In *3rd International Workshop on Statistical and Computational Theories of Vision*
at ICCV, Nice (France), October 12, 2003.

Efficient Graph Cuts for Unsupervised Image Segmentation using Probabilistic Sampling and SVD-based Approximation

J. Keuchel, C. Schnörr
Computer Vision, Graphics, and Pattern Recognition Group
Department of Mathematics and Computer Science
University of Mannheim, 68132 Mannheim, Germany
{jkeuchel,schnoerr}@uni-mannheim.de

Abstract

The application of graph theoretic methods to unsupervised image partitioning has been a very active field of research recently. For weighted graphs encoding the (dis)similarity structure of locally extracted image features, unsupervised segmentations of images into coherent structures can be computed in terms of extremal cuts of the underlying graphs.

In this context, we focus on the normalized cut criterion and a related recent convex approach based on semidefinite programming. As both methods soon become computationally demanding with increasing graph size, an important question is how the computations can be accelerated. To this end, we study an SVD approximation method in this paper which has been introduced in a different clustering context. We apply this method, which is based on probabilistic sampling, to both segmentation approaches and compare it with the Nyström extension suggested for the normalized cut. Numerical results confirm that by means of the sampling-based SVD approximation technique, reliable segmentations can be computed with a fraction (less than 5%) of the original computational cost.

1 Introduction

Motivation and Overview. In the context of unsupervised image segmentation, recent research has focused on approaches utilizing various graph-theoretical methods [1, 2, 3, 4]. In principle, such approaches take as input data just a set of pairwise (dis)similarities of locally extracted image features and partition an image by successively computing extremal cuts of the corresponding weighted graph. Since the global combinatorial optimization problem of computing such cuts is NP-hard, methods from spectral graph theory [5] have successfully been applied to compute suboptimal cuts [1, 6]. Recently, a novel relaxation technique (in the field of computer vision) based on semidefinite programming has been suggested for *constrained* extremal cuts [4], providing an alternative (depending on the application) to *normalized* extremal cuts considered in [1]. A brief review of both these approaches will be given in Section 2.

Unfortunately, these approaches become computationally demanding (or even intractable) with increasing size of the images, especially as the corresponding similarity matrices of the graphs do no longer fit into memory (e.g. for an image of 240×160 pixels, the similarity matrix contains $38400^2 \approx 1500$ mill. entries). In the present paper, we therefore focus on the problem to devise computationally efficient (concerning time and memory requirements) versions of these approaches. This will be accomplished by solving an eigenvalue problem or a semidefinite program, respectively, for a probabilistically sampled small subset of the input data only, the solution of which generalizes well to that of the original large-scale problem. For normalized cuts, such an approach has been proposed recently [7] based on a method originating from the numerical treatment of integral equations (see, e.g., [8]).

Contribution. A natural alternative to the Nyström extension considered in [7] are probabilistic SVD approximation methods which have been introduced in a different clustering context [9]. In Section 3, we contrast both sampling methods with each other and point out their overall structural similarity as well as a subtle difference resulting in an advantage of the sampling-based SVD technique regarding computational robustness. A novel extension of the probabilistic SVD approximation method to the normalized cut and the semidefinite programming approaches is the objective of Section 4. In particular, this extension allows the application of the semidefinite programming approach [4] to real images which otherwise is infeasible due to the size of the matrices involved. Finally, Section 5 collects numerical results which reveal the success of the sampling-based SVD approximation approach in the context of image segmentation: While maintaining a reliable quality, the computational efficiency is increased by at least 95%. Several examples of real world image segmentations also demonstrate the similarity of the different sampling-based approaches in practice.

Finally, as semidefinite programming is nowadays used in a large number of diverse fields (see, e.g., [10]), we would like to point out that our results may be relevant for other application fields besides image segmentation as well.

2 Image Partitioning via Graph Cuts

In this section, we briefly review two approaches to solve unsupervised binary partitioning problems based on extremal graph cuts: The normalized cut method [1] and a semidefinite programming approach [4]. To find a segmentation of the image into more than two parts, both methods can be applied hierarchically on the obtained segments.

In the following, let the image be represented by a weighted graph $G(V, E)$ with locally extracted image features as vertices V and pairwise similarity values $w_{ij} \in \mathbb{R}_0^+$ as edge-weights. The binary partitioning problem then consists in finding two coherent groups S and $\overline{S} = V \setminus S$ within V by minimizing some given cost function $f(S, \overline{S})$ which depends on the weight of the corresponding cut: $\mathrm{cut}(S, \overline{S}) = \sum_{i \in S, j \in \overline{S}} w_{ij}$.

In the following, let $|V| = n$, and denote by $W \in \mathbb{R}^{n \times n}$ the symmetric adjacency matrix containing the similarity values w_{ij}. Furthermore, let D denote the diagonal matrix with the degrees $d_i = \sum_{j \in V} w_{ij}$ of the vertices on its diagonal, and let $L = D - W$ denote the corresponding Laplacian matrix. Representing a partition by an indicator vector $x \in \{-1, +1\}^n$, the weight of a cut then is given by:

$$\mathrm{cut}(S, \overline{S}) = \frac{1}{4} x^\top L x. \tag{1}$$

2.1 Normalized Cut

Shi and Malik [1] suggested to find a partitioning based on minimizing the following objective function (*normalized cut*):

$$\frac{\mathrm{cut}(S, \overline{S})}{w(S)} + \frac{\mathrm{cut}(S, \overline{S})}{w(\overline{S})} = \frac{1}{2} \left(\frac{x^\top L x}{x^\top D(x+e)} + \frac{x^\top L x}{x^\top D(x-e)} \right), \tag{2}$$

where $e = (1, \ldots, 1)^\top$, and $w(S) = \sum_{i \in S} d_i$ denotes the sum of the degrees within S. The normalization of the cut-value in (2) serves the purpose to avoid unbalanced partitions which are likely when $\mathrm{cut}(S, \overline{S})$ is minimized directly.

