Mind the Eigen-Gap, or How to Accelerate Semi-Supervised Spectral Learning Algorithms

Dimitrios Mavroeidis
Radboud University Nijmegen
Nijmegen, Netherlands
d.mavroeidis@cs.ru.nl

Abstract

Semi-supervised learning algorithms commonly incorporate the available background knowledge such that an expression of the derived model's quality is improved. Depending on the specific context quality can take several forms and can be related to the generalization performance or to a simple clustering coherence measure. Recently, a novel perspective of semi-supervised learning has been put forward, that associates semi-supervised clustering with the efficiency of spectral methods. More precisely, it has been demonstrated that the appropriate use of partial supervision can bias the data Laplacian matrix such that the necessary eigenvector computations are provably accelerated. This result allows data mining practitioners to use background knowledge not only for improving the quality of clustering results, but also for accelerating the required computations. In this paper we initially provide a high level overview of the relevant efficiency maximizing semi-supervised methods such that their theoretical intuitions are comprehensively outlined. Consecutively, we demonstrate how these methods can be extended to handle multiple clusters and also discuss possible issues that may arise in the continuous semi-supervised solution. Finally, we illustrate the proposed extensions empirically in the context of text clustering.

1 Introduction

In machine learning, many popular clustering frameworks are related to computationally hard (often *NP*-Hard) optimization problems that need to be effectively approximated such that useful clustering models are derived for the available data. Spectral Clustering [Luxburg, 2007] constitutes a popular approximation technique that employs the eigenvectors and eigenvalues of an appropriate input matrix for computing the clustering output. The fact that spectral algorithms rely on eigenvalue-eigenvector computations may create the impression that the computational aspect is not relevant to the core data mining process and can be addressed solely by using a state of the art matrix compression technique, or a standard linear algebra eigensolver. This consideration is dominant in most spectral algorithms, with the exception of Pagerank [Brin and Page, 1998], where the computational aspect is studied in depth as a consequence of its application in the large WWW graph. In the context Pagerank it has been demonstrated that the introduction of the appropriate supervised (or even random) bias to the input probability matrix, can considerably accelerate the computation of the relevant eigenvector solution [Haveliwala and Kamvar, 2003]. This work has illustrated that efficiency enhancements do not need to be "external" to the data mining process and can be achieved by the appropriate supervised-bias of the input data matrix. Albeit the considerable innovations of the work of Haveliwala and Kamvar, its impact has been mostly restrained within the context of computing stationary random walk distributions.

In recent works, [Mavroeidis and Bingham, 2008; Mavroeidis, 2010; Mavroeidis and Bingham, 2010], the results of Haveliwala and Kamvar have been extended for Spectral Clustering and Spectral Ordering. More precisely, these works have demonstrated that the incorporation of the appropriate supervised bias to the data Laplacian matrix can enhance the efficiency of the required eigenvector computations, thus accelerating Spectral Clustering/Ordering. These findings come in support to the general intuition that semi-supervised problems should be "easier" to solve than unsupervised ones and provide data mining practitioners with a novel algorithmic framework for clustering large and sparse matrices.

In this paper we initially provide a high level overview of the relevant efficiency maximizing semi-supervised methods highlighting their theoretical intuitions. Moreover, we discuss certain issues that are of practical importance. More precisely, we extend these methods to handle multiple clusters (k>2) and also analyze some issues that may arise in the continuous semi-supervised solution. Finally, we empirically validate the efficiency and cluster quality enhancements in the context of text clustering.

2 Spectral Clustering

Spectral Clustering is a slightly ambiguous term that is used to describe the clustering algorithms that employ eigenvectors and eigenvalues for approximating a clustering objective. Spectral Clustering is commonly used for approximating the Normalized Cut ($Ncut$) objective, which is a known NP-

Complete problem [Shi and Malik, 2000]. The $NCut$ objective for two clusters is defined as:

$$NCut = \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)}$$

where A, B denote the two cluster sets, $cut(A,B) = \sum_{i \in A, j \in B} W(i,j)$, with $W(i,j)$ denoting the similarity between instances i, j and $vol(A) = \sum_{i \in A} D(i,i)$, with $D(i,i) = \sum_{j=1}^{n} W(i,j)$. A clustering algorithm essentially aims to retrieve the clustering solution $\{A, B\}$ that minimizes this objective.

