Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA [C. Alzate and J. Suykens, 2010]

Seminar Report

Julius Adorf

Technische Universität München

2012-06-25

Abstract. This article is a report on the paper *Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA* by C. Alzate and J. Suykens, 2010. The method introduced in the paper is summarized: it is a spectral clustering approach formulated as weighted kernel principal component analysis, extended to more than two clusters, and endowed with an out-of-sample extension that permits to assign labels to data points that have not been part of the initial clustering process. The differences to other state-of-the-art approaches are highlighted in order to delineate the contributions by Alzate and Suykens, and to give the reader a quick overview of the field. The experiments described by Alzate and Suykens are revisited, as well as pointers to helpful literature provided to the reader.

1 Introduction

Clustering is a field widely researched in Machine Learning. It falls into the category of unsupervised learning. As such, clustering aims at finding structure in data for which we have no labelings that can be used for guiding the learning process.

Clustering has applications in a broad range of fields. For example, within computer science, clustering can be used to segment images. In biology, clustering helps to analyze gene expression matrices that originate from microarrays. In business, specific advertisements can be provided to customers based on cluster analysis of previous buyers' records.

This report concerns the problem of image segmentation in order to illustrate and discuss the clustering approach proposed in the paper "Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA" [1]. For the rest of this report, we refer to both the discussed paper and its authors as *AS10*.

First, Alzate and Suykens show in detail, how spectral clustering can be formulated in terms of weighted kernel principal component analysis. Second, they explain how to extend this spectral clustering framework to out-of-sample points. Third, they propose a new criterion for model selection.

We state the clustering problem in Section 2. The clustering method proposed by AS10 is explained in Section 3. We refer to other state-of-the-art approaches in Section 4. The experimental results and the paper as a whole are discussed in Section 5.

2 Problem

Let us define the clustering problem as follows: we are given data $X = \{\mathbf{x}^t\}_{t=1}^m$ consisting of m points, where $\mathbf{x}^t \in \mathbb{R}^n$ are the *features* of the t-th data point[1]. The objective is to assign each point to one of k clusters, such that similar points are grouped together, and dissimilar points are assigned to different clusters.

There are several questions that need to be resolved: First, how to choose the features. Second, how to determine the number of clusters. Third, how to define similarity between data points. Fourth, how to formalize our intuitive notion of the objectives in clustering. All these questions are addressed in the following sections.

We can cast the problem of image segmentation as a clustering problem. Here the pixels are the data points to be clustered. Pixels that belong to the same clusters are not necessarily connected in the image.

Image segmentation poses a challenge to any clustering algorithm, as any image contains lots of data. Even an image that seems to be "small" compared to nowadays' megapixel cameras, constitutes in fact a large problem instance. To illustrate this, see the example 481x321 image from the Berkeley Segmentation Dataset which consists of $m \approx 150.000$ pixels (Figure 4).

3 Method

3.1 Spectral Clustering through Laplacian Eigenmaps

Spectral clustering methods are based on the spectral decomposition of a particular matrix. The idea in spectral clustering is to formulate clustering as a graph partitioning problem. We define the notion of *similarity* as a function $s : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ which can be computed from the features of two data points. The similarity should be non-negative [3]. We arrange

[1] We are following the notation introduced in [2].

the data points as nodes in an undirected weighted graph, the *similarity graph* (see Figure 1). The similarity between a pair of points defines the weight on the incident edge. Now the problem of clustering turns into the problem of cutting the graph into several components; within a component, edges shall have large weights, whereas edges on the cut shall have small weights [3]. The *normalized cut* [4] criterion formalizes both objectives. Finding an optimal normalized cut is NP-complete [4]. When we relax this criterion, we can find an approximate solution by solving an eigenvalue problem.

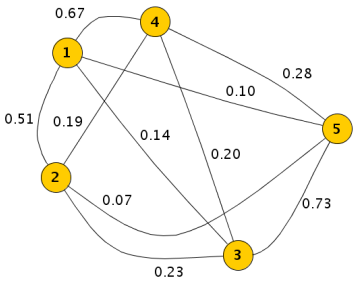

Fig. 1. A toy similarity graph.

