05-205

Pochet N.L.M.M., Ojeda F., De Smet F., De Bie T., Suykens J.A.K., De Moor B.L.R., ``Kernel clustering for knowledge discovery in clinical microarray data analysis", in Chapter 3 *Kernel methods in bioengineering, communications and image processing*, (Camps-Valls G., Rojo-Alvarez J.L., and Martinez-Ramon M., eds.), Idea Group Inc (Hershey, Pennsylvania), 2005, pp. 64-92., Lirias number: 181277.

Chapter III

Kernel Clustering for Knowledge Discovery in Clinical Microarray Data Analysis

Nathalie L. M. M. Pochet, Katholieke Universiteit Leuven, Belgium

Fabian Ojeda, Katholieke Universiteit Leuven, Belgium

Frank De Smet, Katholieke Universiteit Leuven, Belgium
& National Alliance of Christian Mutualities, Belgium

Tijl De Bie, Katholieke Universiteit Leuven, Belgium

Johan A. K. Suykens, Katholieke Universiteit Leuven, Belgium

Bart L. R. De Moor, Katholieke Universiteit Leuven, Belgium

Abstract

Clustering techniques like k-means and hierarchical clustering have shown to be useful when applied to microarray data for the identification of clinical classes, for example, in oncology. This chapter discusses the application of nonlinear techniques like kernel k-means and spectral clustering, which are based on kernel functions like linear and radial basis function (RBF) kernels. External validation techniques (e.g., the Rand index and the adjusted Rand index) can immediately be applied to these methods for the assessment of clustering results. Internal validation methods like the global silhouette index, the distortion score, and the Calinski-Harabasz index (F-statistic), which have been commonly used in the input space, are reformulated in this chapter for usage in a kernel-induced feature space.

Introduction

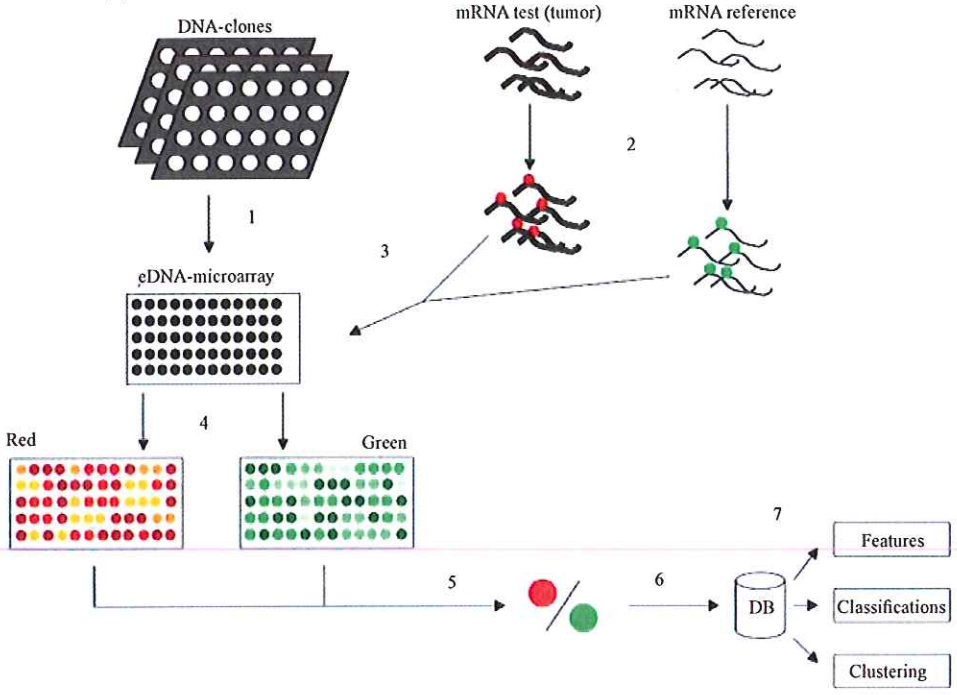
Microarrays are a recent technology that allows for determining the expression levels of thousands of genes simultaneously. One important application area of this technology is clinical oncology. Parallel measurements of these expression levels result in data vectors that contain thousands of values, which are called expression patterns. A microarray consists of a reproducible pattern of several DNA probes attached to a small solid support. Labeled cDNA, prepared from extracted mRNA, is hybridized with the complementary DNA probes attached to the microarray. The hybridizations are measured by means of a laser scanner and transformed quantitatively. Two important types of microarrays are cDNA microarrays and oligonucleotide arrays. cDNA microarrays consist of about 10,000 known cDNA (obtained after PCR amplification) that are spotted in an ordered matrix on a glass slide. Oligonucleotide arrays (or DNA chips) are constructed by the synthesis of oligonucleotides on silicium chips. Figure 1 gives a schematic overview of an experiment with the cDNA technology. Both technologies have specific characteristics that will not be discussed here. When studying, for example, tumor tissues with microarrays, the challenge mainly lies in the analysis of the experiments in order to obtain relevant clinical information. Most of the techniques that have been widely used for analyzing microarrays require some preprocessing stage such as gene selection, filtering, or dimensionality reduction, among others. These methods cannot directly deal with high-dimensional data vectors. Moreover, these are methods that are specifically designed to deal with the particular challenges posed by gene expression data and thus they do not provide a more general framework that can be easily extended to other kinds of data. For this purpose, methods and algorithms capable of handling high-dimensional data vectors and that are capable of working under a minimal set of assumptions are required. The chapter by Jean-Philippe Vert in this book focuses on the classification of high-dimensional data, while this chapter elaborates on the cluster analysis of these high-dimensional data.

