A Random Walks View of Spectral Segmentation

Marina Meilă
University of Washington
mmp@stat.washington.edu

Jianbo Shi
Carnegie Mellon University
jshi@cs.cmu.edu

Abstract

We present a new view of clustering and segmentation by pairwise similarities. We interpret the similarities as edge flows in a Markov random walk and study the eigenvalues and eigenvectors of the walk's transition matrix. This view shows that spectral methods for clustering and segmentation have a probabilistic foundation. We prove that the Normalized Cut method arises naturally from our framework and we provide a complete characterization of the cases when the Normalized Cut algorithm is exact. Then we discuss other spectral segmentation and clustering methods showing that several of them are essentially the same as NCut.

1 Introduction

This paper focuses on *pairwise* (or *similarity-based*) clustering and image segmentation. In contrast to statistical clustering methods, that assume a probabilistic model that generates the observed data points (or pixels), pairwise clustering defines a *similarity function* between pairs of points and then formulates a criterion (e.g. maximum total intracluster similarity) that the clustering must optimize. The optimality criteria quantify the intuitive notion that points in a cluster (or pixels in a segment) are similar, whereas points in different clusters are dissimilar. The similarities are considered as given in the context of the clustering algorithm; in practice (document clustering, image segmentation) finding a "good" similarity function is part of the art of the domain practitioner.

An increasingly popular approach to similarity based clustering and segmentation is by spectral methods. These methods use eigenvalues and eigenvectors of a matrix constructed from the pairwise similarity function (e.g. LSA [2]). Spectral methods are sometimes regarded as approximations of previously formulated criteria (e.g. [10, 4]) and sometimes are motivated by graph theoretical considerations (e.g. the web clustering method of [6]). As demonstrated in [10, 6], these methods are capable of delivering impressive image segmentation results using simple low-level image features. Moreover, computational efficiency is achieved using sparse [10, 3] matrix techniques.

The main achievement of this work is to show that there is a simple probabilistic interpretation that can offer insights and serve as an analysis tool for all the spectral methods cited above. We view the pairwise similarities as edge flows in a Markov random walk and study the properties of the eigenvectors and values of the resulting transition matrix. Using this view, we were able to show that several of the above methods are subsumed by the Normalized Cut (NCut) image segmentation algorithm of [10] in a sense that will be described. Therefore, in the following, we will focus on the NCut algorithm and will adopt the terminology of image segmentation (i.e. the data points will be *pixels* and the set of all pixels is the *image*), keeping in mind that all the results are also valid for similarity based clustering.

2 The Normalized Cut criterion and algorithm

Here and in the following, an image will be represented by a set of pixels I. A segmentation is a partioning of I into mutually disjoint subsets. For each pair of pixels $i, j \in I$ a similarity $S_{ij} = S_{ji} \geq 0$ is given. In the NCut framework the similarities S_{ij} are viewed as weights on the edges ij of a graph G over I. If $S_{ij} = 0$ then G has no edge ij. The matrix $S = [S_{ij}]$ plays the role of a "real-valued" adjacency matrix for G. Let $d_i = \sum_{j \in I} S_{ij}$, called the *degree* of node i, and the *volume* of a set $A \subset I$ be $\text{vol}\, A = \sum_{i \in A} d_i$.. The set of edges between A and its complement $\bar{A}$ is an *edge cut* or shortly a *cut*. The *normalized cut* (NCut) criterion of [10] is a graph theoretical criterion for segmenting an image into two by minimizing

$$NCut(A, \bar{A}) \;=\; \left(\frac{1}{\mathrm{vol}\, A} + \frac{1}{\mathrm{vol}\, \bar{A}}\right) \sum_{i \in A, j \in \bar{A}} S_{ij} \quad (1)$$

over all cuts $A, \bar{A}$. Minimizing $NCut$ means finding a cut of relatively small weight between two subsets with strong internal connections. In [10] it is shown that optimizing the $NCut$ criterion is NP hard.

