphs: Vertices are partitioned into three color classes of same size. There are no intra-partition edges and inter-partition edges are there with probability p. p is varied from 0.1 to 1.0.
- Nearly bipartite graphs: Bipartite graphs are perturbed by adding intra-partition edges with probability p. p is varied from 0.1 to 0.5. Inter-partition edges are present with fixed probability 0.5.

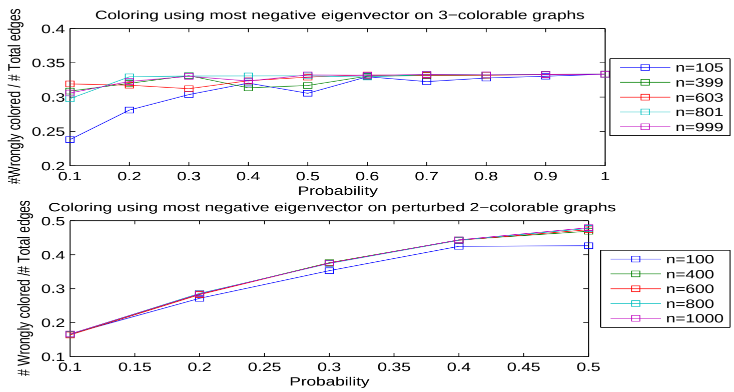

Figure 1: Performance of $MostNeg$

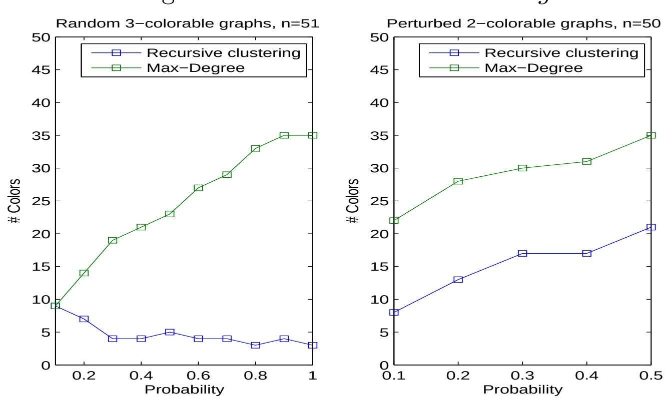

Figure 2: Performance of $RecCut$ versus $MaxDeg$

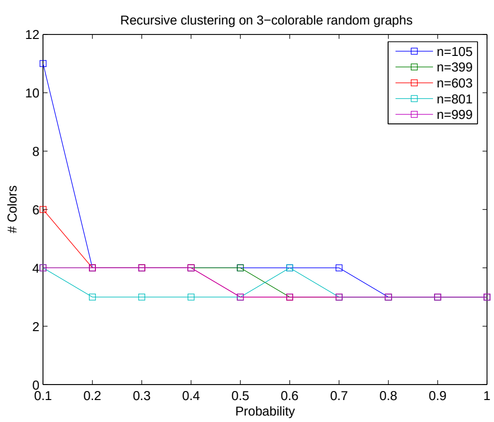

Figure 3: Performance of $RecCut$ for random 3-colorable graphs

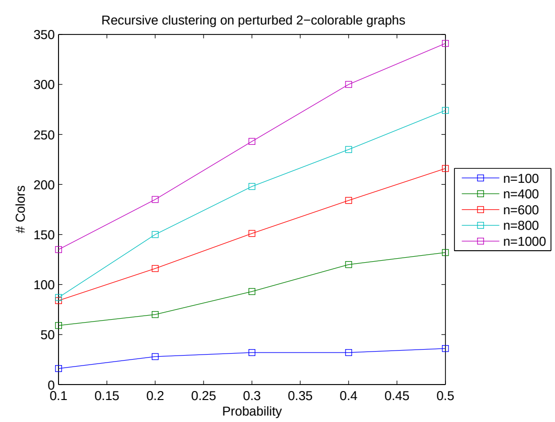

Figure 4: Performance of $RecCut$ for nearly bipartite graphs

Observations

1. In figure 1, we observe that $MostNeg$ algorithm performs independent of the number of vertices in graph. For random 3-colorable graphs, it is independent of both number of vertices and edges in graph. Moreover, fraction of correctly colored edges is always (for random 3-colorable graphs and nearly bipartite graphs) more than the lower bound provided by edwards formula [3].

2. *RecCut* performs better on dense graphs than on sparse graphs as seen in figure 2.

3. *RecCut* out-performs $MaxDeg$ (figure 2), the trivial upper bound on coloring in both classes of graphs under consideration. However, *RecCut* uses more colors than used by $MaxDeg$ for peterson graph and knesar graph with $n = 12, k = 5$ which means that graph has 792 vertices.

4 Conclusion and future work

We observe that our proposed algorithm recursive clustering performs well in practice. It is closely related to the spectral graph partitioning algorithm mentioned in [6]. Currently, no bounds exist which tell us about the number of partitions resulting after applying the spectral partitioning algorithm. In our algorithm, number of partitions equals the number of colors used to correctly color the graph. Hence, it is still an open question to get a theoretical bound on number of colors.

We also observe that eigenvector corresponding to most negative eigenvalue of adjacency matrix of graph gives nice information about clustering the graph into two partitions with low intra-partition bounds. Instead of using just one eigenvector, devise algorithm which uses more eigenvectors to get a better coloring. The conjecture mentioned in [1], that a correct coloring can be obtained by considering all negative eigenvalues still remains open.

References

[1] Bengt Aspvall and John R. Gilbert, Graph Coloring Using Eigenvalue Decomposition, Technical Report, Cornell University, 1983.

[2] Noga Alon and Nabil Kahale, A Spectral Technique for Coloring Random 3-Colorable Graphs, SIAM Journal on Computing, 1997.

[3] P. Erdos, A. Gyarfas and Y. Kohayakawa, The size of the largest bipartite subgraphs, Discrete Mathematics (1997), 267-271.

[4] Sanjeev Arora, Satish Rao, Umesh Vazirani, Expander Flows, Geometric Embeddings and Graph Partitioning, Proceedings of Symposium on the Theory of Computing, 2004.

[5] David Karger, Rajeev Motwani, Madhu Sudan, Approximate Graph Coloring by Semidefinite Programming, IEEE Symposium on Foundations of Computer Science, 1994.

[6] D. Cheng, R. Kannan, S. Vempala and G. Wang, A Divide-and-Merge Methodology for Clustering, Proc. of the ACM Symposium on Principles of Database Systems, 2005.