The connection to eigenvector (spectral) methods can be made apparent if one writes this objective in the following equivalent form:

$$\text{argmin}_{\{A,B\}} NCut = \text{argmax}_{\{q\}} q^T L q$$

where L is the normalized Laplacian matrix $L = D^{-1/2} W D^{-1/2}$ (D denotes the degree matrix and W the instance similarity matrix) and q contains the discrete cluster assignments in the form: $q(i) = \sqrt{\frac{vol(B)}{vol(A)} D(i,i)}$ if $i \in A$ and $q(i) = -\sqrt{\frac{vol(A)}{vol(B)} D(i,i)}$ if $i \in B$.

Based on this equivalent formulation, the minimization of the $NCut$ objective resembles an eigenvalue problem for matrix L. We use the term "resembles" since q is a discrete and not continuous vector, thus the problem is still not equivalent to an eigenvector problem. Spectral methods consider here the "continuous relaxation" step that relaxes the hard discrete constraint and approximate the discrete cluster assignments using the eigenvectors of matrix L. For two-cluster $NCut$, the continuous solution is derived by the eigenvector that corresponds to the second largest eigenvalue of L, while for multi-cluster $NCut$, the continuous solution is derived by the k dominant eigenvectors of L. For discretizing the continuous results several methods have been proposed [Luxburg, 2007], with a popular choice being the simple use of a k-means algorithm on the the k-dimensional Laplacian eigenspace.

It can be observed that the computational cost of Spectral clustering is dominated by the cost of solving an eigenvector problem on the input Laplacian matrix. In the following section we will describe the relevant linear algebra algorithms that can be employed and the factors that determine their efficiency.

3 Mind the Eigen-Gap: Computing Spectral Solutions

The problem of computing the eigenvectors of a real symmetric matrix has been extensively studied in the context of linear algebra and several algorithms that correspond to different memory/processor requirements have been proposed. When solely a few eigenvectors are desired, it is common to employ iterative methods, such as the Power Method and Krylov subspace methods [Golub and Van Loan, 1996]. These methods work by iteratively performing (matrix × vector), instead of the more expensive (matrix × matrix) multiplications and converge after a certain number of steps to the desired eigenvector solution. The computational cost of these methods is determined by the cost of multiplying a matrix with a vector, and the number of steps that are required to converge to the desired solution.

The simplest iterative eigensolver is the Power Method, that uses an initial random vector b_0, and iteratively performs a matrix-vector multiplication, $b_t = \frac{Ab_{t-1}}{||Ab_{t-1}||}$ until convergence. The speed of convergence of this iterative process depends on the difference between the two largest (in absolute value) eigenvalues of matrix A. In the context of Spectral Clustering, the convergence speed of the Power Method depends on the difference between the two largest non-trivial eigenvalues of the Laplacian matrix L[1]. More precisely, if we denote the eigenvalues of the Laplacian matrix as $\lambda_0 = 1 \geq \lambda_1 \geq ... \geq \lambda_n$ then the relevant eigengap that determines the speed of convergence of the Power Method is: $\frac{\lambda_1}{\lambda_2}$. Note that we do not need to order the eigenvalues in absolute value since we can shift appropriately the Laplacian matrix such that it becomes positive semidefinite (illustrated analytically in [Mavroeidis, 2010]). Moreover, if the eigenvalues of the Laplacian matrix are not artificially inflated (i.e. through matrix multiplication), we can also use the eigengap $\lambda_1 - \lambda_2$ as done in [Mavroeidis, 2010] for illustrating the efficiency of the Power Method.