The *NCut algorithm* was introduced in [10] as an approximate method of solving the minimum NCut problem by way of eigenvalues and eigenvectors. It uses the *Laplacian* matrix $L \;=\; D - S$ where D is a diagonal matrix formed with the degrees of the nodes. The algorithm consists of solving the generalized eigenvalues/vectors problem

$$Lx \;=\; \lambda D x \quad (2)$$

The NCut algorithm focuses on the second smallest eigenvalue of (2) and its corresponding eigenvector, call them λ^L and x^L respectively.

Figure 1 shows an example of a similarity matrix that has a pronounced block structure (Ib), and its first 3 generalized eigenvectors (IIIa). In the figure we see that the elements of x^L have approximately the same value within each cluster. In [10] it is shown that when there is a partitioning of $A, \bar{A}$ of I such that

$$x_i^L \;=\; \begin{cases} \alpha, & i \in A \\ \beta, & i \in \bar{A} \end{cases} \quad (3)$$

then $A, \bar{A}$ is the optimal NCut and the value of the cut itself is $NCut(A, \bar{A}) \;=\; \lambda^L$.

This result represents the basis of spectral segmentation by normalized cuts. One solves the generalized spectral problem (2), then finds a partitioning of the elements of x^L into two sets containing roughly equal values. The partitioning can be done by thresholding the elements. The partitioning of the eigenvector induces a partition on I which is the desired segmentation. To obtain more than two segments one proceeds recursively. We call this procedure the NCut algorithm. A vector that satisfies (3) is called *piecewise constant* w.r.t. the partition $(A, \bar{A})$. In section 4 and later we consider eigenvectors which are piecewise constant w.r.t a partition of I into k sets.

As presented above, the NCut algorithm lacks a satisfactory intuitive explanation. In particular, the NCut algorithm and criterion offer little intuition about (1) what causes x^L to be piecewise constant? (2) what happens when there are more than two segments and (3) how does the algorithm degrade its performance when x^L is not piecewise constant?

The random walk interpretation that we describe now will answer the first two questions as well as give a better understanding of what spectral clustering is achieving. We shall not approach the third issue here: instead, we point to the results of [4] that apply to the NCut algorithm as well.

3 Markov walks and normalized cuts

By "normalizing" the similarity matrix S one obtains the stochastic matrix

$$P \;=\; D^{-1} S \quad (4)$$

whose row sums are all 1. As it is known from the theory of Markov random walks, P_{ij} represents the probability of moving from node i to j in one step, given that we are in i. The eigenvalues of P are $\lambda_1 = 1 \geq \lambda_2 \geq \ldots \lambda_n \geq -1$; $x^{1 \ldots n}$ are the eigenvectors. The first eigenvector of P is $x^1 = \mathbf{1}$, the vector whose elements are all equal to 1. W.l.o.g we assume that no node has degree 0.

Let us now examine the spectral problem for the matrix P, namely the solutions of the equation

$$Px \;=\; \lambda x \quad (5)$$

Proposition 1 *If λ, x are solutions of (5) and $P = D^{-1}S$, then $(1 - \lambda)$, x are solutions of (2).*

In other words, the NCut algorithm and the matrix P have the same eigenvectors; the eigenvalues of P are identical to the difference between 1 and the generalized eigenvalues in (2). Proposition 1 shows the equivalence between the spectral problem formulated by the NCut algorithm and the eigenvalues/vectors of the stochastic matrix P. This also helps explaining why the NCut algorithm uses the second smallest generalized eigenvector: the smallest eigenvector of (2) corresponds to the largest eigenvector of P, which in most cases of interest is equal to $\mathbf{1}$ thus containing no information. The proof of proposition 1 is elementary and therefore left as an exercise to the reader.