The normalized cut problem can be rewritten in the following way:

$$\begin{aligned} \inf_{x \in \{-b, 1\}^n} \; & \frac{x^\top L x}{x^\top D x} \\ \text{s.t.} \quad & e^\top D x = 0, \end{aligned} \tag{3}$$

with $b = \frac{w(S)}{w(\overline{S})}$ not being known beforehand. Since this optimization problem is intractable, the constraint on the entries of x is dropped in practice. Thus, the resulting relaxation becomes the problem to find the second *smallest* generalized eigenvector of L, as the constraint in (3) demands that the solution vector is perpendicular to the smallest generalized eigenvector e. Finally, the integer constraint is taken into account to obtain an indicator vector by thresholding this eigenvector using some suitable criterion [1].

Defining the normalized similarity matrix $P = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, the relaxation of (3) is equivalent to the following maximization problem:

$$\begin{aligned} \sup_{\|x\|=1} \quad & x^\top P x \\ \text{s.t.} \quad & e^\top D^{\frac{1}{2}} x = 0. \end{aligned} \tag{4}$$

As $D^{\frac{1}{2}} e$ is the eigenvector of P corresponding to the largest eigenvalue 1, this problem can be solved by finding the second *largest* eigenvector of P. Multiplication of this eigenvector with $D^{-\frac{1}{2}}$ gives (after appropriate thresholding) a binary approximative solution of the original normalized cut problem.

2.2 Semidefinite Programming

An alternative technique to find balanced partitionings of a graph is based on a semidefinite programming relaxation of a classical approach from spectral graph theory (see, e.g., [5]) which uses constrained extremal cuts. As a starting point, consider the following problem formulation:

$$\begin{aligned} \inf_{x \in \{-1,+1\}^n} \quad & x^\top L x \\ \text{s.t.} \quad & e^\top x = 0. \end{aligned} \tag{5}$$

This criterion has a clear interpretation: Determine a cut with minimal weight subject to the constraint that each group has an equal number of vertices. Thus, instead of *normalizing* the objective function as in (3), in this case an *additional balancing constraint* is used to compute favorable partitions. Problem (5) may be solved in the same way as the normalized cut problem: Dropping the integer constraint, the second smallest eigenvector of L has to be computed (as e is the smallest one) and is thresholded afterwards using some suitable criterion.

In [4], the authors propose to use a more advanced method to relax and solve the problem (5), which not only takes into account the integer constraint on x in a better way, but also renders appropriate thresholding unnecessary: First, the problem variables are lifted into a higher dimensional matrix space by observing that the objective function in (5) can be rewritten as $x^\top L x = \operatorname{tr}(L x x^\top)$. Replacing the positive semidefinite rank one matrix $x x^\top$ by an arbitrary positive semidefinite matrix X, and lifting the constraints to the higher dimensional space accordingly, one obtains the following relaxation:

$$\begin{aligned} \inf_{X \succeq 0} \quad & \operatorname{tr}(LX) \\ \text{s.t.} \quad & \operatorname{tr}(ee^\top X) = 0 \\ & \operatorname{tr}(e_i e_i^\top X) = 1 \quad i = 1, \dots, n, \end{aligned} \tag{6}$$

with $e_i \in \mathbb{R}^n$ denoting the ith unit vector, and $X \succeq 0$ meaning that the matrix X has to be positive semidefinite.

The problem (6) belongs to the class of *semidefinite programs*, which can be solved using interior point methods (see, e.g., [11, 12]). To finally recover an integer solution x from the computed solution X to (6), the randomized-hyperplane technique as described in [13] is used.

Note that this semidefinite relaxation technique represents an alternative approach to solve the binary partitioning problem which allows for inclusion of additional, application dependent constraints on partitions. It can not be applied to solve the normalized cut problem (3), however, due to the normalization of the objective function.

3 Approximation based on Probabilistic Sampling

Both image segmentation methods presented in the last section soon become computationally infeasible when the size of the image increases, as the corresponding similarity matrices get very large and thus may no longer fit into memory. To overcome this problem, we suggest to use an SVD approximation method based on probabilistic sampling which was proposed by Drineas et al. [9] in a different clustering context. The basic idea of this algorithm is to approximate a given problem matrix $M \in \mathbb{R}^{n \times n}$ by a matrix $\hat{M}$ of lower rank $k \ll n$. For this approximate matrix, the image segmentation problems can be solved by using only a small subset of the input data, and the solutions generalize well to the original large-scale problem.

Before presenting the SVD approximation algorithm in Section 3.2, we will briefly discuss how to pick the sample points. Moreover, we will show in Section 3.3 how the probabilistic SVD algorithm is related to the Nyström extension considered in [7].

3.1 Sample Selection

The success of any sampling approach depends on selecting suitable samples, i.e. to pick points which give enough information to result in a good approximation to the complete problem. To this end, Drineas et al. [9] propose to sample the columns of the problem matrix M with probabilities proportional to their squared norm. Based on such a selection, they are able to prove a theoretical bound on the approximation quality, which, however, depends on a large number of samples to be selected. On the other hand, they also state that in practice it suffices to pick a much smaller number of samples to obtain good approximation results.