Orthogonal iterations [Golub and Van Loan, 1996] presents the natural generalization of the Power Method for computing the k largest eigenvectors of an input matrix. This method starts with a random $n \times k$ initial matrix and iteratively perform matrix multiplication and orthonormalization until convergence. The speed of convergence of this method depends on the difference between the k and the $k+1$ largest eigenvalues of a matrix. More precisely, if we denote the Laplacian eigenvalues as $\lambda_0 = 1 \geq \lambda_1 \geq ... \geq \lambda_n$, and also denote $gap = \frac{\lambda_k}{\lambda_{k-1}} \leq 1$ then the number of steps required by the orthogonal iteration for convergence are $O(\frac{1}{1-gap})$ [Bach and Jordan, 2006]. Lanczos method [Golub and Van Loan, 1996], that presents another popular choice for deriving the k dominant eigenvectors also has a dependance on the size of the relevant eigengap and requires a number of iterations that is $O(\frac{1}{\sqrt{1-gap}})$ [Bach and Jordan, 2006].

It can be observed that the speed of convergence of these methods depends on the eigengap between the k and the $k+1$ largest eigenvalues of a matrix. Thus a method that enlarges this eigengap will consequently accelerate the relevant iterative eigensolvers. In the subsequent paragraph we will recall the work of [Mavroeidis, 2010] where the size of the relevant eigengap is enlarged in the context of semi-supervised learning, thus accelerating Spectral Clustering.

[1] Due to the special structure of matrix L, it is known that it has a "trivial" largest eigenvalueλ_0 , that is equal to one, $\lambda_0 = 1$ and a corresponding largest eigenvector v_0 that is equal to $v_0 = \sqrt{\frac{D(i,i)}{\sum_i D(i,i)}}$.

4 Semi-supervised Clustering

In the relevant semi-supervised works that accelerate the Power Method computations [Mavroeidis and Bingham, 2008; Mavroeidis, 2010; Mavroeidis and Bingham, 2010], partial supervision is incorporated in the form of a rank-1 update to the graph Laplacian matrix. More precisely, in [Mavroeidis, 2010] the semi-supervised Laplacian matrix is defined as:

$$L_{semi} = L_{data} + \gamma v_1 v_1^T$$

where L_{data} is the Laplacian matrix, as computed from the input data, γ is a real valued parameter and v_1 is a vector that contains the input data labels and is defined as:

$$v_1(i) = \begin{cases} \sqrt{\frac{d_i}{vol(A^{in})}} f(i) & \text{if } i \in A^{in} \\ 0 & \text{if } i \notin A^{in} \end{cases}$$

where $A^{in} = A_1^{in} \cup A_2^{in}$ denotes the set of labeled instances (input supervision) and $f(i)$ are defined as: $f(i) = \sqrt{\frac{vol(A_2^{in})}{vol(A_1^{in})}}$, if $i \in A_1^{in}$ and $f(i) = -\sqrt{\frac{vol(A_1^{in})}{vol(A_2^{in})}}$, if $i \in A_2^{in}$.

In [Mavroeidis, 2010], it is rigorously demonstrated that this bias will impose a lower bound to the relevant eigengap of L_{semi} that controls the speed of convergence of the Power Method. More precisely it was demonstrated that $\lambda_1(L_{semi}) - \lambda_2(L_{semi}) \geq \gamma - 2$ (and $\lambda_1(L_{semi}) - \lambda_2(L_{semi}) \geq \gamma - 1$ when L_{data} is positive semidefinite). It can be easily derived (by simply changing the last step in the proof of Theorem 2 in [Mavroeidis, 2010]) that the γ parameter also bounds the multiplicative eigengaps: $\frac{\lambda_1(L_{semi})}{\lambda_2(L_{semi})} \geq \gamma - 1$ for general L_{data} matrices and $\frac{\lambda_1(L_{semi})}{\lambda_2(L_{semi})} \geq \gamma$ when L_{data} is positive semidefinite. Thus, setting γ to a moderately large value is guaranteed to speed up the convergence rate of the Power Method.