An important role in spectral clustering is played by the *graph Laplacian* $\mathbf{L}$, for which several variants exist in the literature [3]. The simplest version is computed by $\mathbf{L} = \mathbf{D} - \mathbf{S}$, given the adjacency matrix $\mathbf{S}$ and the degree matrix $\mathbf{D}$ of the similarity graph. In the following, we assume that this graph is connected. The eigenvectors of the graph Laplacian contain structure, which is useful for clustering. For an approximate solution to the normalized cut, these are obtained by solving the following generalized eigenvalue problem [4]:

$$\mathbf{L}\boldsymbol{\alpha} = \lambda \mathbf{D}\boldsymbol{\alpha} \tag{1}$$

The diagonal matrix $\mathbf{D}$ is invertible since the similarity graph is assumed to be connected, so the generalized eigenvalue problem can be written as a standard eigenvalue problem $\mathbf{D}^{-1}\mathbf{L}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$. The eigenvector $\mathbf{1} \in \mathbb{R}^m$ for eigenvalue zero is of no interest. For the toy similarity graph in Figure 1, the graph Laplacian, the eigenvector for the second-smallest eigenvalue look like this:

$$\mathbf{L}_{\text{toy}} = \begin{pmatrix} 1.41 & -0.51 & -0.14 & -0.67 & -0.10 \\ -0.51 & 1.00 & -0.23 & -0.19 & -0.07 \\ -0.14 & -0.23 & 1.29 & -0.20 & -0.73 \\ -0.67 & -0.19 & -0.20 & 1.34 & -0.28 \\ -0.10 & -0.07 & -0.73 & -0.28 & 1.18 \end{pmatrix}, \boldsymbol{\alpha}^{(2)}_{\text{toy}} \approx \begin{pmatrix} 0.44 \\ 0.41 \\ -0.50 \\ 0.23 \\ -0.58 \end{pmatrix} \tag{2}$$

For a value of $k = 2$, the sign of the components of the eigenvector for the second-smallest eigenvalue decides the assignment to the clusters.

In general, with $k \geq 2$, we arrange the first $k-1$ eigenvectors for the smallest non-zero eigenvalues in a m-by-$(k-1)$ matrix $\mathbf{U}$. Then a *reclustering* step is performed on the new features for the points given by the rows of $\mathbf{U}$. AS10 call these new features the *score variables*. In order to cluster the points based on the score variables, we might use k-means [3], but AS10 made another choice [1]: First binarize $\mathbf{U}$. Then let the k binary vectors that appear most often in the rows be the representatives of the k clusters. Finally, the i-th data point is assigned to the cluster where the Hamming distance between the cluster representative and the i-th row of the binarized matrix $\mathbf{U}$ is minimum [1]. AS10 do not hint at what shall be done in case of a draw.

3.2 Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised method known for its applications to dimensionality reduction [2] or lossy data compression [5].

In PCA, we choose a linear subspace such that the variance in the projections is maximized. Thus we select directions, called the principal components, that explain most of the variance in the data. An equivalent formulation is that PCA finds the linear projection that minimizes the mean-squared distance between the data in the original space and the projected data on the subspace [5]. The intuition, why PCA might be useful for clustering, is fed by Figure 2 (left) where centered, linearly separable data can be clustered by thresholding the projections onto the principal component.

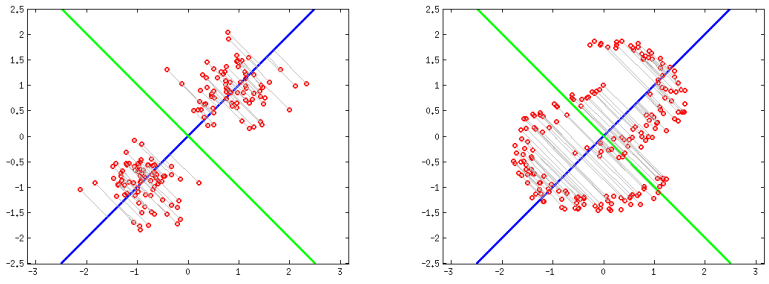

Fig. 2. Linear PCA in two dimensions. The figure shows data sampled from two bivariate Gaussian distributions (left), and data arranged in the form of two moons (right). It indicates, how the data points are orthogonally projected onto the principal component.