If the matrix M is *dense*, Frieze et al. [14] argue that the probability $P_i = \frac{1}{n}$ may be taken for picking a column i to fulfill the requirements of the proven bound. This results in uniformly sampling the columns of M, so that given the number of samples s, they can be selected independently at random from the whole set of columns. For the Nyström extension, Belongie et al. [15] successfully used the same proceeding of picking the samples of their dense problem matrices uniformly at random. Due to these facts, and as the matrices of the binary partitioning problems considered in this paper are dense, we also rely on this simple sample selection procedure.

To facilitate the analysis of the following sections, we will assume that the entries of the symmetric problem matrix M were reordered so that the first s columns and rows of M correspond to the s selected samples. Note that such reordering does not change the spectral structure of the matrix! The matrix $M \in \mathbb{R}^{n \times n}$ can then be subdivided into smaller submatrices:

$$M = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix}, \tag{7}$$

with $A \in \mathbb{R}^{s \times s}$, $B \in \mathbb{R}^{s \times n-s}$ and $C \in \mathbb{R}^{n-s \times n-s}$.

3.2 Probabilistic SVD Approximation Algorithm

It is a well known fact from Linear Algebra [16], that the best rank k approximation of a matrix M in a suitable matrix norm can be derived from the singular value decomposition (SVD) of M: If

$$M = \sum_{i=1}^{n} \sigma_i q_i p_i^\top$$

denotes the SVD of M, then the best rank k approximation is given by

$$\min_{D:\mathrm{rank}(D)=k} \|M - D\| = \|M - Q_k Q_k^\top M\|, \tag{8}$$

with $Q_k \in \mathbb{R}^{n \times k}$ comprising the left orthonormal singular vectors q_i for the k largest singular values $\sigma_1 \geq \cdots \geq \sigma_k$ of M. More specifically, the approximation (8) holds for every unitarily invariant matrix norm (cf. [17]), e.g. the spectral norm $\|.\|_2$ or the Frobenius norm $\|.\|_{\mathrm{F}}$.

Drineas et al. [9] now propose to use a sampling-based SVD algorithm to approximate the top k left singular vectors of M, to arrive at an approximation to the best rank k approximation $Q_k Q_k^\top M$ to M:

$$M \approx Q_k Q_k^\top M \approx \hat{Q}_k \hat{Q}_k^\top M = \hat{M},$$

with the matrix $\hat{Q}_k \in \mathbb{R}^{n \times k}$ containing the set of k orthonormal vectors which approximate the largest left singular vectors of M. These approximate singular vectors $\hat{q}_i$ are calculated as the top k left singular vectors of the sampled $n \times s$ submatrix

$$S = \begin{pmatrix} A \\ B^\top \end{pmatrix}.$$

This can be accomplished by finding the eigenvectors w_i corresponding to the k largest eigenvalues $\lambda_1 \geq \cdots \geq \lambda_k \geq 0$ of the much smaller $s \times s$ matrix $S^\top S$, and calculating $\hat{q}_i = S w_i / \|S w_i\| = S w_i / \sqrt{\lambda_i}$ for $i = 1, \ldots, k$.

Note that the left singular vectors of S are the same as the top singular vectors (and the eigenvectors) of the symmetric positive semidefinite matrix $SS^\top$, and that the same holds for the matrices M and $MM^\top$. This yields a different interpretation of the SVD approximation algorithm: The matrix $MM^\top$ is approximated with the matrix $SS^\top$ of smaller rank s in the following way:

$$\begin{aligned} MM^\top &= \begin{pmatrix} AA^\top + BB^\top & AB + BC^\top \\ (AB)^\top + CB^\top & B^\top B + CC^\top \end{pmatrix} \\ &\approx \begin{pmatrix} AA^\top & AB \\ (AB)^\top & B^\top B \end{pmatrix} \\ &= \begin{pmatrix} A \\ B^\top \end{pmatrix} \begin{pmatrix} A^\top & B \end{pmatrix} = SS^\top \\ &= \hat{Q}_s \Sigma \hat{Q}_s^\top, \end{aligned} \tag{9}$$

where the thin[1] singular value decomposition $\hat{Q}_s \Sigma \hat{Q}_s^\top$ of $SS^\top$ is obtained from the eigenvalue decomposition of the positive semidefinite and much smaller $s \times s$ matrix $S^\top S$:

$$S^\top S = \begin{pmatrix} A^\top & B \end{pmatrix} \begin{pmatrix} A \\ B^\top \end{pmatrix} = Y \Sigma Y^\top \tag{10}$$

$$\Rightarrow \quad \hat{Q}_s = \begin{pmatrix} A \\ B^\top \end{pmatrix} Y \Sigma^{-\frac{1}{2}}, \tag{11}$$

where Σ denotes the diagonal matrix containing the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_s \geq 0$ of $S^\top S$ (which are equal to the top singular values of $SS^\top$) on the diagonal.

3.3 Comparison to the Nyström Extension

Belongie et al. [15] propose another sampling-based method for spectral clustering which is derived from the Nyström extension [8]: The matrix M is directly approximated with a rank s matrix that is calculated from the sampled submatrix S by implicitly approximating the submatrix C with the matrix $B^\top A^{-1} B$:

$$\begin{aligned} M &= \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix} \\ &\approx \begin{pmatrix} A & B \\ B^\top & B^\top A^{-1} B \end{pmatrix} \\ &= \begin{pmatrix} A \\ B^\top \end{pmatrix} A^{-1} \begin{pmatrix} A^\top & B \end{pmatrix} = S A^{-1} S^\top \\ &= \hat{P}_s \Lambda \hat{P}_s^\top. \end{aligned} \tag{12}$$

The thin SVD $\hat{P}_s \Lambda \hat{P}_s^\top$ of the approximating matrix $\hat{M} = SA^{-1}S^\top$ can then be calculated by finding the eigenvector decomposition of the smaller $s \times s$ matrix $A' = A + A^{-\frac{1}{2}} BB^\top A^{-\frac{1}{2}} =$

[1] In this context, thin SVD means the SVD of the matrix without the singular vectors corresponding to the zero singular values.