As analyzed in [Mavroeidis, 2010], an intuitive way to understand the behavior of the supervised rank-1 updates, is to interpret them as a supervised similarity learning mechanism that increases the weights of instances that belong to the same cluster and decreases the similarities of objects that belong to different clusters. This perspective can be illustrated if we write the semi-supervised Laplacian as:

$$L_{semi}(i,j) =$$
$$\begin{cases} L_{data}(i,j) - \gamma \frac{\sqrt{d_i d_j}}{vol(A^{in})} & \text{if } i,j \text{ in diff. clusters} \\ L_{data}(i,j) + \gamma \frac{\sqrt{d_i d_j}}{vol(A^{in})} \frac{vol(A_2^{in})}{vol(A_1^{in})} & \text{if } i,j \in A_1^{in} \\ L_{data}(i,j) + \gamma \frac{\sqrt{d_i d_j}}{vol(A^{in})} \frac{vol(A_1^{in})}{vol(A_2^{in})} & \text{if } i,j \in A_2^{in} \\ L_{data}(i,j) & \text{otherwise} \end{cases}$$

The afore explicit writing of matrix L_{semi} illustrates the effects of partial supervision to the elements of matrix L_{data} (recall that $L_{data} = D^{-1/2} W D^{-1/2}$ is essentially a normalized similarity matrix). The similarity of instances that belong to different clusters will be decreased by $-\gamma \frac{\sqrt{d_i d_j}}{vol(A^{in})}$, while the similarity for the objects that belong to the same cluster will be increased by $\gamma \frac{\sqrt{d_i d_j}}{vol(A^{in})} \frac{vol(A_2^{in})}{vol(A_1^{in})}$ and $\gamma \frac{\sqrt{d_i d_j}}{vol(A^{in})} \frac{vol(A_1^{in})}{vol(A_2^{in})}$ respectively. The similarity between objects for which we do not have any label information, will remain intact.

Although the same general intuition is followed by several semi-supervised similarity learning methods that adjust the data similarities such that input supervision is taken into account (see related work section of [Mavroeidis, 2010] for appropriate references), the specific weighting scheme of L_{semi} also entails the acceleration property that substantially differentiates it from the relevant work.

Although the supervised rank-1 Laplacian updates presented in this section introduce several novelties for handling partial supervision, the discussion is confined for two-way clustering and only for partial supervision provided in the form of cluster labels. In the following section we will illustrate how this framework can be extended for multiple clusters, i.e. $k > 2$.

5 Multiple Clusters

In the context of semi-supervised multi-way clustering we consider as input a set of cluster labels for each cluster, i.e. $A_i^{in} \subseteq A_i$ for $i = 1, 2, ..., k$. Based on this information we can formulate the semi-supervised Laplacian matrix as a rank-k update of the original data-Laplacian matrix as:

$$L_{semi} = L_{data} + \gamma \sum_{i=1}^{k} v_i v_i^T$$

where

$$v_j(i) = \begin{cases} \sqrt{\frac{d_i}{vol(A_j^{in})}} & \text{if } i \in A_j^{in} \\ 0 & \text{if } i \notin A_j^{in} \end{cases}$$

The differences between the definition of the v_j vectors for $k > 2$ and $k = 2$ can be understood if one looks into the differences between the $NCut$ quadratic formulations for $k = 2$ and $k > 2$ in [Luxburg, 2007]. There, it can be observed that the v_1 definition for 2-way clustering essentially resembles a cluster indicator vector for $k = 2$, while the vectors v_j used for $k > 2$, resemble the cluster indicator vectors for $k > 2$. The differences are also justified by the fact that the continuous cluster solution for $k = 2$ is derived by the second largest eigenvector, while for $k > 2$ the continuous solution is derived by the k largest eigenvectors.