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be the *data matrix* formed with the features as its rows. The data covariance matrix [2] is defined as:

$$\hat{\mathbf{\Sigma}} = \frac{1}{m} \sum_{t=1}^{m} (\mathbf{x}^t - \bar{\mathbf{x}})(\mathbf{x}^t - \bar{\mathbf{x}})^T \tag{3}$$

The eigenvectors of the covariance matrix, ordered by eigenvalue in descending order, define the principal components of the data:

$$\hat{\mathbf{\Sigma}}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha} \tag{4}$$

Principal Component Analysis performs a linear projection of the data. If the data is not linearly separable as shown in Figure 2 (right), standard PCA will not help in clustering.

Kernel PCA provides a way out. The idea of kernel PCA is to map the features into high-dimensional space, and to hope that the clustering becomes possible by performing standard PCA in the high-dimensional space. The mapping can be non-linear; this makes the method more powerful but also more complicated to deal with. For a derivation of kernel PCA, see [2][5]. Especially in the context of understanding the argumentation of AS10, the keys are: First, we can link standard PCA to the dot product. Second, we avoid working directly in the high-dimensional space; instead we replace the dot product in the high-dimensional space by a *kernel* that is evaluated in the original feature space of the data.

Let $\mathbf{X}_c$ be the centered data matrix with rows $\mathbf{x}_c^t = \mathbf{x}^t - \bar{\mathbf{x}}$. Then the data covariance matrix simply becomes:

$$\hat{\mathbf{\Sigma}} = \frac{1}{m} \sum_{t=1}^{m} \mathbf{x}_c^t (\mathbf{x}_c^t)^T = \frac{1}{m} \mathbf{X}_c^T \mathbf{X}_c \tag{5}$$

For a more principled treatment of PCA and centering, see [6]. We are only interested in the eigenvectors of $\mathbf{X}_c^T \mathbf{X}_c \boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$. Equivalently, we can solve $\mathbf{X}_c \mathbf{X}_c^T \mathbf{v} = \lambda\mathbf{v}$. Both eigensystems are related. They share the same eigenvalues, and the eigenvectors $\boldsymbol{\alpha}$ and $\mathbf{v}$ are related by a linear transformation [5]. Noting that $\left(\mathbf{X}_c \mathbf{X}_c^T\right)_{ij} = (\mathbf{x}_c^i)^T \mathbf{x}_c^j$ we can see the role of the dot product in standard PCA.

Kernel PCA uses a mapping Φ into the high-dimensional space. In this space, we perform standard PCA. Thus, we proceed analogously and compute the eigenvectors of the *kernel matrix* $\mathbf{\Omega}$ with entries:

$$\mathbf{\Omega}_{ij} = \Phi(\mathbf{x}_c^i)^T \Phi(\mathbf{x}_c^j) \tag{6}$$

If the mapping Φ is chosen wisely, computing $\Phi(x_c^t)$ directly can be avoided by employing the *kernel trick*, where the dot product in the high-dimensional space is replaced by the evaluation of a kernel $\mathrm{K} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$:

$$\boldsymbol{\Omega}_{ij} = \Phi(\mathbf{x}_c^i)^T \Phi(\mathbf{x}_c^j) = \mathrm{K}(\mathbf{x}_c^i, \mathbf{x}_c^j) \tag{7}$$

Hence, we can perform kernel PCA by solving an $m \times m$ eigensystem $\boldsymbol{\Omega}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$. Taking this further by pre-multiplying the kernel matrix with some weighting matrix $\mathbf{V}$, we obtain the *weighted kernel PCA*:

$$\mathbf{V}\boldsymbol{\Omega}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha} \tag{8}$$

3.3 Spectral Clustering through Weighted Kernel PCA

AS10 show, how weighted PCA with different weighting schemes fit into the framework of spectral clustering. They heavily rely on previous work in [7], which shows the connection between weighted kernel PCA and spectral clustering methods. In [1], Alzate and Suykens extend their clustering approach in [7] to multiple clusters.

Depending on the choice of the weighting matrix $\mathbf{V}$, the weighted kernel PCA corresponds to different spectral clustering formulations. These correspondences are summarized in Table 1 in both [1] and [7]. For example, the relaxed solution for the normalized cut in Equation 1 is obtained by choosing $\mathbf{V} = \mathbf{D}^{-1}$ and selecting the eigenvector for the *second-largest* eigenvalue in Equation 8.