$A^{-\frac{1}{2}}S^\top SA^{-\frac{1}{2}}$:

$$A^{-\frac{1}{2}}S^\top SA^{-\frac{1}{2}} = A^{-\frac{1}{2}}\left(A^\top\ B\right)\begin{pmatrix} A \\ B^\top \end{pmatrix} A^{-\frac{1}{2}} = V\Lambda V^\top \tag{13}$$

$$\Rightarrow \quad \hat{P}_s = \begin{pmatrix} A \\ B^\top \end{pmatrix} A^{-\frac{1}{2}} V \Lambda^{-\frac{1}{2}}, \tag{14}$$

with Λ denoting the diagonal matrix containing the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_s$ of A' on the diagonal in descending order, and $V \in \mathbb{R}^{s\times s}$ denoting the matrix of the corresponding eigenvectors. In practice, only the top k eigenvectors of the matrix A' are calculated, which leads to the rank k approximation $\hat{P}_k \Lambda \hat{P}_k^\top$ of M.

The similarity of this approach to the probabilistic SVD algorithm is revealed by comparing the approximations (9) and (12): Whereas the probabilistic SVD algorithm calculates the eigenvectors $\hat{q}_i$ of the matrix $SS^\top$ as an approximation to the top eigenvectors of $MM^\top$ (which are the same as the top eigenvectors of M if the matrix M is positive semidefinite), the Nyström extension approximates the top eigenvectors of M by computing the eigenvectors $\hat{p}_i$ of the matrix $SA^{-1}S^\top$. Thus, for positive semidefinite matrices M, both approximations will become very similar when the submatrix A is close to the identity matrix, $A \approx I$. Moreover, it can be verified that $S^\top \hat{Q}_s \hat{Q}_s^\top S = S^\top \hat{P}_s \hat{P}_s^\top S$, so the inner products of the columns of the sample matrix S after projecting them onto the subspaces spanned by the approximative eigenvectors are the same for both approaches.

An essential requirement for the Nyström extension is that the inverse A^{-1} and the square root $A^{\frac{1}{2}}$ of the submatrix A exist. The second requirement is fulfilled if the matrix M is positive semidefinite: In this case, A is also positive semidefinite, so that the square root $A^{\frac{1}{2}}$ always exists. Nevertheless, the inverse A^{-1} may still not be calculated if any of the eigenvalues of A are 0. As a remedy for this case, Belongie et al. [15] propose to use the pseudoinverse instead of A^{-1}. Moreover, they also present a modification of the Nyström method which can be applied to indefinite problem matrices. However, besides increasing the computational effort, this modification may lead to a significant loss in numerical precision [15].

In contrast to that, the probabilistic SVD approximation algorithm does not need to calculate any inverse matrices, which makes the approach computationally less complex and less sensitive when being applied to nearly singular matrices. Indeed, it can also be used to calculate rank k approximations for non-positive semidefinite matrices. However, in this case, one has to be cautious when applying the sampling technique to spectral partitioning problems which are based on the largest eigenvectors of the problem matrix: As the largest singular vectors (which are approximated in this case) could correspond to eigenvectors of negative eigenvalues, they may yield incorrect partitionings!

4 Application to Binary Partitioning

In this section, we describe how the image segmentation problems presented in Section 2 can be solved by applying the probabilistic SVD approximation procedure introduced in Section 3.2.

4.1 Normalized Cut

Provided that the normalized similarity matrix P from (4) is positive semidefinite, both sampling techniques presented in the previous section can be applied directly to P to approximate its top eigenvectors. Otherwise, since all eigenvalues of P are known to be larger than -1, we can transform P into a positive semidefinite matrix by adding the identity matrix: $\tilde{P} = P + I$. This transformation increases the eigenvalues by 1, but does not change the eigenvectors or the order of the eigenvalues. However, as the positive eigenvalues are mostly dominating for real image data, this transformation is usually not necessary in practice. [2]

The calculation of a binary solution based on the approximative eigenvectors poses two problems: First, the information contained in the second largest eigenvector of the full problem matrix may now be shifted to another of the top approximative eigenvectors, so using only the second approximative eigenvector may be misleading. Second, for large problem instances as they are naturally obtained from real images with thousands of pixels, it is not possible to use the complete problem matrix P to calculate the optimal binary solution, due to time and memory restrictions.

To handle these problems, we use a method to calculate the binary solution which is still directly based on the normalized cut criterion, but only needs the sampled part of the problem matrix W and the complete vector $d = De$ of the degrees: After multiplying the approximative eigenvectors $\hat{Q}_k$ of the problem matrix P with $D^{-\frac{1}{2}}$ to find approximative eigenvectors of the original normalized cut problem, the rows of the matrix $D^{-\frac{1}{2}}\hat{Q}_k$ are normalized to unit length to project them onto the unit sphere (cf. [6, 18]). Then using individually one of the top projected eigenvectors, the threshold is computed which gives a binary vector that minimizes the following adjusted version of (2):

$$\frac{x^\top L_s x}{x^\top D_s(x+e)} + \frac{x^\top L_s x}{x^\top D_s(x-e)}, \tag{15}$$

where L_s and D_s are obtained from L and D, respectively, by setting all columns to zero which correspond to points that were not sampled. Doing this for each of the first few projected eigenvectors, the final solution is given by the binary vector which minimizes (15).

