Appl. Comput. Harmon. Anal. • • • (• • • •) • • •–• • •

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Case Studies

A fast algorithm for manifold learning by posing it as a symmetric diagonally dominant linear system ☆

Praneeth Vepakomma [a,b,*], Ahmed Elgammal [c]

[a] *Department of Statistics, Rutgers, The State University of New Jersey, New Brunswick, USA*
[b] *Motorola Solutions, USA*
[c] *Department of Computer Science, Rutgers, The State University of New Jersey, New Brunswick, USA*

ARTICLE INFO

Article history:
Received 23 July 2015
Received in revised form 5 October 2015
Accepted 11 October 2015
Available online xxxx
Communicated by Charles K. Chui

Keywords:
Fast manifold learning
Symmetric diagonally dominant (SDD) system
Fast sparse Cholesky decomposition
Fast SDD solver
Linearly constrained optimization

ABSTRACT

We present a fast manifold learning algorithm by formulating a new linear constraint that we use to replace the weighted orthonormality constraints within Laplacian Eigenmaps; a popular manifold learning algorithm. We thereby convert a quadratically constrained quadratic optimization problem into a simpler formulation that is a linearly constrained quadratic optimization problem. We show that solving this problem is equivalent to solving a symmetric diagonally dominant (SDD) linear system which can be solved very fast using a combinatorial multigrid (CMG) solver. In addition to this we also suggest another method that can exploit any sparsity within the graph Laplacian matrix via a fast sparse Cholesky decomposition to produce an alternative solution in addition to the SDD based method. We compare the improvements in run-times using both our SDD system based method and our fast sparse Cholesky decomposition based method against the well known Nystrom method based fast manifold learning and present competitive results.

1. Introduction

The problem of manifold learning is concerned with finding low-dimensional representations of high-dimensional data. There has been extensive research in the recent years in developing various manifold learning techniques like Diffusion Maps [1], Hessian Eigenmaps [2], Laplacian Eigenmaps [3], Semidefinite Embedding [4], Locally Linear Embedding [5], ISOMAP [6], Continuum Isomap [7], Local Tangent Space Alignment [8], Maximum Variance Unfolding [9], t-distributed Stochastic Neighborhood Embedding [10]. As part of the solution to the manifold learning problem, most of these techniques require a spectral decomposition. The authors in [11–14] have quite successfully employed the Nystrom method in the speeding-up

☆ This is only an example.

* Corresponding author.
E-mail addresses: praneeth@scarletmail.rutgers.edu (P. Vepakomma), elgammal@cs.rutgers.edu (A. Elgammal).

http://dx.doi.org/10.1016/j.acha.2015.10.004
1063-5203/

of the spectral decomposition within this setting. The goal of this paper is to formulate a modified objective function that is much simpler to optimize than the one's used in the popular techniques referred above along with providing an efficiently computable solution to the modified problem and comparatively benchmarking it's run-times across real-life datasets in comparison to the existing Nystrom based methods. The following is the structure of this paper: In Section 2, we examine the objective function of Laplacian Eigenmaps which is a Quadratically Constrained Quadratic Optimization problem. We then formulate a new constraint that is instead linear and can replace the quadratic constraint used in Laplacian Eigenmaps. We then show that the solution of this newly formulated problem can be obtained by solving a symmetric diagonally dominant (SDD) linear system. In addition to this new SDD solution, we also provide a second solution which can be obtained by a fast sparse Cholesky decomposition that can exploit the sparsity of the graph Laplacian matrix involved in the quadratic objective function. In Section 3, we provide comparative results of runtimes as well as of the quality of our solution over various real-life datasets, both visually and quantitatively. We compare our SDD based solution as well as our alternate sparse Cholesky decomposition based solution with the baseline technique of the Nystrom method based manifold learning and show highly competitive improvements in speed while maintaining a similar quality of the solution.