In order to illustrate this, consider the following example data matrix:

$$\mathbf{X} = \begin{pmatrix} -0.40 & 0.40 \\ -0.60 & -0.40 \\ 0.60 & -0.60 \\ 0.20 & 0.60 \\ 1.00 & -0.20 \end{pmatrix} \tag{9}$$

When choosing a radial basis function $s : (\mathbf{x}^i, \mathbf{x}^j) \mapsto \exp(-||x^i - x^j||_2^2)$ as similarity between the data points, we obtain the similarity graph as given in the toy example in Figure 1, and the example graph Laplacian in Equation 2 up to rounding. This similarity function s also qualifies as a valid kernel K as described in Section 3.2. Then the adjacency matrix of the toy similarity graph becomes the example kernel matrix:

$$\boldsymbol{\Omega}_{\text{toy}} = \begin{pmatrix} 1.00 & 0.51 & 0.14 & 0.67 & 0.10 \\ 0.51 & 1.00 & 0.23 & 0.19 & 0.07 \\ 0.14 & 0.23 & 1.00 & 0.20 & 0.73 \\ 0.67 & 0.19 & 0.20 & 1.00 & 0.28 \\ 0.10 & 0.07 & 0.73 & 0.28 & 1.00 \end{pmatrix} \tag{10}$$

The corresponding degree matrix is $\mathbf{D}_{\text{toy}} = \mathrm{diag}(2.41, 2.00, 2.29, 2.34, 2.18)$. Now, we can verify that the nonlinear principal component corresponding to the second-largest eigenvalue of $\mathbf{D}_{\text{toy}}^{-1}\boldsymbol{\Omega}_{\text{toy}}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$ is the same vector – up to scale – as $\boldsymbol{\alpha}_{\text{toy}}^{(2)}$ in Equation 2.

Based on the weighted kernel PCA framework, Alzate and Suykens provide an out-of-sample extension to spectral clustering.

3.4 Out-of-Sample Extension

The purpose of an out-of-sample extension is to cluster points that have not been part of the initial clustering process. Such an extension is helpful when not all data is available at the time of the initial clustering, or in applications where the dataset is too large. The latter is the case for images, where spectral clustering through weighted kernel PCA would require solving an eigensystem with approximately 150,000 dimensions. In order to make spectral clustering methods feasible again, we can partition the pixels into a *training* set used in spectral clustering, and a *test* set clustered by the out-of-sample extension after obtaining a clustering model from the training set.

There appears to be no simple way of treating out-of-sample data in spectral clustering based on the graph Laplacian. AS10 show how to formulate an out-of-sample extension in the framework of weighted kernel PCA.

In standard PCA, out-of-sample points can be projected onto the principal components. In doing so, we can obtain new features that can be assigned to clusters in the same way as in the reclustering step of the training set. Analogously, in kernel PCA, the idea is to map out-of-sample points into the high-dimensional space and project them onto its principal components. A recipe is given by equation 12.82 in [5].

3.5 Model Selection

Model selection describes the process of choosing appropriate parameters and complexity for the model to be learnt from available data. In the context of clustering, this usually comprises the number of clusters. In the application to image segmentation as described in AS10, the parameters also include the kernel bandwidth parameter for a radial basis function.

AS10 propose the *Balanced Line Fit* (BLF) criterion for model selection. The data to be clustered is partitioned into three sets: the training set for obtaining the model, the validation set for selecting a model, and the test set for speeding up the clustering process. The BLF is a measure evaluated on the score variables of the validation data. It is a linear combination of a "linefit" and a "balance" criterion. The score variables for the validation data are obtained using the out-of-sample extension.

AS10 define the balance to be the ratio between the smallest cluster cardinality and the largest cluster cardinality. For each cluster, the line fit requires computing the covariance matrix $\mathbf{C}^{(p)} \in \mathcal{R}^{(k-1)\times(k-1)}$ of the centered score variables, only involving data assigned to this particular cluster. Then, the line fit evaluates how well the variance of these score

variables is explained by the principal component of the in this cluster. This principal eigenvector is computed from $\mathbf{C}^{(p)}$ by PCA, and can also be interpreted as a line, hence the name "line fit". In the best case, all the centered validation data lies on that line which is the case if there is only one non-zero eigenvalue.