The interpretation of the adjustment (15) is easy: Instead of using the complete problem graph to find a cut, a sparser graph is examined containing only the edges between the samples

[2] Note that in [15], a slightly different method is proposed: Instead of P, the original similarity matrix W is approximated by the Nyström method and normalized afterwards. But as this proceeding is not applicable for the SVD approximation method, we assume that the sampled part S of the problem matrix P can be calculated exactly by computing the degree-vector $d = De$ once before starting the segmentation.

with their full weight and the half-weighted edges between samples and non-samples. In this way, the confidence in the eigenvector entries after the sampling process is also represented in the objective function.

A different, commonly used technique to obtain a partitioning (see, e.g., [7, 18]) is to calculate an embedding of the points into the k-dimensional space $\mathbb{R}^k$ from the row entries of the approximative eigenvectors $\hat{Q}_k$, and to use the k-means algorithm in this space to identify clusters. In this way, the image can be split into more than 2 subsets directly. But an important drawback of the use of k-means is that the solution is no longer based on the original objective criterion to minimize the normalized cut. In fact, it has been shown [19] that standard embedding techniques yield a grouping problem in the corresponding vector space which is not equivalent to the original normalized cut criterion. For this reason, and as we focus on *binary* partitionings, we did not study this method in this paper.

4.2 Semidefinite Programming

To apply the probabilistic SVD approximation technique to the semidefinite programming approach presented in Section 2.2, the following ideas are used: First we transform the minimization problem (6) to a maximization problem by using the fact that

$$\inf_{X \succeq 0} \operatorname{tr}(LX) = \sum_{i=1}^{n} d_i - \sup_{X \succeq 0} \operatorname{tr}(WX),$$

as the second constraint in (6) ensures that the diagonal of X contains only ones. If the similarity matrix W is not positive semidefinite, we can transform it appropriately by adding a multiple of the identity matrix cI with c being large enough (cf. Section 4.1). We now use the fact that the randomized hyperplane technique [13] calculates an integer solution x based on the incomplete Cholesky decomposition of the solution matrix to (6): Using this decomposition, $X = GG^\top$, the objective function becomes $\operatorname{tr}(G^\top W G)$, with the rows G_i of G having unit norm due to the second constraint in (6). Disregarding the first constraint, the complete eigenvector decomposition $W = Q\Lambda Q^\top$ yields a special instance of this objective function:

$$\sup_{G_i G_i^\top = 1} \operatorname{tr}(G^\top W G) \geq \operatorname{tr}(Q^\top W Q) = \sum_{i=1}^{n} \lambda_i,$$

as $QQ^\top = Q^\top Q = I$. For this reason, we suggest to calculate an approximative Cholesky decomposition of the solution matrix X in the same way as the approximative top eigenvectors of W are obtained by using the sampling-based SVD approximation method.

In more detail, the solution steps are as follows:

1. Calculate the sampled submatrix S of W to obtain the matrix $S^\top S$.

2. Solve the following *small-sized* semidefinite program:

$$\begin{aligned} \sup_{\tilde{X} \succeq 0} \quad & \operatorname{tr}(S^\top S \tilde{X}) \\ \text{s.t.} \quad & \operatorname{tr}(ee^\top \tilde{X}) = 0 \\ & \operatorname{tr}(e_i e_i^\top \tilde{X}) = 1 \quad i = 1, \ldots, s. \end{aligned} \tag{16}$$

3. Compute the approximative Cholesky factor $\hat{G} \in \mathbb{R}^{n \times s}$ of the solution matrix X of (6) from the Cholesky factor $\tilde{G}$ of the solution $\tilde{X} = \tilde{G}\tilde{G}^\top$ of (16): $\hat{G} = S\tilde{G}$.

4. As for the normalized cut, normalize the rows of $\hat{G}$. In this way, the original norm constraint on the rows of the Cholesky factor is satisfied.

5. Adapt the randomized hyperplane technique [13] by using random vectors r from the unit sphere in $\mathbb{R}^s$ to calculate binary vectors x: $x = \operatorname{sgn}(\hat{G}r)$, with $\operatorname{sgn}(y)$ meaning to apply the sgn-function to each entry of the vector y. The final solution is the binary vector x which minimizes the following adjusted version of the objective function in (5):

$$x^\top L_s x,$$

with L_s being defined as in (15). As this is equivalent to maximizing $x^\top W_s x$, note that the degree vector d is not needed for the semidefinite programming approach!

5 Experiments and Discussion

In this section, we present the results of applying the probabilistic SVD approximation method to binary partitioning problems based on the normalized cut criterion and the semidefinite programming (SDP) relaxation, respectively. Additionally, we also provide a comparison with the results obtained for the Nyström extension for normalized cuts.

For all the experiments, the similarity values w_{ij} were calculated from the Mahalanobis distances between the extracted image feature vectors y_i and y_j:

$$w_{ij} = e^{-\frac{1}{2}(y_i - y_j)^\top \Sigma^{-1}(y_i - y_j)},$$

with Σ denoting the diagonal matrix containing the scaling factors σ_k for the entries of the feature vectors. Each feature vector comprises the position and the color in the perceptually uniform L*u*v* space of the corresponding pixel in the image (or only the position for point sets). As the resulting similarity matrix W and the corresponding normalized matrix P are positive definite, they can be used unchanged for all the applications. More intricate similarity measures could of course also be employed in this context [4]. However, as the main objective of this paper is to show the efficiency of sampling-based techniques to solve large scale image segmentation problems, we did not work on more elaborate similarity measures.