Based on this formulation of L_{semi}, Theorem 2 in [Mavroeidis, 2010] can be extended such that eigengap bounds are derived for the difference between the k and $k+1$ largest eigenvalues of matrix L_{semi}. More precisely, it can be shown that $\lambda_{k-1}(L_{semi}) - \lambda_k(L_{semi}) \geq \gamma - 2$ for general matrices and $\lambda_{k-1}(L_{semi}) - \lambda_k(L_{semi}) \geq \gamma - 1$ when L_{data} is positive semidefinite. One can also derive the multiplicative eigengaps: $\frac{\lambda_{k-1}(L_{semi})}{\lambda_k(L_{semi})} \geq \gamma - 1$ for general L_{data} matrices and $\frac{\lambda_{k-1}(L_{semi})}{\lambda_k(L_{semi})} \geq \gamma$ when L_{data} is positive semidefinite. These bounds can be derived by the careful application of Weyl's theorem in a similar fashion as in [Mavroeidis, 2010].

In order to illustrate the effect of the supervised rank-k update on the elements of the data Laplacian matrix L_{data} we write:

$$L_{semi} = \begin{cases} L_{data}(i,j) + \gamma \frac{\sqrt{d_i d_j}}{vol(A_l^{in})} & \text{if } i,j \in A_l^{in} \\ L_{data}(i,j) & \text{otherwise} \end{cases}$$

The afore explicit writing of matrix L_{semi} illustrates that the similarity of the objects that belong to the same clusters will be increased by $\gamma \frac{\sqrt{d_i d_j}}{vol(A_l^{in})}$, while the similarity for the other objects, for which we do not have background knowledge will be based solely on the information encoded in L_{data}.

Having described the multi-cluster extension, we can move on to the experiments section where the proposed framework is verified empirically in the context of text clustering.

6 Experiments

In order to validate our approach for $k > 2$ clusters, we have used four multi-cluster subsets of the 20-newsgroup Dataset:
Newsgroups Dataset 1:{*comp.graphics/comp.os.ms-windows.misc/comp.sys.ibm.pc.hardware*}
Newsgroups Dataset 2:{*rec.motorcycles/rec.sport.baseball/ rec.sport.hockey*}
Newsgroups Dataset 3:{*talk.politics.guns/talk.politics.mideast/ talk.politics.misc*}
Newsgroups Dataset 4:{*sci.crypt/sci.electronics/sci.med/ sci.space/soc.religion.christian*}

For each datasets we have employed the tf-idf weighting scheme, using the idf values of the whole 20-newsgroup corpus. The similarity matrix W was consequently created using the inner product similarity between documents. For discretizing the continuous clustering results V_k[2], we have normalized the rows of V_k, and consecutively applied k-means clustering.

We have employed two configurations for k-means: k-means-*RANDOM* and k-means-*FIXED*. k-means-*RANDOM* works with random initial cluster centers, while k-means-*FIXED* considers as initial cluster centers (random) elements that are contained in the input label supervision.

We have experimented with different sizes of supervision, ranging from 1% to 50% of the input data (i.e. at each run x% of each cluster is used for forming the rank-k update). For each level of supervision, we report the (multiplicative) relevant eigengap, as well as the Normalized Mutual Information (NMI) of the derived clustering. The reported results are averaged over 10 runs using random samples for the specified supervision size.

In the experiments we have set the γ parameter to a fixed value of $\gamma = 1.25$. This setting asserts that the relevant eigengap will be larger than 0.25 (and the multiplicative relevant eigengap $\frac{\lambda_{k-1}}{\lambda_k}$ will be larger than 1.25), thus guaranteeing a quick convergence to the required spectral solution. Moreover, we study the behavior of the proposed semi-supervised framework using "random" supervision. In these cases the rank-k updates are computed using random instances not taking into account the instance labels.