2. Linearly constrained embedding

Consider the real matrix $X_{n\times p}$ to be a high-dimensional input dataset consisting of n points in p dimensions. We now describe some notation used in the objective function of Laplacian Eigenmaps, a popular manifold learning algorithm. We denote L to be a symmetric graph Laplacian matrix constructed from a graph with an adjacency matrix W, that has been obtained by applying a Gaussian kernel function to the points (rows) in X as $W_{ij} = \frac{exp(-d_{ij}^2(X))}{\sigma}$ with $d_{ij}^2(X)$ denoting the squared Euclidean distance between the rows i and j of the matrix X. We use D to denote the degree matrix corresponding to the above mentioned adjacency matrix and restate the well-known identity of $L = D - W$. Using this notation, the objective in Laplacian Eigenmaps can be reiterated as the problem of finding a real matrix $Z_{n\times d}$ for a chosen $d < p$ that minimizes the following:

$$\underset{\text{s.t. } Z^TDZ=I}{\operatorname{argmin}} \; Tr[Z^TLZ] \tag{1}$$

where I denotes the Identity matrix. One of the main reasons for using this particular quadratic constraint, $Z^TDZ = I$ is in order to prevent a zero matrix (all entries being zero) Z^* from being the optimal minimizer to this problem. Our first contribution in this paper is a linear constraint that we propose in order to prevent a zero matrix from being a minimizer of this objective. On expanding the trace $Tr[Z^TLZ]$, we can see that this term is exactly the same as $\sum_{ij} W_{ij}d_{ij}^2(Z)$ with $d_{ij}^2(Z)$ denoting the squared Euclidean distance between the rows i and j of the matrix Z. To avoid confusion we restate that this W is the adjacency matrix that corresponds to the graph Laplacian matrix L in this objective. Our linear constraint hence has to prevent a solution where all the rows of Z are not exactly the same leading to $d_{ij}(Z)$ being zero $\forall i, j$. If such a linear constraint can be formulated, it's linearity makes it easier to practically enforce it due to the quadratic nature of the loss function thereby producing a much simpler optimization problem.

Definition 2.1 *(Row-unique matrix).* A matrix M is row-unique, if all the rows in the matrix are distinct.

Proposition 2.1. *For any row-unique matrix $M_{n\times p}$, and for any given Laplacian matrix $L_{n\times n}$, if $Tr(Z^TLM) \neq 0$, then there exist at least two rows in $Z_{n\times p}$, that are distinct.*

Proof. $Tr(Z^TLM) = \sum_{i,j} W_{ij}\phi_{ij}(Z, M)$, where $\phi_{ij}(Z, M) = \sum_{a=1}^{p}(Z_{ia} - Z_{ja})(M_{ia} - M_{ja})$. Hence, for a row unique M, there exist at least two rows in Z, such that $Z_{i.} \neq Z_{j.}$ in order to satisfy the inequality on $Tr(Z^TLM)$. Note that, $\phi_{ij}(Z, Z) = d_{ij}^2(Z)$. □

Please cite this article in press as: P. Vepakomma, A. Elgammal, A fast algorithm for manifold learning by posing it as a symmetric diagonally dominant linear system, Appl. Comput. Harmon. Anal. (2015), http://dx.doi.org/10.1016/j.acha.2015.10.004

We define our proposed linear constraint as follows:

$$Tr(Z^T S\Upsilon) = \nu \tag{2}$$

where $\nu \neq 0$, $S = \text{Laplacian}(\mathbf{1}_{n\times n})$ is a Laplacian formed over an adjacency matrix of all 1's and Υ denotes any row-unique matrix. As far as the optimization is concerned, Υ is a fixed matrix and so is the scalar ν fixed. The following is the total loss function, $T(Z, \lambda)$ obtained when our proposed linear constraint is combined with the convex objective function with λ being a positive multiplier over the constraint:

$$T(Z, \lambda) = Tr(Z^T LZ) + \lambda \left[Tr(Z^T S\Upsilon) - \nu \right] \tag{3}$$

The gradient, is given by:

$$\bigtriangledown T(Z, \lambda) = 2LZ + \lambda S\Upsilon \tag{4}$$

We get the following solution, by setting the gradient equal to zero.

$$Z^* = -\frac{\lambda}{2} L^+ S\Upsilon \tag{5}$$

where the inverse on L is the pseudoinverse. Now upon substituting this for Z in $Tr(Z^T S\Upsilon) = \nu$, we get the following update, for the multiplier:

$$\lambda = \frac{-2\nu}{Tr(\Upsilon^T S L^+ S\Upsilon)} \tag{6}$$

2.1. Faster algorithm using sparsity of Laplacian matrix

Using the sparsity of L, we can compute a Cholesky factor R using the 'fast Cholesky decomposition for sparse matrices' as introduced in [15–19] with implementations in [20–22] such that $L = R^T R$ and then we can solve for a Y using the lower-triangular Cholesky factor matrix R^T as below:

$$R^T Y = S\Upsilon \tag{7}$$

Now upon solving for a G using the upper-triangular Cholesky factor matrix R as below:

$$RG = Y \tag{8}$$

gives us a faster way of obtaining the embedding Z by exploring the sparse structure in L as:

$$Z = \lambda G \text{ where } \lambda = -2\nu / \|Y\|_F^2 \tag{9}$$

2.2. Even faster algorithm by solving a symmetric diagonally dominant system

As L is a symmetric diagonally dominant matrix, our solution in eqn. (5) is equivalent to solving a symmetric diagonally dominant linear system. There has been absolutely excellent recent research resulting in algorithms that can solve such systems really fast as in [24–30,32] with the fastest algorithm to date requiring just nearly $m \log^{1/2} n$ time for the solution [32]. This leads to an SDD solution via the usage of our linear constraint based formulation of manifold learning thereby resulting in highly competitive run-times.

Please cite this article in press as: P. Vepakomma, A. Elgammal, A fast algorithm for manifold learning by posing it as a symmetric diagonally dominant linear system, Appl. Comput. Harmon. Anal. (2015), http://dx.doi.org/10.1016/j.acha.2015.10.004

Table 1
Comparison of runtimes.

Dataset	Nystrom IKLR	Ours (Sparse Cholesky)	Ours (SDD)
SARCOS Robot Arm: 10 000 by 21	231	116	**0.0325**
Regulated genes: 4824 by 14	22.901	15.157	**0.0168**
USPS handwritten digits: 4649 by 256	39.497	22.604	**0.0152**
Frey Human Faces: 2000 by 560	3.405	1.323	**0.0074**
Olivetta Human Faces: 400 by 4096	0.029	0.013	**0.0090**
UMIST Human Faces: 575 by 10 304	0.021	0.008	**0.0023**

3. Experimental results

In order to compare the run-times of our proposed technique we use the well-known Improved Nystrom Method for Low-Rank Approximations (Nystrom IKLR) [11,12] as our baseline to compare the run-times against each other. Approximation of eigenvectors within the spectral formulation of Laplacian Eigenmaps using the Nystrom method [13] has been a recent breakthrough in speeding up manifold learning algorithms and there has been extensive research in this direction recently in speeding up various low-rank kernel approximations also applied on an extensively large scale as in [14]. We use the code provided by the authors of IKLR, to run the Nystrom based Laplacian Eigenmaps and we note the run-times for various datasets in Table 1. We fix the adjacency matrices used in both the methods for any given dataset. We obtain results on 6 popular real-life, machine learning datasets by reducing the dimensionality in all the cases to 2. We first describe these datasets below.

3.1. Dataset description

We tested our algorithm on three well known human face machine learning datasets: Frey Human Faces: 2000 points of 560 dimensions, Olivetta Human Faces: 400 points of 4096 dimensions and UMIST Human Faces: 575 points of 10 304 dimensions which are collections of gray scale images (of levels 0–255) of dimensions 20 by 28 (stacked to a vector of 560 dimensions), 64 by 64 (stacked to 4096) and 112 by 92 (stacked to 10 304) respectively. We also tested it on 10 000 sampled points of data from the SARCOS Robot Arm which relates to an inverse dynamics problem for a seven degrees-of-freedom SARCOS anthropomorphic robot arm. The task here is to map from a 21-dimensional input space to a lower dimension. We also use a 4824 point dataset of temporary regulated genes obtained in the development of barley endosperm tissue over 14 time points (14 dimensions) from the bioinformatics experiment in [23]. We also use a 4649 point USPS images of handwritten digits dataset of handwritten digits of 7, 3, 0 in 256 dimensions.

We compare the run-times of our method against Nystrom IKLR in Table 1 showing the improvement in run-time on all the datasets upon using our two methods over the well-known Nystrom IKLR technique. We also show the results of the embeddings of the UMIST Images of Human Faces (10 304 dimensions), Olivetta Images of Human Faces (4096 dimensions) and USPS images of handwritten digits (7, 3, 0 digits in 256 dimensions) datasets in Figs. 1, 2 and 3 respectively. We can see similar faces as well as digits grouped together in these 2D solutions obtained from our algorithm.

In addition to comparing the run-times and looking at the visual quality of our solutions, we quantitatively compare the quality of our solutions with that of the solutions from the popular manifold learning technique of Diffusion Maps by using the quantitative measure of 'Local Continuity Meta Criterion (LCMC)' as described in [31]. We show the differences in LCMC obtained between the two techniques in Table 2, where we find that our method reasonably preserves the quality of the solution as in a way that is similar to the quality obtained from Diffusion Maps.