5.1 Performance Evaluation

To measure the performance of the sampling-based versions of the partitioning methods statistically, we created two different point sets as depicted in Figures 1(a) and (b). Using the complete similarity matrix, both the normalized cut and SDP were able to separate the clusters correctly. For several percentages of sampled points, we then computed the approximate solutions for both partitioning approaches using the sampling techniques and compared them to the optimal solution by counting the number of misclassified points. To derive some significant statistics, this experiment was repeated 100 times for each percentage value with different samples selected.

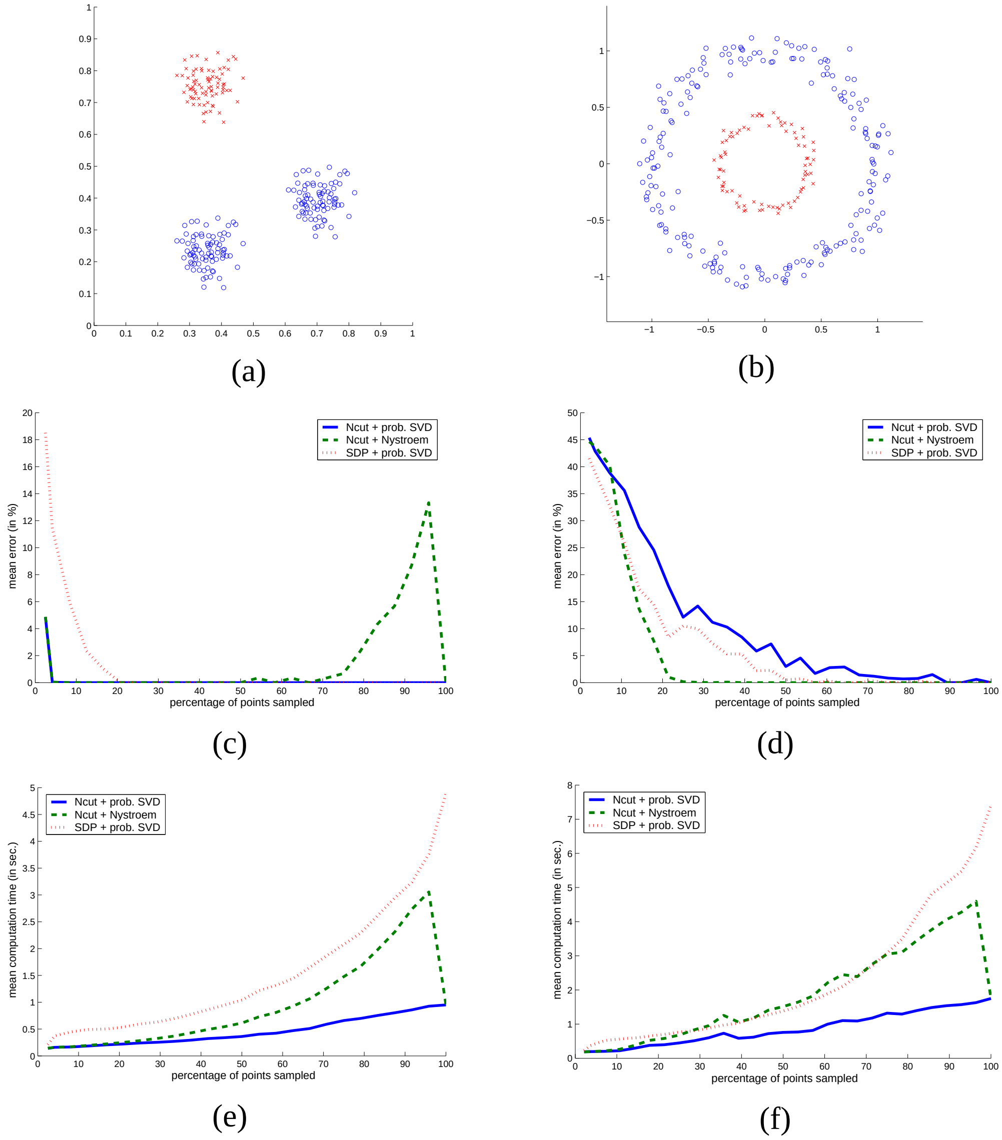
The results in Figures 1(c)-(f) reveal the good performance of all methods, also for relatively small sample rates, especially for the quite simple example in Figure 1(a): In this case, the mean error is always lower than 5% if at least 10% of the points are sampled. Note that for the example in Figure 1(b), small sample numbers result in a significant loss of the structure as the similarity values are based on Euclidean distances only, which makes this problem quite intricate. Although for this reason, the mean error increases for smaller numbers of samples, it should be mentioned that each method was still able to find the optimal solution at least once down to a sample rate of 10%.

The bad performance of the normalized cut method based on the Nyström extension for large sample numbers depicted in Figure 1(c) is due to the fact that for this example the similarity matrix W is nearly singular, which leads to inaccurate results in the calculation of the inverse A^{-1}. Additionally, this calculation of an inverse matrix also leads to an increase of the computational effort for the Nyström method, thus making it inefficient for higher sample rates. On the other hand, note that especially for SDP sampling strongly reduces the computational effort (*quadratically* with the number of points), so that it becomes comparable to the normalized cut sampling methods for small sample sizes.