The NCut criterion can also be understood in this framework. First define $\pi^\infty = [\pi_i^\infty]_{i \in I}$ by

$$\pi_i^\infty \;=\; \frac{d_i}{\mathrm{vol} I}. \quad (6)$$

It is easy to verify that $P^T \pi^\infty \;=\; \pi^\infty$ and thus that π^∞ is a *stationary distribution* of the Markov chain. If the chain is ergodic, which happens under mild conditions [1], then π^∞ is the only distribution over I with this property. Note also that the Markov chain is *reversible* because

$$\pi_i^\infty P_{ij} \;=\; \pi_j^\infty P_{ji} \;=\; S_{ij}/\mathrm{vol} I. \quad (7)$$

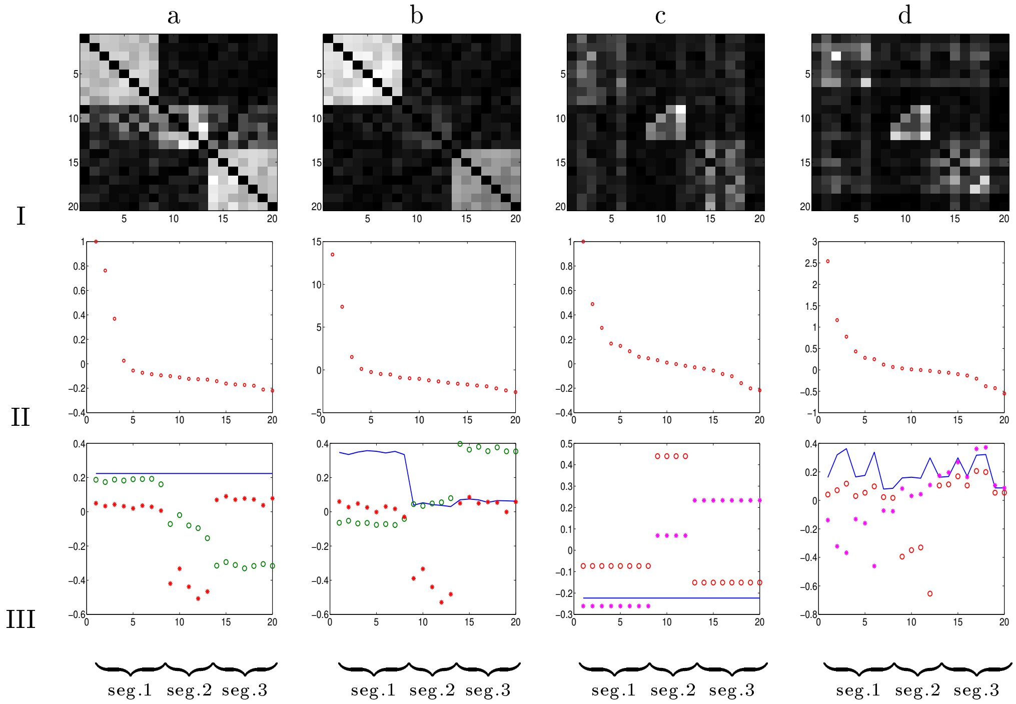

Figure 1: Four matrices (row I), their eigenvalues (row II) and first 3 eigenvectors: x^1 '—', $x^2 (= x^L$ in b,d) '$\circ$', x^3 '$\star$' (row III). All matrices are represented on a gray-scale with black for 0 and lighter shades for higher values. All matrices correspond to "images" of 20 pixels forming 3 segments. (a) An approximately block-diagonal stochastic matrix P_1. The second and third eigenvector are approximately piecewise constant and contain information about the segmentation. (b) The symmetric similarity matrix which produced P_1. Note that all three first eigenvectors contain information about the segmentation. The eigenvectors solving (2) for this matrix are identical to the eigenvectors of P_1. (c) A block-stochastic matrix P_2. The second and third eigenvectors are piecewise constant and reflect the correct segmentation. (d) The symmetric similarity matrix that produced P_2. The first 3 eigenvectors are only roughly piecewise constant and result in a wrong segmentation.

