Graph Nodes Clustering Based on the Commute-Time Kernel

Luh Yen[1], Francois Fouss[1], Christine Decaestecker[2], Pascal Francq[3], and Marco Saerens[1]

[1] Université catholique de Louvain, ISYS, IAG, Louvain-la-Neuve, Belgium
{luh.yen, francois.fouss, marco.saerens}@uclouvain.be
[2] Université libre de Bruxelles, Institut de Pharmacie, Bruxelles, Belgium
cdecaes@ulb.ac.be
[3] Université libre de Bruxelles, STIC, Bruxelles, Belgium
pfrancq@ulb.ac.be

Abstract. This work presents a kernel method for clustering the nodes of a weighted, undirected, graph. The algorithm is based on a two-step procedure. First, the sigmoid commute-time kernel ($\mathbf{K}_{\mathrm{CT}}$), providing a similarity measure between any couple of nodes by taking the indirect links into account, is computed from the adjacency matrix of the graph. Then, the nodes of the graph are clustered by performing a kernel k-means or fuzzy k-means on this CT kernel matrix. For this purpose, a new, simple, version of the kernel k-means and the kernel fuzzy k-means is introduced. The joint use of the CT kernel matrix and kernel clustering appears to be quite successful. Indeed, it provides good results on a document clustering problem involving the newsgroups database.

1 Introduction

This work presents a general methodology for clustering the nodes of a weighted, undirected, graph. Graph nodes clustering is an important issue that has been the subject of much recent work; see for instance [4], [5], [7], [11], [17] and [19].

On the other hand, kernel-based algorithms are characterized by two properties: they allow (i) to compute implicitly similarities in a high-dimensional space where the data are more likely to be well-separated and (ii) to compute similarities between structured objects that cannot be naturally represented by a simple set of features. In this paper we propose a new kernel matrix on a weighted, undirected, graph, which defines similarities between the nodes. These similarities take both direct and indirect links into account; they therefore take the indirect paths between the nodes into consideration. Two nodes are considered as similar if there are many short paths connecting them.

Based on this kernel matrix, nodes are clustered thanks to a kernel clustering. The kernel clustering algorithms proposed in this paper differ from existing ones ([2], [9], [10], [20], [22] and [23]) by the fact that a prototype vector is explicitly defined for each cluster. This is more natural since it allows to mimic the iterative update rules reminiscent from k-means and fuzzy k-means in the sample space,

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1037–1045, 2007.

instead of the feature space. In addition to be very similar to the original feature-based algorithms, this sample-based method can easily be extended to variable-metric or multi-prototype kernel k-means, in the same way as the original k-means and fuzzy k-means [6]. In addition to this, the resulting algorithm is very simple and natural.

The performances are evaluated on the problem of clustering newsgroups documents, and compared to the popular spherical k-means algorithm, which is especially designed for document clustering [3], as well as a classic spectral clustering method [12]. The collection of documents is viewed as a graph and the basic problem is to cluster the documents in order to eventually retrieve the newsgroups. The results indicate that the introduced algorithms perform well in comparison with the spherical k-means and the spectral clustering, with significant improvement.

The paper is organized as follows. Section 2 introduces the sigmoid commute-time kernel ($\mathbf{K}_{\mathrm{CT}}$) on a graph that will be used as similarity measure for clustering the nodes. Section 3 derives our version of the kernel k-means and kernel fuzzy k-means, while Section 4 shows the results obtained on the newsgroups database. Section 5 is the conclusion.

2 The Sigmoid Commute-Time Kernel on a Graph

Let us consider that we are given a weighted, undirected, graph, G, with symmetric weights $w_{ij} > 0$ on the edges connecting pairs of nodes i, j. The elements a_{ij} of the adjacency matrix $\mathbf{A}$ of the graph are defined in a standard way as $a_{ij} = w_{ij}$ if node i is connected to node j and 0 otherwise. Based on the adjacency matrix, the Laplacian matrix $\mathbf{L}$ of the graph is defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathbf{Diag}(a_{i.})$ is the degree matrix, with diagonal entries $d_{ii} = [\mathbf{D}]_{ii} = a_{i.} = \sum_{j=1}^{n} a_{ij}$. We suppose that the graph has a single connected component; that is, any node can be reached from any other node of the graph. In this case, $\mathbf{L}$ has rank $n - 1$, where n is the number of nodes. Moreover, it can be shown that $\mathbf{L}$ is symmetric and positive semidefinite (see for instance [8]).