5.2 Image Segmentation

Figure 2 gives the results for a small color image when 6.2% of the points are sampled. For this example, we also computed the optimal normalized cut and SDP solutions (Figures 2(b) and (c)) to compare the performance of the different techniques. While all binary partitions calculated based on sampling are reasonable approximations to the optimal segmentations, the computational effort needed to produce these results is drastically reduced: From 13.5 minutes for the complete SDP, and 3.5 minutes for the complete normalized cut, to 5–6 seconds for *all* sampling-based approaches!

Finally Figure 3 depicts the results of the sampling-based segmentation methods for several real world images from the Corel dataset. For these examples, the different techniques were applied hierarchically to produce partitionings into more than two segments. In every step, we computed a binary partitioning of each segment based on 100 randomly selected pixels (0.26% of the entire image), and selected the cut giving the lowest normalized cut value.


(a)

(b)

(c)

(d)

(e)

(f)

Figure 1: **Two clustering problems.** The statistical performance is measured for different numbers of sampled points, based on 100 experiments for each value. (a,b) The optimal solutions found by both the SDP and the normalized cut formulation of the binary segmentation problem, using all points. (c,d) Mean errors. All methods give good results also for relatively small sample rates, especially for the quite simple example in (a). (e,f) Computation times. Note that especially for SDP, the computational effort is reduced strongly, so that it becomes comparable to the normalized cut sampling methods for small sample numbers.

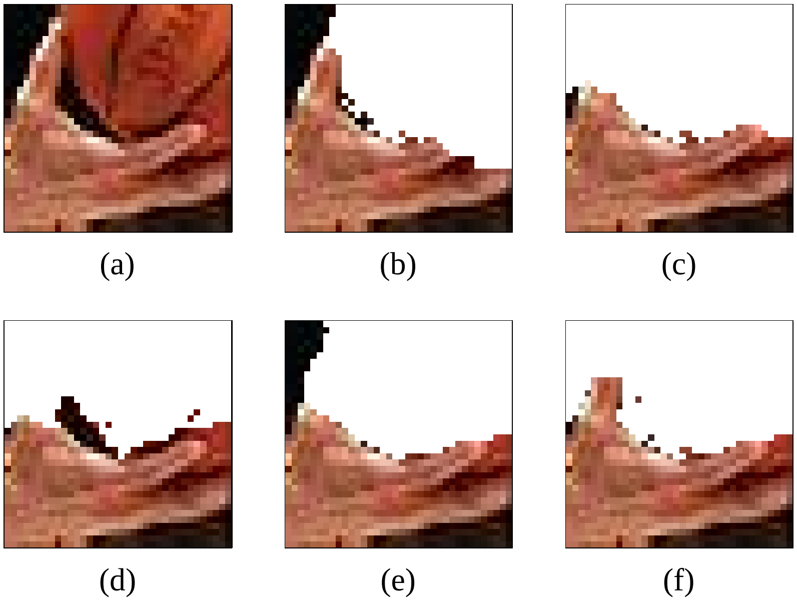


Figure 2: (a) A color image of size 36×36 pixels. 6.2% of the pixels were sampled. (b,c) Optimal segmentations obtained with normalized cut and SDP, respectively. (d,e,f) Approximate solutions calculated with the normalized cut using the probabilistic SVD method, using the Nyström extension and with SDP using the probabilistic SVD method, respectively. While maintaining a satisfying segmentation quality, the computational effort to produce these results is reduced by more than 95%.

This procedure was stopped after four steps, thus yielding a segmentation into at most five segments. Note that for problems of this size, the calculation of the binary solution becomes the most time consuming step: The vectors to be examined are by orders of magnitude larger than the solutions of the corresponding sampling-based small scale problems.

The results reveal that for both segmentation techniques, the application of the sampling-based SVD approximation method is successful: Taking into account that no effort was made to smooth the segments or to stop the partitioning process at a more adequate number of segments, the segmentations obtained are quite good and comparable to the results of the normalized cut approach based on the Nyström extension. Concerning the computational effort, just about 350 seconds for normalized cut sampling and 110 seconds for SDP sampling were needed to find the first binary partitioning of the images. The larger solution time for normalized cut sampling is due to the fact that several approximative eigenvectors were tested for good cut values, while for SDP, only a fixed number of random hyperplanes was used.

6 Conclusion

In this paper, we have shown the potential of solving unsupervised image segmentation problems by using a sampling-based SVD approximation technique. While increasing the compu-