Define $P_{AB} = Pr[A \to B | A]$ as the probability of the random walk transitioning from set $A \subset I$ to set $B \subset I$ in one step if the current state is in A and the random walk is started in its stationary distribution.

$$P_{AB} = \frac{\sum_{i \in A, j \in B} \pi_i^\infty P_{ij}}{\pi^\infty(A)} = \frac{\sum_{i \in A, j \in B} S_{ij}}{\text{vol}(A)} \qquad (8)$$

From this it follows that

$$NCut(A, \bar{A}) = P_{A\bar{A}} + P_{\bar{A}A} \qquad (9)$$

If the NCut is small for a certain partition $A, \bar{A}$ then it means that the probabilities of evading set A, once the walk is in it and of evading its complement $\bar{A}$ are both small. Intuitively, we have partioned the set I into two parts such that the random walk, once in one of the parts, tends to remain in it.

The NCut is strongly related to a the concept of low conductivity sets in a Markov random walk. A *low conductivity set* A is a subset of I such that $h(A) = \max(P_{A\bar{A}}, P_{\bar{A}A})$ is small. They have been studied in spectral graph theory in connection with the *mixing time* of Markov random walks [1]. More recently, [4] uses them to define a new criterion for clustering. Not coincidentally, the heuristic analyzed there is strongly similar to the NCut algorithm.

4 Stochastic matrices with piecewise constant eigenvectors

In the following we will use the transition matrix P to achieve a better understanding of the NCut algorithm. Recall that the NCut algorithm looks at the second "largest" eigenvector of P, denoted by x^2 and equal to x^L, in order to obtain a partioning of I. We define a vector x to be *piecewise constant* relative to a partition $\Delta = (A_1, A_2, \ldots A_k)$ of I iff $x_i = x_j$ for i, j pixels in the same set A_s, $s = 1, \ldots k$. Note that the first eigenvector of P, being $\mathbf{1}$, is always piecewise constant. Since having piecewise constant eigenvectors is essential for spectral segmentation, it is important to understand when the matrix P has this desired property. We study when the first k out of n eigenvectors are piecewise constant.

Proposition 2 *Let P be a matrix with rows and* *columns indexed by I that has independent eigenvectors. Let $\Delta = (A_1, A_2, \ldots A_k)$ be a partition of I. Then, P has k eigenvectors that are piecewise constant w.r.t. Δ and correspond to non-zero eigenvalues if and only if the sums $P_{is} = \sum_{j \in A_s} P_{ij}$ are constant for all $i \in A_{s'}$ and all $s, s' = 1, \ldots k$ and the matrix $R = [P_{ss'}]_{s,s'=1,\ldots k}$ (with $P_{ss'} = \sum_{j \in A'_s} P_{ij}$, $i \in A_s$) is non-singular.*

Lemma 3 *If the matrix P of dimension n is of the form $P = D^{-1}S$ with S symmetric and D non-singular then P has n independent eigenvectors.*

The proof of the lemma is elementary and therefore omitted; proposition 2 is proved in the appendix. We call a stochastic matrix P satisfying the conditions of Proposition 2 a block-stochastic matrix. Intuitively, Proposition 2 says that a stochastic matrix has piecewise constant eigenvectors if the underlying Markov chain can be aggregated into a Markov chain with state space $\Delta = \{A_1, \ldots A_k\}$ and transition probability matrix $\hat{P}$. This opens interesting connections between the field of spectral segmentation and the body of work on aggregability or (*lumpability*) [5] of Markov chains.

It has been already shown [12, 4, 10] that for a disconnected graph G (resulting in a block diagonal S) the NCut algorithm and several others work correctly. A block diagonal S is a block-stochastic matrix for which $\hat{P}$ is the unit matrix. It represents the case when pixels in different segments are strongly dissimilar. This case, illustrated in figure 1 (a,b), is by far the easiest situation for a segmentation problem.

Now Proposition 2 shows that in fact *spectral clustering is able to group pixels by the similarity of their transition probabilities to subsets of I*. This situation is shown in figure 1,c,d. Experiments [10] show that NCut works well on many graphs that are not disconnected supporting this result with practical evidence.

However, having piecewise constant eigenvectors is only part of the story. It is also necessary that the eigenvalues of $\hat{P}$, corresponding to the piecewise constant eigenvectors be larger than the other $n-k$ eigenvalues of P, that we shall call *spurious* eigenvalues.