The "commute time" kernel [14], [8] takes its name from the **average commute time**, $n(i, j)$, which is defined as the average number of steps a random walker, starting in node $i \neq j$, will take before entering a node j for the first time, and go back to i. Indeed, we associate a Markov chain to the graph in the following obvious manner. A state is associated to every node (n in total), and the transition probabilities are given by $p_{ij} = a_{ij}/a_{i.}$ where $a_{i.} = \sum_{j=1}^{n} a_{ij}$. One can show [14], [8] that, in this case, the average commute time can be computed thanks to $n(i, j) = V_G\,(\mathbf{e}_i - \mathbf{e}_j)^{\mathrm{T}}\mathbf{L}^{+}(\mathbf{e}_i - \mathbf{e}_j)$ where every node i of the graph is represented by a basis vector, $\mathbf{e}_i$ (the i-th column of the identity matrix $\mathbf{I}$), in the Euclidean space $\Re^n$ and $V_G = a_{..}$ is the volume of the graph. $\mathbf{L}^{+}$ is the Moore-Penrose pseudoinverse of the Laplacian matrix of the graph and is positive semidefinite. Thus, $n(i, j)$ is a **Mahalanobis distance** between the nodes of the graph and is referred to as the "commute time distance" or the

"resistance distance" because of a close analogy with the effective resistance in electrical networks [8].

One can further show that $\mathbf{L}^+$ is the matrix containing the inner products of the node vectors in the Euclidean space where these node vectors are exactly separated by commute time distances. In other words, the entries of $\mathbf{L}^+$ can be viewed as similarities between nodes and $\mathbf{L}^+$ can be considered as a kernel matrix:

$$\mathbf{K} = \mathbf{L}^+ \tag{1}$$

The **sigmoid commute time kernel** $\mathbf{K}_{\mathrm{CT}}$ is obtained by applying a sigmoid transform [15] on $\mathbf{K}$. In other words, each element of the kernel matrix is given by the formula

$$[\mathbf{K}_{\mathrm{CT}}]_{ij} = 1/(1 + \exp[a\ l_{ij}^+/\sigma]) \tag{2}$$

where $l_{ij}^+ = [\mathbf{L}^+]_{ij}$ and σ is a normalizing factor, corresponding to the standard deviation of the elements of $\mathbf{L}^+$. The parameter a will be set to a constant value determined by informal preliminary tests. The sigmoid function aims to normalize the range of the similarities in the interval $[0, 1]$ [15]. Notice, however, that the resulting matrix is not necessarily positive semi-definite so that, strictly speaking, it is not a kernel matrix.

3 Kernel k-Means and Fuzzy k-Means

We now introduce our kernel, prototype-based, version of the k-means and fuzzy k-means clustering algorithms.

3.1 Kernel k-Means

The goal is to design an iterative algorithm aiming to minimize a cost function which, in the case of a standard k-means, can be defined, in the feature space, as the total within-cluster inertia:

$$J(\mathbf{g}_1, \ldots, \mathbf{g}_m) = \sum_{k=1}^{m} \sum_{i \in C_k} ||\mathbf{x}_i - \mathbf{g}_k||^2 \tag{3}$$

where the first sum is taken on the m clusters, while the second sum is taken on the nodes i belonging to cluster k, $i \in C_k$. In Equation (3), $\mathbf{x}_i$ is the feature vector corresponding to node i, $\mathbf{g}_k$ is a prototype vector of cluster k in the **feature space** and $||\mathbf{x}_i - \mathbf{g}_k||$ is the Euclidean distance between the node vector and the cluster prototype it belongs to. The number of clusters, m, is provided a priori by the user.