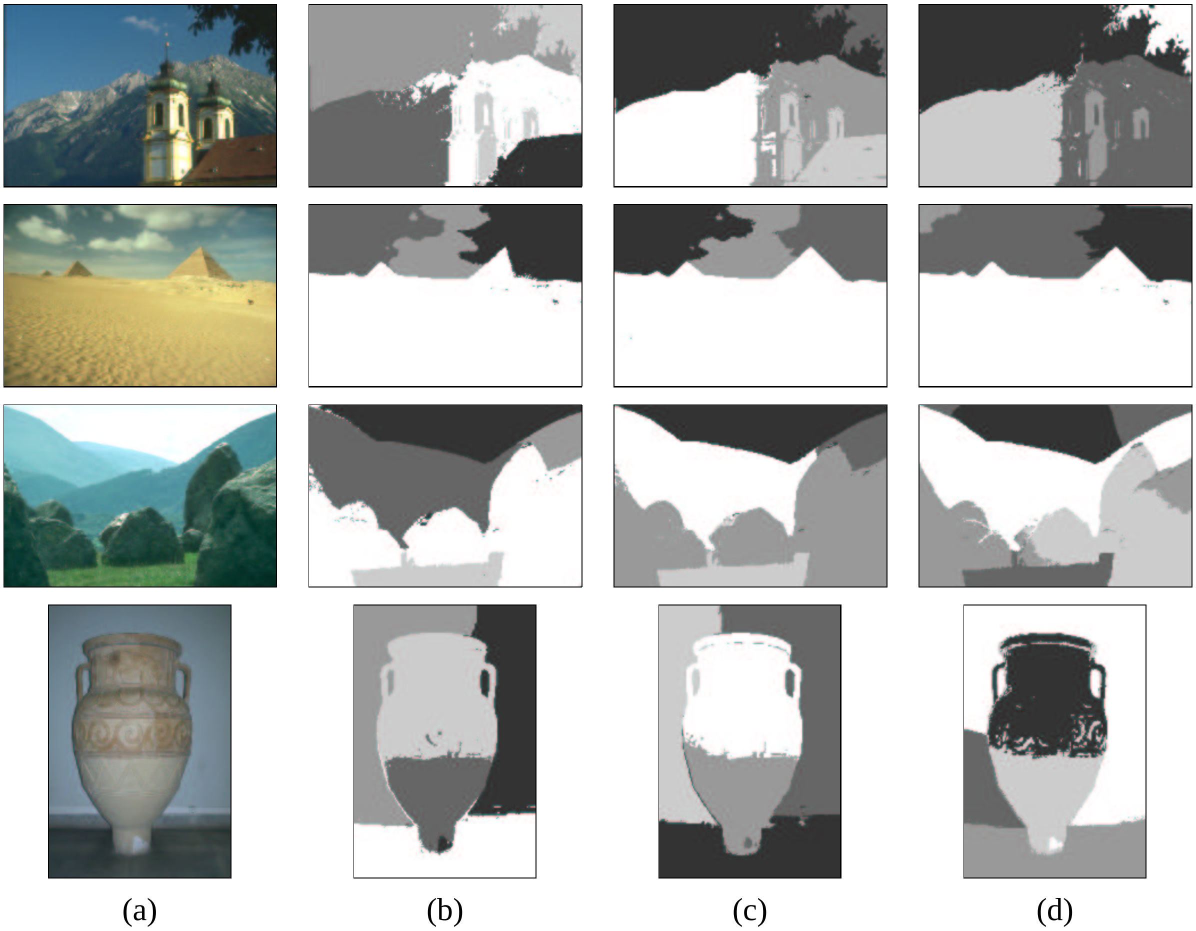


Figure 3: (a) Four color images from the Corel dataset of size 240×160 pixels. 0.26% of the pixels were sampled in the first step. (b,c,d) Partitionings calculated with the normalized cut using the probabilistic SVD method, using the Nyström extension and with SDP using the probabilistic SVD method, respectively. All approximation techniques give satisfactory results.

tational efficiency by more than 95%, a good quality of the partitionings is maintained. Especially note that after having reduced the size of the problem by sampling, the computational effort for the semidefinite programming approach — which usually is very high due to the required computation of a matrix quadratic in the number of pixels — becomes comparable to the normalized cut method.

The comparison of the probabilistic SVD method with the Nyström extension revealed their structural similarity in theory, which was approved by the results of applying both methods to solve the normalized cut problem. If one keeps in mind the computational problems that may be involved due to the calculation of an inverse matrix, it may prove valuable to also apply the Nyström method to the semidefinite programming approach in future work.

Finally, the reduced computational effort also permits to apply sampling multiple times to an image, in order to obtain possibly different segmentations from which the best one could be picked according to some previously defined criterion. In this way, unsatisfactory results due to random selection of a non-representative set of sample pixels can be averted.

Acknowledgment. This work was supported by the Deutsche Forschungsgemeinschaft (DFG; grant Schn457/3).

References

[1] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 22(8):888–905, 2000.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[3] Y. Gdalyahu, D. Weinshall, and M. Werman. Self-organization in vision: Stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(10):1053–1074, 2001.

[4] J. Keuchel, C. Schnörr, C. Schellewald, and D. Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. Patt. Anal. Mach. Intell.*, 2003. In press.

[5] B. Mohar and S. Poljak. Eigenvalues in combinatorial optimization. In R.A. Brualdi, S. Friedland, and V. Klee, editors, *Combinatorial and Graph-Theoretical Problems in Linear Algebra*, volume 50 of *IMA Vol. Math. Appl.*, pages 107–151. Springer, 1993.

[6] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *Proc. Int. Conf. Comp. Vision (ICCV'99)*, pages 975–982, 1999.

[7] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the Nyström method. In *Proc. CVPR*, volume 1, pages 231–238, 2001.

[8] C.T.H. Baker. *The Numerical Treatment of Integral Equations*. Clarendon Press, Oxford, 1977.

[9] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *Proc. 10th Symp. Discrete Algorithms*, pages 291–299. ACM-SIAM, 1999.

[10] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*, Boston, 2000. Kluwer Acad. Publ.

[11] Y. Ye. *Interior Point Algorithms: Theory and Analysis*. Wiley, 1997.

[12] S.J. Benson, Y. Ye, and X. Zhang. Mixed linear and semidefinite programming for combinatorial and quadratic optimization. *Optimiz. Methods and Software*, 11&12:515–544, 1999.

[13] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[14] A.M. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proc. on FOCS*, pages 370–378, 1998.

[15] S. Belongie, C. Fowlkes, F. Chung, and J. Malik. Spectral partitioning with indefinite kernels using the Nyström extension. In *ECCV (3)*, pages 531–542, 2002.

[16] G.H. Golub and C.F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[17] G.W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[18] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[19] V. Roth, J. Laub, M. Kawanabe, and J.M. Buhmann. Optimal cluster preserving embedding of non-metric proximity data. Technical Report IAI-TR-2002-5, University of Bonn, 2002.