3.6 Application to Image Segmentation

This section discusses the application of clustering to image segmentation. AS10 extract local features from the image for each pixel as follows: First, the image is quantized using minimum variance quantization [8] into eight different colors. A local color histogram – computed from a 5x5 window around a pixel – can then serve as features for this pixel. AS10 use the $\mathcal{X}^2$-distance between two of such histograms to define similarity between two pixels [1].

4 State of the Art

There are many different approaches to clustering. In this section, two more recent spectral clustering approaches to unsupervised image segmentation are presented because of their relation to AS10. Peherstorfer et al. provide an alternative approach to spectral clustering with out-of-sample extensions on sparse grids [9]. Socher et al. show a supervised, semantic approach to image segmentation using neural networks [10].

4.1 Nyström Extension

Throughout the paper, AS10 refer the reader to [11] for an alternative out-of-sample extension based on the Nyström method [12]. The Nyström extension [11] approximates the eigenvectors of the normalized graph Laplacian associated with all data. It does so by extending the eigenvectors from the normalized graph Laplacian associated with the training set. It uses eigenfunctions computed with the Nyström method in order to extrapolate the eigenvector components at out-of-sample points.

4.2 Sparse Grids

A recent spectral clustering approach [9] learns eigenfunctions on a sparse grid. While the out-of-sample extensions for Laplacian Eigenmaps in [11] and [9] are similar in spirit, the sparse grid approach is an attempt to alleviate the curse of dimensionality. Here the eigenfunction basis is defined on a sparse grid, rather than on the training set. Figure 3 shows such a sparse grid. The key idea in [9] is that the size of the eigenvalue problem depends only on the number of points on the sparse grid.

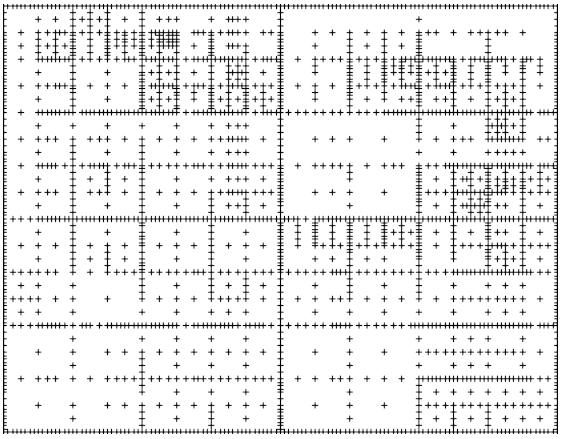

Fig. 3. An adaptively refined sparse grid.

5 Discussion

This section discusses the experimental results presented in AS10. The strengths and weaknesses of the paper are addressed, and the paper is viewed in the context of the existing literature.

5.1 Experimental Results

Alzate and Suykens show the performance of their reference implementation on toy problems and on the Berkeley Segmentation Dataset[2] [13].

The toy problems are helpful for the reader to see the proposed clustering algorithm and the model selection criterion at work. For example, a situation with a perfect line fit is illustrated in Figure 4a of [1]. In Figure 4c of [1], the meaning of the axis labels is never explained although it is possible to infer the meaning from the context. The toy problems are evaluated using the adjusted Rand index (ARI), which can be used because ground truth data is known for these artificial data. As expected, one of the toy examples is three concentric rings where k-means is bound to fail, one particular case where spectral clustering is superior.

The most interesting experiments were performed on the test images of the Berkeley Segmentation Dataset. Researchers using this dataset must play fair, and not manually tune their learning algorithms on the test set. As far as can be seen from the paper, AS10 abide to this rule and exploit the ability of their system to select a model automatically. AS10 have evaluated their system on all 100 test images of the dataset, when

[2] BSDS300: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench

Fig. 4. An image from the Berkeley Segmentation Dataset [13].

comparing performance using the F-measure against manually collected ground truth [11]. Thus, no selection was made after having obtained the results. However, there was no explanation how the 20 test images printed in Figure 10 of [1] were selected.

At the time of writing, no implementation was made available to the public by the authors, although there seem to be plans to do so. The lack of a reference implementation renders repeating the experiments difficult. However, the results on the test images are visually pleasing, and AS10 outperform [11] in terms of the F-measure in most of the test images.