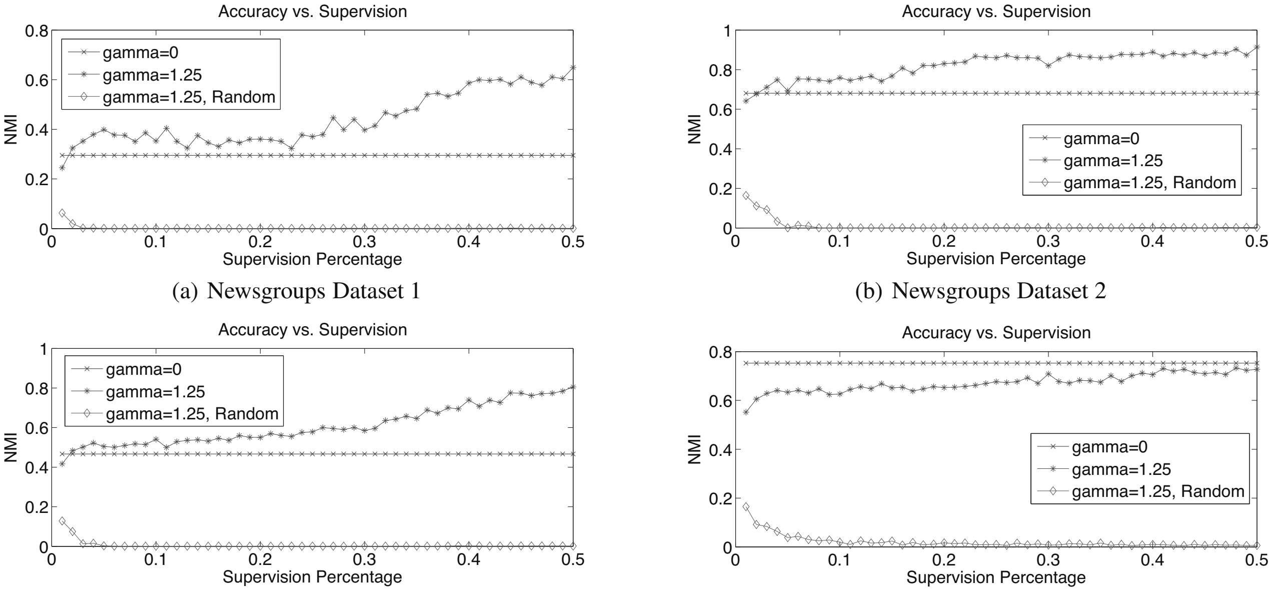
In Figure 1, we present the quality of unsupervised clustering (gamma=0) vs. the semi-supervised results (gamma=1.25) vs. the quality of the semi-supervised framework with random supervision (gamma=1.25, Random). In this Figure we employ the k-means-*RANDOM* approach for discretizing the results (random initial cluster centers). One initial observation is that the semi-supervised framework with random supervision (gamma=1.25, Random) degrades the quality of clustering results (essentially it produces random clusters since NMI quickly becomes zero). This result illustrates that random supervision, although it improves on the efficiency of the relevant eigenvector computations (see Figure 3), is not an appropriate mechanism for accelerating Spectral Clustering.

With regards to the semi-supervised cluster quality, the general observation is that in two Figures, Figures 1(b) and 1(c), the proposed framework enhances the quality of the clustering results even for very small amounts of input supervision (above 1%), while in the other two cases, Figures 1(a) and 1(d) it does not behave well. As demonstrated in Figure 2, the problematic cases can be corrected if one uses the k-means-*FIXED* approach for discretizing the results. We report here both discretization approaches in order to emphasize the importance of careful discretization for the semi-supervised continuous results. With the appropriate discretization procedure, the quality of clustering results is improved in most experiments for small amounts of input supervision.