We denote by $\mathbf{X}$ the data matrix containing the transposed node vectors as rows, that is, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^{\mathrm{T}}$. Let us now define the following change of parameter:

$$\mathbf{g}_k \rightarrow \mathbf{X}^{\mathrm{T}} \mathbf{h}_k \tag{4}$$

corresponding to the "kernel trick" (see [16]). It aims to express the prototype vectors, $\mathbf{g}_k$, as a linear combination of the node vectors, $\mathbf{x}_i$ (the columns of

$\mathbf{X}^{\mathrm{T}}$). The $\mathbf{h}_k$ will be called the prototype vectors in the n-dimensional **sample space**. Now, recompute the within-class inertia in terms of the $\mathbf{h}_k$ and the inner products:

$$\begin{aligned} J(\mathbf{h}_1,\ldots,\mathbf{h}_m) &= \sum_{k=1}^{m}\sum_{i\in C_k}(\mathbf{x}_i-\mathbf{g}_k)^{\mathrm{T}}(\mathbf{x}_i-\mathbf{g}_k) \\ &= \sum_{k=1}^{m}\sum_{i\in C_k}(k_{ii}-2\mathbf{k}_i^{\mathrm{T}}\mathbf{h}_k+\mathbf{h}_k^{\mathrm{T}}\mathbf{K}\mathbf{h}_k) \\ &= \sum_{k=1}^{m}\sum_{i\in C_k}(\mathbf{e}_i-\mathbf{h}_k)^{\mathrm{T}}\mathbf{K}(\mathbf{e}_i-\mathbf{h}_k) \end{aligned} \tag{5}$$

where $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathrm{T}}$, $k_{ii}=[\mathbf{K}]_{ii}=\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i$, $\mathbf{k}_i=\mathbf{X}\mathbf{x}_i=\mathbf{col}_i(\mathbf{K})$.

The k-means iteratively minimizes J by proceeding in two steps, (1) re-allocation of the node vectors while keeping the prototype vectors fixed, and (2) re-computation of the prototype vectors, $\mathbf{h}_k$, while maintaining the cluster labels of the nodes fixed. Clearly, the re-allocation step minimizing J is

$$l_i = \arg\min_k \left\{(\mathbf{e}_i-\mathbf{h}_k)^{\mathrm{T}}\mathbf{K}(\mathbf{e}_i-\mathbf{h}_k)\right\} \tag{6}$$

where l_i contains the cluster label of node i.

For the computation of the prototype vector, by taking the gradient of J with respect to $\mathbf{h}_k$ and setting the result equal to $\mathbf{0}$, we obtain $\mathbf{K}\mathbf{h}_k=\frac{1}{n_k}\sum_{i\in C_k}\mathbf{k}_i=\mathbf{K}\frac{1}{n_k}\sum_{i\in C_k}\mathbf{e}_i$ where n_k is the number of nodes belonging to cluster k. By looking carefully, we immediately observe from the left-hand side of the equation that $\mathbf{K}\mathbf{h}_k$ is a linear combination of the $\mathbf{k}_i$, while the right-hand side is also a linear combination of the $\mathbf{k}_i$. Therefore, one solution to this linear system of equations is simply the following:

$$\mathbf{h}_k = \frac{1}{n_k}\sum_{i\in C_k}\mathbf{e}_i \tag{7}$$

In other words, $\mathbf{h}_k$ contains $1/n_k$ if $i\in C_k$ and 0 otherwise. This two-step procedure (equations (6) and (7))is iterated until convergence.

3.2 Kernel Fuzzy k-Means

We now apply the same procedure for deriving a kernel fuzzy k-means. This time, the cost function is

$$J(\mathbf{g}_1,\ldots,\mathbf{g}_m;\mathbf{U})=\sum_{k=1}^{m}\sum_{i=1}^{n}u_{ik}||\mathbf{x}_i-\mathbf{g}_k||^2 \text{ with } \sum_{k=1}^{m}u_{ik}^{1/q}=1 \text{ for all } i \tag{8}$$

where the u_{ik} define the degree of membership of node i to cluster C_k. The parameter $q>1$ is controlling the degree of fuzzyness of the membership functions.