5.2 Paper in Context

Alzate and Suykens contribute to the field of Machine Learning. They provide detailed insights between weighted kernel PCA and other spectral clustering methods. While the links between kernel PCA and spectral clustering have been known before (see [14]), AS10 took efforts to present, integrate, and recombine the broad relevant knowledge from the existing literature into a complete clustering method. This includes the explicit treatment of the optimized loss function and an algorithm recipe. They propose a new automatic model selection criterion.

5.3 Literature

This section refers to the literature that is connected to the multiway spectral clustering approach in AS10. Additionally, references for further reading are given. Von Luxburg [3] describes spectral clustering methods from viewpoints of graph partitioning, random walks, and perturbation theory. Shi and Malik [4] introduce the normalized cut, and provide a physical interpretation of spectral clustering in terms of a spring-mass

system. Chung [15] treats spectral graph theory in general. The fundamentals of PCA and kernel machines can be studied in the textbooks by Alpaydin [2] and by Bishop [5]. Filippone et al. [14] provide a survey paper on kernel methods and spectral methods for clustering. Alzate and Suykens [7] often refer to a previous paper about weighted kernel PCA. They compare their approach with the work of Fowlkes et al. [11]. Wu [8] shows how to efficiently implement the minimum variance quantization employed by AS10, whereas Heckbert [16] gives a more general introduction to color quantization.

6 Conclusion

Alzate and Suykens have shown how to do spectral clustering using weighted kernel PCA in general. Rather than focusing on a single stage in the clustering process, they present a whole pipeline for image segmentation from feature extraction over similarity measures to out-of-sample extension and model selection. The performance is comparably with at least one other state-of-the-art approach. The paper lacks an implementation. However, Alzate and Suykens evaluated the system on real data rather than just on artificial data which makes the experiments more credible. Future work for Alzate and Suykens might involve providing an implementation which shows how to transform the theory into code efficient enough to segment images.

References

1. Alzate, C., Suykens, J.A.K.: Multiway Spectral Clustering with Out-of-Sample Extensions through Weighted Kernel PCA. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(2) (2010) 335–47
2. Alpaydin, E.: Introduction to Machine Learning. 2 edn. MIT Press (2010)
3. von Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing **17**(4) (2007) 395–416
4. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8) (2000) 888–905
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
6. Miranda, A.A., Le Borgne, Y.A., Bontempi, G.: New Routes from Minimal Approximation Error to Principal Components. Neural Processing Letters **27**(3) (2008) 197–207

7. Alzate, C., Suykens, J.A.K.: A Weighted Kernel PCA Formulation with Out-of-Sample Extensions for Spectral Clustering Methods. In: Proceedings of the 2006 IEEE International Joint Conference on Neural Networks, IEEE (2006) 138–144

8. Wu, X.: Efficient Statistical Computations for Optimal Color Quantization. In Arvo, J., ed.: Graphics Gems II. Academic Press (1991) 126–133

9. Peherstorfer, B., Pflüger, D., Bungartz, H.J.: A Sparse-Grid-Based Out-of-Sample Extension for Dimensionality Reduction and Clustering with Laplacian Eigenmaps. In: AI 2011: Advances in Artificial Intelligence. Volume 7106 of Lecture Notes in Computer Science. Springer (2011) 112–121

10. Socher, R., Lin, C.C., Ng, A.Y., Manning, C.D.: Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In: Proceedings of the 26th International Conference on Machine Learning. (2011)

11. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral Grouping Using the Nystrom Method. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(2) (2004) 214–225

12. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes: The Art of Scientific Computing. Cambridge University Press (2007)

13. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: Proceedings of the 2001 IEEE International Conference on Computer Vision. Volume 2. (2001) 416–423

14. Filippone, M., Camastra, F., Masulli, F., Rovetta, S.: A Survey of Kernel and Spectral Methods for Clustering. Pattern Recognition **41**(1) (January 2008) 176–190

15. Chung, F.R.K.: Spectral Graph Theory. Volume 92 of CBMS Regional Conference Series in Mathematics. American Mathematical Society (February 1997)

16. Heckbert, P.: Color Image Quantization for Frame Buffer Display. Proceedings of the 9th Annual Conference on Computer Graphics and Interactive Techniques **16**(3) (1982) 297–307