Please cite this article in press as: P. Vepakomma, A. Elgammal, A fast algorithm for manifold learning by posing it as a symmetric diagonally dominant linear system, Appl. Comput. Harmon. Anal. (2015), http://dx.doi.org/10.1016/j.acha.2015.10.004

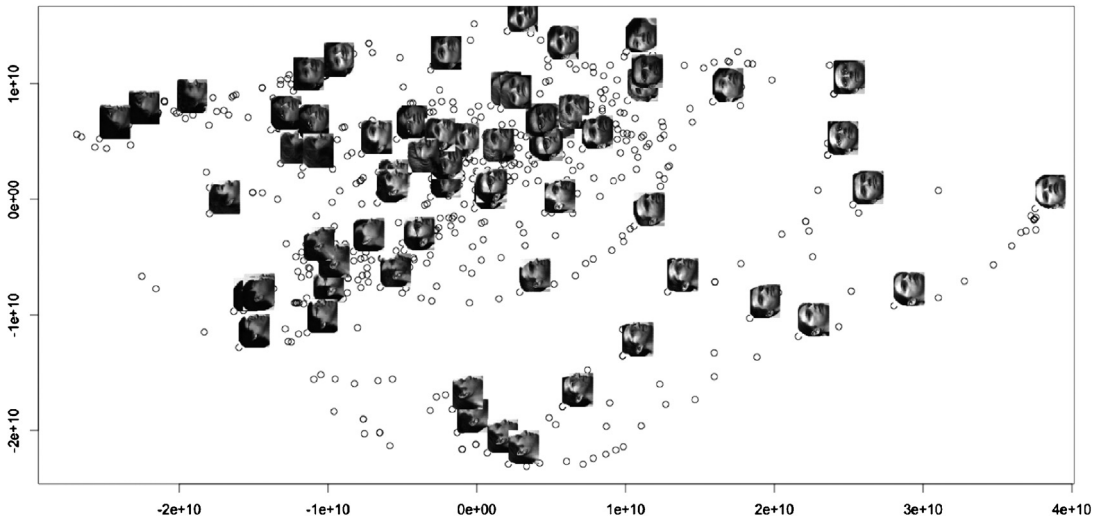

Fig. 1. UMIST Faces: 10 304 dimensional points embedded in 2D.

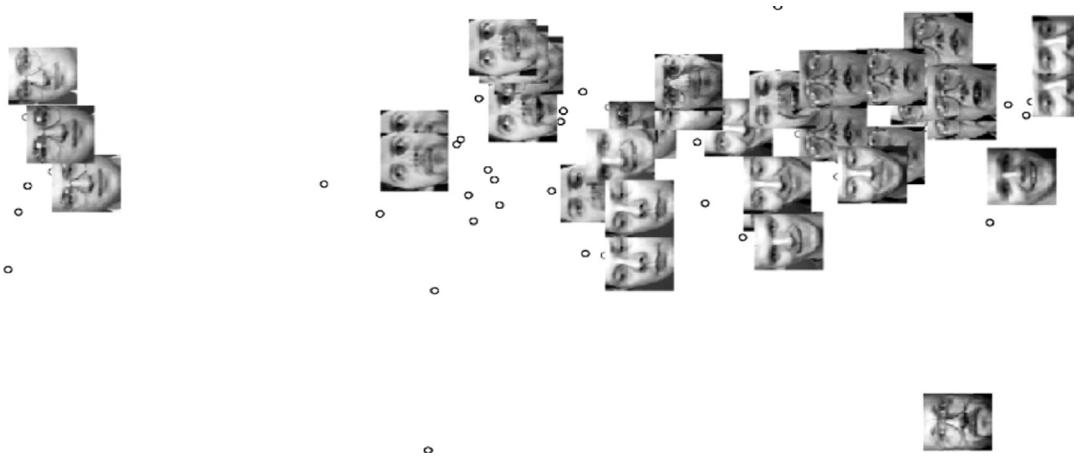

Fig. 2. Olivetta Faces: 4096 dimensional points embedded in 2D.