As for the kernel k-means, we perform the change of parameter (4), leading to the following update formula for the membership function.

$$u_{ik} = \left[\frac{\left((\mathbf{e}_i - \mathbf{h}_k)^{\mathrm{T}} \mathbf{K} (\mathbf{e}_i - \mathbf{h}_k)\right)^{-1/(q-1)}}{\sum_{l=1}^{m} \left((\mathbf{e}_i - \mathbf{h}_l)^{\mathrm{T}} \mathbf{K} (\mathbf{e}_i - \mathbf{h}_l)\right)^{-1/(q-1)}} \right]^{q} \tag{9}$$

and the re-computation of the prototype vectors is simply,

$$h_{ki} = [\mathbf{h}_k]_i = \frac{u_{ik}}{\sum_{j=1}^{n} u_{jk}} \tag{10}$$

4 Experiments

4.1 Data Set

In order to test the performances of the $\mathbf{K}_{\mathrm{CT}}$ k-means and the $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means, both algorithms will be assessed on a real data set and compared to classical clustering algorithms. The idea is to assess both algorithms on graph data set where only the information on relation between nodes is given. The tested graphs are extracted from the newsgroups data set (Available from http://people.csail.mit.edu/jrennie/20Newsgroups/); it is composed of 20,000 unstructured documents, taken from 20 discussion groups (newsgroups) of the Usernet diffusion list. As the data set is composed of documents, the clustering performances of both methods will be compared to the spherical k-means [3], which is a reference in text mining; and to Ng's spectral clustering [12], which presents some similarities with our approach.

For our experiment, 9 subsets including different topics are extracted from the original database, as listed in figure 1. More precisely, for each subset, 200 documents are sampled from different newsgroups. Thus, the three first subsets (G-2cl-A, G-2cl-B, G-2cl-C) contain 400 documents sampled from two newsgroups topics, the next three subsets (G-3cl-A, G-3cl-B, G-3cl-C) contain 600 documents sampled from three topics and the last three subsets (G-5cl-A, G-5cl-B, G-5cl-C) contain 1000 documents sampled from five topics. The selected topics can be related such as politics/mideast and politics/guns in subset G-5cl-A. Both the classification rate (obtained by comparing the clustering to the real newsgroups and performing an optimal assignment) and the adjusted Rand index (with values scaled in $[0, 1]$) will be reported.

4.2 Graph Definition

The newsgroups data set can be seen as a large bipartite graph between documents and terms. Each document node is connected to terms nodes contained

G-2cl-A	politics/general, sport/baseball
G-2cl-B	computer/graphics, motor/motorcycles
G-2cl-C	space/general, politics/mideast
G-3cl-A	sport/baseball, space/general, politics/mideast
G-3cl-B	computer/windows, motor/autos, religion/general
G-3cl-C	sport/hockey, religion/atheism, medicine/general
G-5cl-A	computer/windowsx, cryptography/general, politics/mideast, politics/guns, religion/christian
G-5cl-B	computer/graphics, computer/pchardware, motor/autos, religion/atheism, politics/mideast
G-5cl-C	computer/machardware, sport/hockey, medicine/general, religion/general, forsale/general

Fig. 1. Document subsets used in our experiments. Nine subsets are selected from the *Newsgroups* data set, with 2, 3 or 5 topics. For each subset, 200 documents are randomly selected from each topic.

in the document, each edge being weighted by the *tf.idf* factor [18]. After some preprocessing steps (see below) aiming to reduce the number of terms, a graph involving only documents is computed from this bipartite graph in the following way: the link between two documents is given by the sum of all document-term-document paths connecting them and passing through the terms they have in common. In other words, if $\mathbf{W}$ represents the term-document matrix containing the *tf.idf* factors, the adjacency matrix of the resulting document-document graph is provided by $\mathbf{A} = \mathbf{W}^{\mathrm{T}}\mathbf{W}$.