With the above insights, we can define an abstract algorithm called *Modified NCut* (MNCut) which finds all k segments in one pass by: (1) computing P from S, its eigenvalues/vectors (2) selecting the largest k eigenvalues and their corresponding eigenvectors (3) extracting the segments by finding the approximately equal elements in the selected eigenvectors. This last step can be done e.g. by projecting onto or by k-means (with k known) in the $k-1$ dimensional space defined by the rows of $[x^2 \ldots x^k]$.

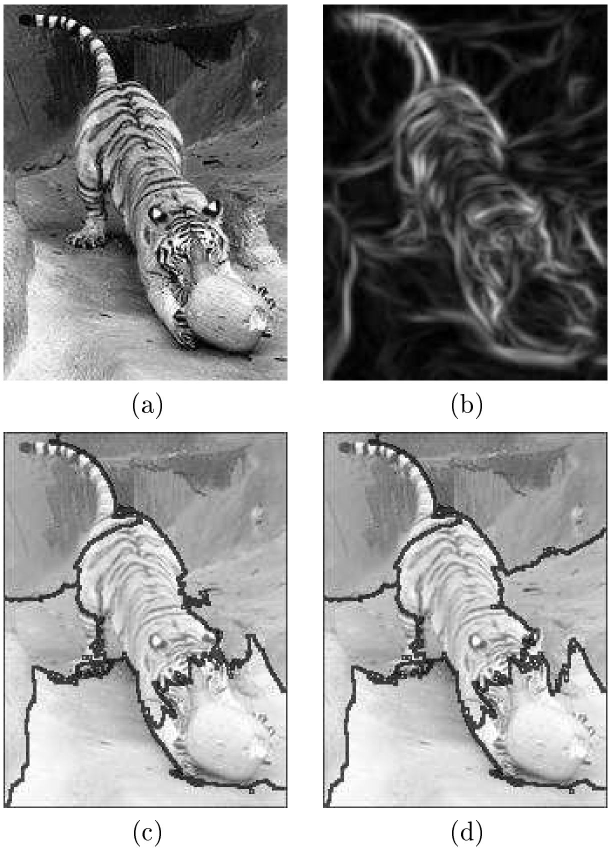

Figure 2: Image segmentation by the MNCut algorithm: (a) the original image; (b) the output of the edge detector; (c,d) segmentation by MNCut using the first 6 respectively 7 eigenvectors and k-means clustering. Two pixels are dissimilar if they are more than 30 apart or if they are separated by an edge; otherwise they are considered similar. Note that even with this simple similarity measure and in spite of the many stripes, most of the tiger is segmented correctly.

Proposition 4 *The MNCut algorithm is exact if P is block-stochastic and the eigenvalues of $\hat{P}$ are larger than the spurious eigenvalues.*

Thus MNCut exploits both dissimilarities between pixels in different segments and similarity of transitions for pixels in the same segment.

The MNCut approach has another potential advantage: if there is a gap between the eigenvalues of $\hat{P}$ and the spurious eigenvalues (as in figure 1, c, d), then the number of segments k can be determined automatically. This is likely to happen when (i) $\hat{P}$ approaches the unit matrix, its eigenvalues tending to 1, and (ii) the rows of P in the same segment tend to be equal, pushing the spurious eigenvalues toward 0. Thus, once again, a mix of dissimilarity between clusters and similarity of transitions describes a data set that is naturally clustered.

5 Relationship to other spectral segmentation methods

The NCut algorithm and criterion is only one of the recently proposed segmentation methods that use eigenvectors. Here we discuss a few others: the segmentation algorithms of Perona and Freeman (PF) [8] and of Scott and Longuet-Higgins (SLH) [9]. In addition, we discuss two clustering methods that have the same flavor: the Kleinberg algorithm for discovering web communities (K) [6] and the long known latent semantic analysis (LSA) in the variant proposed by Kannan, Vempala and Vetta (KVV) [4].