The careful discretization is needed because of two issues that may arise. Firstly, the formulation of a small amount of supervision as a rank-k update may cause large similarity adjustments to a small number of instance-pairs, thus resulting in a L_{semi} matrix with imbalanced entries. Large matrix value imbalances will also translate to large value imbalances in the eigenvectors thus creating a difficult to cluster k-eigenspace. Secondly, a rank-k bias that contains a small number of labeled examples may promote "bad" eigenvectors of the L_{data} matrix that coincidentally agree with the input supervision. This is because the vectors that are used for defining the supervised rank-k update can be written as a linear combination of the Laplacian eigenvectors. Noise will be inserted when "bad" eigenvectors coincidentally have a high correlation with the input supervision vectors. In these cases some "bad" eigenvectors will be able to influence the spectral solution of the L_{semi} matrix. Thus, for small amounts of input supervision, it is expected that a certain level of noise will also be inserted. It should be noted that albeit these issues the proposed semi-supervised framework is demonstrated to enhance the performance of clustering in most cases even for small amounts of input supervision.

The size of the multiplicative relevant eigengap (ratio between the k and the $k+1$ largest eigenvalues of the Laplacian matrix) is reported in Figure 3. In all Figures it can be observed that this eigengap is drastically enlarged and is also above the theory bound of 1.25 ($\gamma = 1.25$). These results illustrate that the relevant eigenvector computations can be accelerated, thus enhancing the efficiency of Spectral Clus-

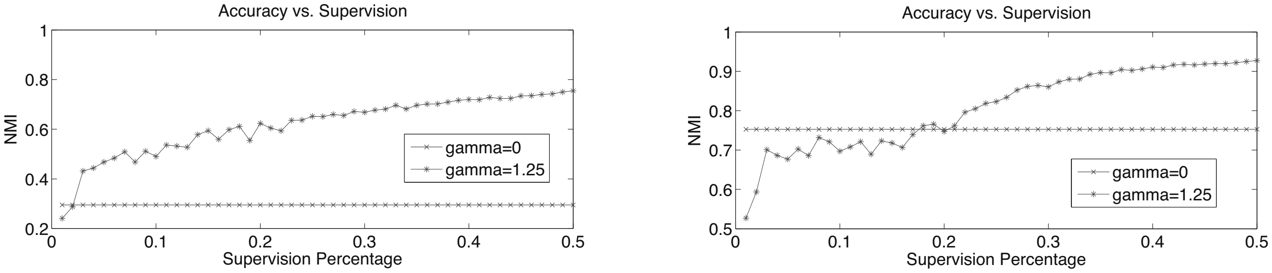
[2] V_k is an (instance × k) matrix that contains as columns the k dominant eigenvectors of the Laplacian matrix



(a) Newsgroups Dataset 1 (b) Newsgroups Dataset 2

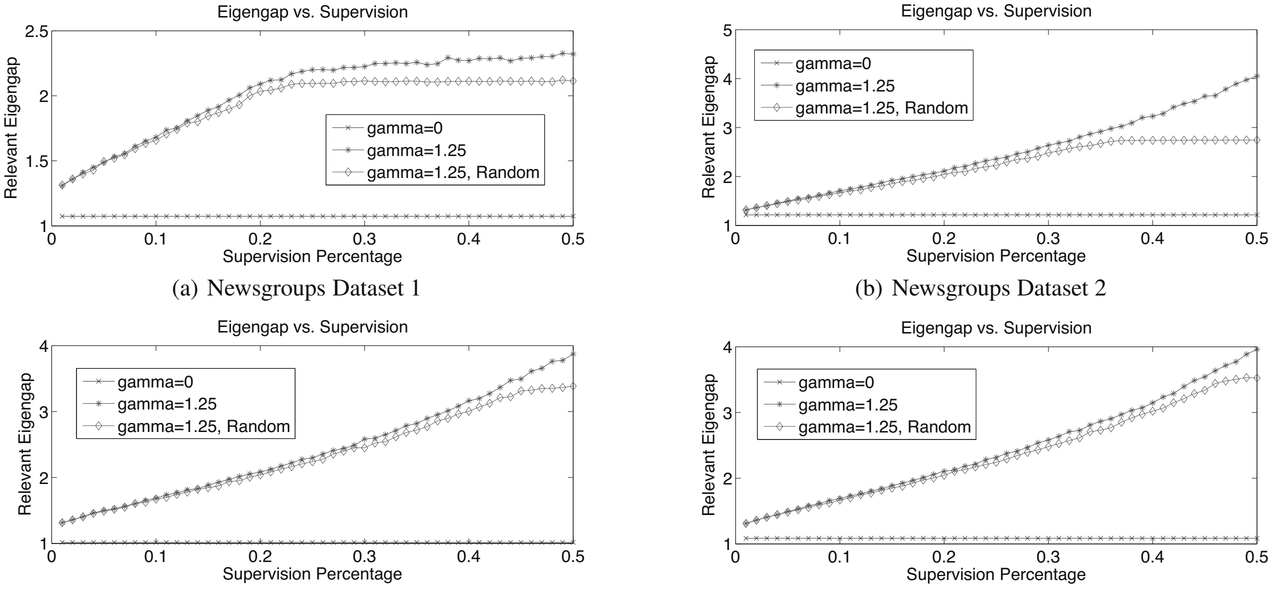
(c) Newsgroups Dataset 3 (d) Newsgroups Dataset 4