4.3 Preprocessing Steps

In order to reduce the high dimensionality of the feature space (terms), the following standard preprocessing steps are performed on the data set before the clustering experiment.

1. Stopwords without useful information are eliminated.
2. Porter's stemming algorithm [13] is applied so that each word is reduced to its "root ".
3. Words that occur too few times (< 3) or in too few documents (< 2) are considered as no content-bearing and are eliminated.
4. The mutual information between terms and documents is computed. For a word y, the mutual information with the documents of the data set [21] is given by
$$I(y) = \sum_x \log p(x, y)/p(x)p(y), \tag{11}$$
where x represents the documents of the data set. Words with a small value of mutual information (fixed at 20% of $I(y)$'s median) are eliminated.
5. The term-document matrix $\mathbf{W}$ is constructed with the remaining words and documents. Element $[\mathbf{W}]_{ij}$ of the matrix contains the value of *tf.idf* factor between the term i and the document j.
6. Each row of the term-document matrix $\mathbf{W}$ is normalized to 1.

Finally, the adjacency matrix of the documents graph $\mathbf{A}$ is given by the document-document matrix $\mathbf{W}^{\mathrm{T}}\mathbf{W}$. Based on $\mathbf{A}$, $\mathbf{K}_{\mathrm{CT}}$ is computed by Equation (2). For example, the subset G-2cl-A is composed of 400 documents, and 2898 terms with stopwords already eliminated. After preprocessing, only 1490 terms are kept. Thus, the clustering algorithm will be run on a 400×400 document-document matrix, instead of a 1490×400 term-document matrix for a standard feature-based algorithm.

4.4 Experimental Settings

Suppose we have a graph of n nodes to be partitioned into m clusters. First, the prototype vectors $\mathbf{h}_i$ $(i = 1, ..., m)$ are initialized by randomly selecting m columns of the identity matrix $\mathbf{I}$. Then, each algorithm is run 30 times (30 runs), and the classification rate as well as the adjusted Rand index, averaged on the 30 runs, are computed. The $\mathbf{K}_{\mathrm{CT}}$ k-means, $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means and Ng's spectral clustering are run on the document-document matrix $\mathbf{A}$, while the spherical k-means is run on the term-document matrix $\mathbf{W}$ after preprocessing.

Each run consists in 50 trials: the clustering algorithm is launched 50 times and the best solution among the 50 trials, having the minimal within-class inertia, is sent back as the solution.

Two parameters need to be tuned. The first one is the parameter a for computing the sigmoid transform of the $\mathbf{K}_{\mathrm{CT}}$ (see Equation (2)). The second one is the parameter q which controls the degree of fuzzyness for the kernel fuzzy k-means (see Equation (9)). Based on preliminary informal experiment, the parameters a and q were set to 7 and 1.2 respectively, for all experiments.

4.5 Experimental Results and Discussion

The results (the classification rate as well as the adjusted Rand index, each averaged on 30 runs) of the four clustering algorithms ($\mathbf{K}_{\mathrm{CT}}$ k-means, $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means, spherical k-means and Ng's spectral clustering) on the nine document subsets are reported in Table 1.

We observe that the $\mathbf{K}_{\mathrm{CT}}$ k-means and the $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means outperform the spherical k-means on the nine subsets. Ng's spectral clustering presents good results on the 2-classes and 3-classes data sets, but degrades when the number of clusters increases. Moreover, the $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means provides slightly better results than the two other methods. This can be partly explained by the fact that the newsgroups data set is fuzzy itself, as discussed in [1]. It is hard to define clear boundaries between the different topics: a discussion within a specific newsgroup can also be related to other domains. A close examination of the data set shows that several discussions can even be out of subject or are simply empty of useful information.