For the algorithms of PF, SLH, and K we established the following: Each of them has an "ideal" case for which it will work exactly. For PF, the ideal case is the case when S is block diagonal. For SLH, when the $n \times n$ matrix $Q = [y^1 y^2 .. y^k][y^1 y^2 .. y^k]^T$, with $y^1 y^2 .. y^k$ the eigenvectors of S, has element $Q_{ij} = 1$ if pixels i, j are in the same segment and 0 otherwise. The K algorithm allows one to pursue a variety of objectives. One of them is finding clusters of related documents. For this objective, the ideal case corresponds to a directed link graph consisting of several disconnected d-regular clusters. Then the second eigenvector used by K will be piecewise constant w.r.t to this partition. In practice, however, the K algorithm finds the elements of the eigenvector that are largest in magnitude and returns them as representative or "authoritative" for the cluster. We conjecture that these elements correspond to the the pages with highest degree (most links) within the cluster. Proving this conjecture is a topic of current research.

It is easy to show that each of the above ideal situations imply that the resulting stochastic matrix P satisfies the conditions of Proposition 4 and thus the MNCut algorithm will also work exactly in these situations. In this sense NCut subsumes PF, SLH and (certain variants of) K. Moreover, none of the three other methods takes into account more information than NCut does.

Another important aspect of a spectral clustering algorithm is robustness. Empirical results of [12] show that NCut is at least as robust as PF and SLH in practical situations.

The algorithm of KVV is essentially a special case of MNCut where: S_{ij} is defined as $f_i^T f_j$ with f_i, f_j vectors of positive features; the method in step (3) is projection onto the scaled eigenvectors $\lambda_s x^s$. [4] proves error bounds that depend on the deviation of S from block-diagonality for both KVV and the recursive NCut algorithm. These are the only robustness results for the NCut algorithm that we know of.

6 Conclusions

The relationship between the Laplacian of a graph and Markov chains has been known [1] but so far it has been used mainly to estimate mixing properties of chains by way of cuts. This paper opens a new perspective: revealing the properties of the underlying weighted graph by ways of the Markov chain. This shift in perspective is made even more valuable because of the successes of sampling techniques [10, 3] in tractably obtaining low rank approximations to very large matrices. As the case of LSA proves it, these algorithms are used in practice on large scale problems.

Our view has provided an elegant analysis method. It has enabled us to give a complete and intuitive characterization of the NCut algorithm. We analyzed several other algorithms with the same tool to realize that they look at the same kind of features (mainly dissimilarity between pixels in different clusters) so that both technically and from the end result point of view, they are in fact all variants of the same algorithm.

We argue for studying the MNCut algorithm as a *clustering criterion in its own right*. MNCut is one of the rare cases when a clustering method is both understandable, computationally tractable (or approximable with known bounds) and yielding itself to analysis. We may then study other clustering criteria (see [3]) as approximating MNCut and conclude that they are not so different from each other after all.

But we can also formulate clustering criteria that are genuinely different: for example, an eigenvalue of P near -1 is an indication that the graph is bipartite. We can easily imagine an algorithm for *bipartite clustering* by simply looking at the eigenvector corresponding to the most negative eigenvalue.

Another exciting issue is finding ways to balance number of clusters and clustering quality, in other words automatically finding the number of clusters. We think that the Markov chain perspective can be fruitful in this respect as well. Two very innovative approaches exist already in [4] and [11].

The implications are even further reaching: For example, in many cases S is obtained from a positive symmetric kernel. We can transfer our results about P to characterizations of the kernel classes that satisfy certain requirements or to characterizations of the data distribution that is "fit for clustering". The transition matrix view also tells us how to combat "ridge effects" in kernel derived similarity matrices.

In vision, a common issue is combining multiple criteria (e.g color, texture) into one similarity matrix. The Markov walk perspective helps us to find combination operators that preserve the underlying clustering (i.e. that preserve block stochasticity). For example, a convex combination of transition matrices preserves it, while elementwise product, a popular method for combining multiple S matrices, doesn't. We address this issues and propose a method for learning the optimal combination in [7].