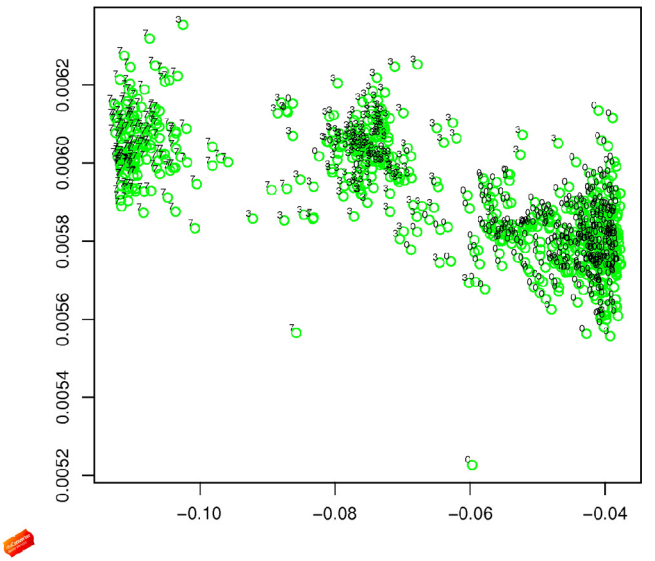

Fig. 3. USPS Digits: 256 dimensional images of handwritten digits embedded as points in 2D.

Please cite this article in press as: P. Vepakomma, A. Elgammal, A fast algorithm for manifold learning by posing it as a symmetric diagonally dominant linear system, Appl. Comput. Harmon. Anal. (2015), http://dx.doi.org/10.1016/j.acha.2015.10.004

Table 2
Difference of local continuity meta criterions.

Dataset	LCMC (diffusion maps) – LCMC (our SDD method)
SARCOS Robot Arm	0.08
Regulated genes	−0.12
USPS handwritten digits	0.03
Frey Human Faces	0.04
Olivetta Human Faces	−0.09
UMIST Human Faces	−0.1

4. Conclusion and future work

In this work, we provided a new formulation for manifold learning that is computationally very fast. We achieved highly competitive run-times on 6 real-life high-dimensional datasets in comparison to existing Nystrom based fast algorithms for manifold learning. As part of the future work, we would also like to investigate various combinatorial properties of our proposed zero-matrix preventing constraint with regards to the trade-off between the error observed versus the number of unique (non-overlapping) points produced in the result. We would also try to deploy and apply our algorithm on large scale learning problems as in [14] within a massively large-scale high-performance computing based industrial/laboratory deployment. We would also like to use the recently proposed graph-sparsification algorithms in [33–36] in order to be able to speedily generate the graph Laplacian matrices from the data as part of this deployment in the future.

References

[1] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, S. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data, Proc. Natl. Acad. Sci. 102 (2005) 7426–7431.
[2] David L. Donoho, Carrie Grimes, Hessian eigenmaps: locally linear embedding techniques for high-dimensional data, Proc. Natl. Acad. Sci. 100 (10) (2003) 5591–5596.
[3] Mikhail Belkin, Partha Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (June 2003) 1373–1396.
[4] Kilian Q. Weinberger, Lawrence K. Saul, Unsupervised learning of image manifolds by semidefinite programming, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2004.
[5] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[6] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 22 (2000) 2319–2323.
[7] Hongyuan Zha, Zhenyue Zhang, Continuum isomap for manifold learnings, Comput. Statist. Data Anal. 52 (2007) 184–200.
[8] Zhenyue Zhang, Hongyuan Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, SIAM J. Sci. Comput. 26 (1) (2004) 313–338.
[9] K.Q. Weinberger, L.K. Saul, An introduction to nonlinear dimensionality reduction by maximum variance unfolding, Association for Advancement of Artificial Intelligence, AAAI, 2006.
[10] L.J.P. van der Maaten, G.E. Hinton, Visualizing high-dimensional data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[11] Kai Zhang, Ivor W. Tsang, James T. Kwok, Improved Nystrom low rank approximation and error analysis, in: The 25th International Conference on Machine Learning, 2008.
[12] Kai Zhang, James T. Kwok, Clustered Nystrom method for large scale manifold learning and dimension reduction, IEEE Trans. Neural Netw. 21 (10) (2010) 1576–1587.
[13] Christopher Williams, Matthias Seeger, Using the Nystrom Method to Speed Up Kernel Machines, Adv. Neural Inf. Process. Syst., vol. 13, MIT Press, 2001, pp. 682–688.
[14] Ameet Talwalkar, Sanjiv Kumar, Henry Rowley, Large-scale manifold learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[15] T.A. Davis, W.W. Hager, Dynamic supernodes in sparse Cholesky update/downdate and triangular solves, ACM Trans. Math. Software 35 (4) (2009).
[16] Y. Chen, T.A. Davis, W.W. Hager, S. Rajamanickam, CHOLMOD, supernodal sparse Cholesky factorization and update/downdate, ACM Trans. Math. Software 35 (3) (2009).
[17] T.A. Davis, W.W. Hager, Row modifications of a sparse Cholesky factorization, SIAM J. Matrix Anal. Appl. 26 (3) (2005) 621–639.
[18] T.A. Davis, W.W. Hager, Multiple-rank modifications of a sparse Cholesky factorization, SIAM J. Matrix Anal. Appl. 22 (4) (2001) 997–1013.