Figure 1: NMI Results with k-means-*RANDOM*



(a) Newsgroups Dataset 1 (b) Newsgroups Dataset 4

Figure 2: NMI Results with k-means-*FIXED*



(a) Newsgroups Dataset 1 (b) Newsgroups Dataset 2

(c) Newsgroups Dataset 3 (d) Newsgroups Dataset 4

Figure 3: Relevant Eigengap vs. Supervision

tering.

7 Discussion and Further Work

In further work, we aim to study in depth the input supervision size effect. This is an important issue, since the labeled data size determines the bias magnitude that is imposed to each element of the L_{data} matrix. I.e. using a very small number of instances as input supervision may create large imbalances in the elements of the L_{semi} matrix, thus creating analogous imbalances in the continuous spectral solution. It should be noted though, that in the empirical results reported in this paper, the value imbalancing issue did not seem to affect the ability of the proposed framework to increase the quality of the clustering results.

In some cases, imbalancing can be avoided by setting the appropriate size for input supervision. For example in large and sparse graphs with approximately fixed degree (i.e. when using a k-nearest neighbor graph), this effect can be completely avoided if the number of labeled data is roughly equal to the average number of instances that determine the graph degrees. Moreover, if the values of L_{data} matrix are already imbalanced, partial supervision can also be used for balancing the matrix values (instead of being a source of imbalance). This discussion illustrates that the effect of input supervision size should be further studied and possibly associated to properties of the input data.

Apart from the issue of input supervision size, we aim to study the connection between semi-supervised algorithms and efficiency beyond spectral algorithms.

Acknowledgements

This work was partially supported by the Netherlands Organization for Scientific Research (NWO) within NWO project 612.066.927

References

[Bach and Jordan, 2006] Francis R. Bach and Michael I. Jordan. Learning spectral clustering, with application to speech separation. *J. Mach. Learn. Res.*, 7:1963–2001, December 2006.

[Brin and Page, 1998] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[Golub and Van Loan, 1996] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[Haveliwala and Kamvar, 2003] Taher Haveliwala and Sepandar Kamvar. The second eigenvalue of the google matrix. *Stanford University Technical Report*, (2003-20), 2003.

[Luxburg, 2007] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, December 2007.

[Mavroeidis and Bingham, 2008] Dimitrios Mavroeidis and Ella Bingham. Enhancing the stability of spectral ordering with sparsification and partial supervision: Application to paleontological data. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 462–471, Washington, DC, USA, 2008. IEEE Computer Society.

[Mavroeidis and Bingham, 2010] Dimitrios Mavroeidis and Ella Bingham. Enhancing the stability and efficiency of spectral ordering with partial supervision and feature selection. *Knowl. Inf. Syst.*, 23:243–265, May 2010.

[Mavroeidis, 2010] Dimitrios Mavroeidis. Accelerating spectral clustering with partial supervision. *Data Min. Knowl. Discov.*, 21:241–258, September 2010.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:888–905, August 2000.