A Proof of Proposition 2

"$\Rightarrow$" We assume that P has k independent and piecewise constant eigenvectors $x^1, \ldots x^k$ w.r.t. to the partition Δ that correspond to non-zero eigenvalues $\lambda_1 \ldots \lambda_k$. For x a piecewise constant vector w.r.t Δ, let $x \mapsto y$ be the one-to-one mapping that associates x with the k-dimensional vector y consisting of one element of x from each segment, i.e.

$$y(x)_s \;=\; x_i \;\text{ for } i \in A_s,\; s = 1, \ldots k \tag{10}$$

Denote by $y^l = y(x^l)$ for $l = 1, \ldots k$.

Fix $i, i' \in A_s$ for some $s = 1, \ldots k$. We have

$$(Px^l)_i \;=\; \sum_{s'=1}^{k} \left(\sum_{j \in A_{s'}} P_{ij} \right) y^l_{s'} \;=\; \lambda_l x^l_i \tag{11}$$

$$(Px^l)_{i'} \;=\; \sum_{s'=1}^{k} \left(\sum_{j \in A_{s'}} P_{i'j} \right) y^l_{s'} \;=\; \lambda_l x^l_{i'} \tag{12}$$

for each eigenvector x^l, $l = 1, \ldots k$. Denote $P_{is'} = \sum_{j \in A_{s'}} P_{ij}$, $P_{i's'} = \sum_{j \in A_{s'}} P_{i'j}$. By substracting equation (12) from (11) we get

$$\sum_{s'=1}^{k} (P_{is'} - P_{i's'}) y^l_{s'} \;=\; 0 \quad \text{for } l = 1, \ldots k \tag{13}$$

This is a linear system of k equations and k unknowns, with coefficients $y^l_{s'}$. Since the eigenvectors are independent, the above system's matrix is non-singular, implying that the system admits only the trivial solution $P_{is'} - P_{i's'} = 0$. Since i, i' and the segment s are arbitrary, it follows that for all $i \in A_s$ the sums $P_{is'}$, $s' = 1, \ldots k$ are constant in each segment A_s and can be denoted by the symbol $P_{ss'}$.

Construct now the matrix $\hat{P} = [P_{ss'}]_{s,s'=1,\ldots k}$. It is easy to verify that the eigenvectors/values of $\hat{P}$ are $y^1, \ldots y^k$ and $\lambda_1, \ldots \lambda_k$. Since the latter are all non-zero, it follows that $\hat{P}$ is non-singular.

"$\Leftarrow$" We now have to prove the converse, i.e. that if $\hat{P}$ exists and is non-singular then P has k eigenvectors that are piecewise constant w.r.t the partition Δ and their eigenvalues are non-zero. Denote by y^l, λ_l, $l = 1, \ldots k$ the eigenvectors/values of $\hat{P}$. Now we can simply verify that $x^l = x(y^l)$ for $l = 1, \ldots k$ are independent eigenvectors of P each corresponding to λ^l. ∎

References

[1] F. R. K. Chung. *Spectral Graph Theory.* American Methematical Society, 1997.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society For Information Science*, 41:391–407, 1990.

[3] P. Drineas, R. Kannan, A. Frieze, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proc. of the 10th ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[4] R. Kannan, S. Vempala, and A. Vetta. On clusterings: good, bad and spectral. In *Proc. of 41st Symposium on the Foundations of Computer Science, FOCS 2000*, 2000.

[5] J. R. Kemeny and J. L. Snell. *Finite Markov Chains.* Van Nostrand, New York, 1960.

[6] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. Technical report, IBM Research Division, Almaden Research Center, 1997.

[7] M. Meilă and J. Shi. Learning segmentation by random walks. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA, 2001. MIT Press. (to appear).

[8] P. Perona and W. Freeman. A factorization approach to grouping. In *European Conference on Computer Vision*, 1998.

[9] G. Scott and H. C. Longuet-Higgins. Feature grouping by relocalsation of eigenvectors of the proximity matrix. In *Proceeding of the British Machine Vision Conference*, 1990.

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[11] N. Tishby and N. Slonim. Data clustering by Markovian relaxation via the information bottleneck method. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA, 2001. MIT Press. (to appear).

[12] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *International Conference on Computer Vision*, 1999.