Please cite this article in press as: P. Vepakomma, A. Elgammal, A fast algorithm for manifold learning by posing it as a symmetric diagonally dominant linear system, Appl. Comput. Harmon. Anal. (2015), http://dx.doi.org/10.1016/j.acha.2015.10.004

[19] T.A. Davis, W.W. Hager, Modifying a sparse Cholesky factorization, SIAM J. Matrix Anal. Appl. 20 (3) (1999) 606–627.
[20] Timothy A. Davis, User Guide for CHOLMOD: A Sparse Cholesky Factorization and Modification Package, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA, 2008.
[21] Reinhard Furrer, SPAM: http://user.math.uzh.ch/furrer/research/software.shtml.
[22] Reinhard Furrer, Florian Gerber, CRAN R package spam: sparse matrix algebra, https://cran.r-project.org/web/packages/spam/spam.pdf.
[23] Marc Strickert, Nese Sreenivasulu, Bjorn Usadel, Udo Seiffert, Correlation maximizing surrogate gene space for visual mining of gene expression patterns in developing barley endosperm tissue, BMC Bioinformatics 8 (2007).
[24] Ioannis Koutis, Gary L. Miller, A linear work, $O(n^{1/6})$ time, parallel algorithm for solving planar Laplacians, in: Proceedings of 18th ACM–SIAM Symposium on Discrete Algorithms, SODA, 2007.
[25] Ioannis Koutis, Gary L. Miller, Graph partitioning into isolated, high conductance clusters: theory, computation and applications to preconditioning, in: Symposium on Parallel Algorithms and Architectures, SPAA, 2008.
[26] Ioannis Koutis, Gary Miller, The combinatorial multigrid solver, Software: http://www.cs.cmu.edu/~jkoutis/cmg.html.
[27] Ioannis Koutis, Gary L. Miller, Richard Peng, Approaching optimality for solving SDD systems, in: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, FOCS '10, 2010.
[28] Ioannis Koutis, Gary L. Miller, Richard Peng, Solving SDD linear systems in time $\tilde{O}(m \log n \log(1/\epsilon))$, CoRR, arXiv: 1102.4842 [abs], 2011.
[29] Ioannis Koutis, Combinatorial and algebraic algorithms for optimal multilevel algorithms, PhD thesis, Carnegie Mellon University, Pittsburgh, May 2007.
[30] Daniel A. Spielman, Shang-Hua Teng, Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems, SIAM J. Matrix Anal. Appl. 35 (3) (2006) 835–885.
[31] Lisha Chen, Local multidimensional scaling for nonlinear dimension reduction, graph layout and proximity analysis, Dissertations available from ProQuest, Paper AAI3225439, http://repository.upenn.edu/dissertations/AAI3225439, 2006.
[32] Michael B. Cohen, Rasmus Kyng, Gary L. Miller, Jakub W. Pachocki, Richard Peng, Anup B. Rao, Shen Chen Xu, Solving SDD linear systems in nearly $m \log^{1/2} n$ time, in: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC '14, 2014.
[33] Daniel A. Spielman, Shang-Hua Teng, Spectral sparsification of graphs, SIAM J. Comput. 40 (4) (2011) 981–1025.
[34] Wai Shing Fung, Ramesh Hariharan, Nicholas J.A. Harvey, Debmalya Panigrahi, A general framework for graph sparsification, in: Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, 2011.
[35] Daniel A. Spielman, Shang-Hua Teng, Spectral sparsification of graphs, SIAM J. Comput. 40 (4) (2011) 981–1025.
[36] Daniel A. Spielman, Nikhil Srivastava, Graph sparsification by effective resistances, SIAM J. Comput. 40 (6) (2011) 1913–1926.

Please cite this article in press as: P. Vepakomma, A. Elgammal, A fast algorithm for manifold learning by posing it as a symmetric diagonally dominant linear system, Appl. Comput. Harmon. Anal. (2015), http://dx.doi.org/10.1016/j.acha.2015.10.004