Table 1. Comparison of the clustering performances (classification rate in % and adjusted Rand index with value scaled in $[0, 1]$) for the $\mathbf{K}_{\mathrm{CT}}$ k-means, $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means, spherical k-means and Ng's spectral clustering

	$\mathbf{K}_{\mathrm{CT}}$ k-means		$\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means		Sph. k-means		Ng spec. clus.	
	class. rate	adj. Rand	class. rate	adj. Rand	class. rate	adj. Rand	class. rate	adj. Rand
G-2cl-A	97.5 %	0.95	97.8 %	0.96	91.8 %	0.85	94.5 %	0.90
G-2cl-B	90.6 %	0.83	91.5 %	0.84	81.5 %	0.70	93.0 %	0.87
G-2cl-C	95.5 %	0.91	96.0 %	0.92	94.8 %	0.90	95.7 %	0.92
G-3cl-A	93.9 %	0.91	94.5 %	0.92	89.2 %	0.85	92.7 %	0.90
G-3cl-B	93.6 %	0.91	93.5 %	0.91	86.7 %	0.82	92.0 %	0.89
G-3cl-C	93.9 %	0.91	92.8 %	0.90	87.4 %	0.83	81.7 %	0.78
G-5cl-A	83.0 %	0.80	85.4 %	0.83	80.4 %	0.79	76.7 %	0.78
G-5cl-B	74.8 %	0.77	78.4 %	0.79	64.4 %	0.69	67.7 %	0.72
G-5cl-C	76.4 %	0.75	80.1 %	0.79	64.9 %	0.69	64.0 %	0.72

5 Conclusions and Further Work

We introduced a new method allowing to cluster the nodes of a weighted graph by exploiting the links between them. It is based on a recently introduced kernel on a graph, the commute-time kernel, combined with a kernel clustering. The obtained results are promising since the proposed methodology outperforms the standard spherical k-means as well as spectral clustering on a difficult graph clustering problem. Further work will be devoted to (1) additional experiments on other text databases, and (2) developing kernel versions of the Gaussian mixture, the entropy-based fuzzy clustering, Ward's hierarchical clustering, and assessing their performances.

References

1. M. W. Berry, editor. *Survey of Text Mining*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
2. I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM Press, 2004.
3. I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.
4. C. Ding and X. He. Linearized cluster assignment via spectral ordering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 30, New York, NY, USA, 2004. ACM Press.
5. P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004.
6. B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. Arnold Publishers, 2001.
7. G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Math*, 1(4):385–408, 2003.

8. F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):355–369,, 2007.
9. M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, May 2002.
10. D.-W. Kim, K. Y. Lee, D. Lee, and K. H. Lee. Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recognition*, 38(4):607–611, 2005.
11. M. Newman. Detecting community structure in networks. *The European Physical Journal B*, 38:321–330, 2004.
12. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processiong Systems*, volume 14, pages 849–856, Vancouver, Canada, 2001. MIT Press.
13. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
14. M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. *Proceedings of the 15th European Conference on Machine Learning (ECML 2004). Lecture Notes in Artificial Intelligence, Vol. 3201, Springer-Verlag, Berlin*, pages 371–383, 2004.
15. B. Scholkopf and A. Smola. *Learning with kernels*. The MIT Press, 2002.
16. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
17. S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.
18. S. Weiss, N. Indurkhya, T. Zhang, and F. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer, 2004.
19. S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, 2005.
20. Z.-D. Wu, W.-X. Xie, and J.-P. Yu. Fuzzy c-means clustering algorithm based on kernel method. In *ICCIMA '03: Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications*, page 49, Washington, DC, USA, 2003. IEEE Computer Society.
21. H. Zha, X. He, C. H. Q. Ding, M. Gu, and H. D. Simon. Bipartite graph partitioning and data clustering. In *Proc. of ACM 10th Int'l Conf. Information and Knowledge Management (CIKM 2001)*, pages 25–32, 2001.
22. D.-Q. Zhang and S.-C. Chen. Fuzzy clustering using kernel method. In *Proceedings of the 2002 International Conference on Control and Automation, 2002. ICCA*, pages 162–163, 2002.
23. D.-Q. Zhang and S.-C. Chen. A novel kernelized fuzzy c-means algorithm with application in medical image segmentation. *Artificial Intelligence in Medicine*, 32(1):37–50, 2004.