Clustering techniques are generally applied to microarray data for the identification of clinical classes, which could allow for refining clinical management. Cluster analysis of entire microarray experiments (expression patterns from patients or tissues) allows for the discovery of possibly unknown diagnostic categories without knowing the properties of these classes in advance. These clusters could form the basis of new diagnostic schemes in which the different categories contain patients with less clinical variability.

Clustering microarray experiments have already shown to be useful in a large number of cancer studies. Alon et al. (1999), for example, separated cancerous colon tissues from noncancerous colon tissues by applying two-way clustering. The distinction between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) has been rediscovered by using self-organizing maps (SOM) by Golub et al. (1999). By using hierarchical clustering, van 't Veer et al. (2002) were able to distinguish between the presence (poor prognosis) and the absence (good prognosis) of distant subclinical metastases in breast cancer patients where the histopathological examination did not show tumor cells in local lymph nodes at diagnosis (lymph node negative).

For this purpose, methods such as the classical k-means clustering and hierarchical clustering are commonly used (Bolshakova, Azuaje, & Cunningham, 2005; Handl, Knowles, & Kell, 2005). These methods are based on simple distance or similarity measures (e.g., the

Figure 1. Schematic overview of an experiment with a cDNA microarray: (1) Spotting of the presynthesized DNA probes (derived from the genes to be studied) on the glass slide. These probes are the purified products from PCR amplification of the associated DNA clones. (2) Labeling (via reverse transcriptase) of the total mRNA of the test sample (tumor in red) and reference sample (green). (3) Pooling of the two samples and hybridization. (4) Readout of the red and green intensities separately (measure for the hybridization by the test and reference sample) in each probe. (5) Calculation of the relative expression levels (intensity in the red channel and intensity in the green channel). (6) Storage of results in a database. (7) Data mining.



Euclidean distance). However, only linear relationships in the data can be discovered using these techniques. Recently, methods have emerged for clustering data in which the clusters are not linearly separable. Two important methods are kernel *k*-means clustering (Dhillon, Guan, & Kulis, 2004a, 2004b; Zhang & Rudnicky, 2002) and the related spectral clustering (Cristianini, Shawe-Taylor, & Kandola, 2002; Ng et al., 2001). Introducing these techniques in microarray data analysis allows for dealing with both high-dimensional data and nonlinear relationships in the data.

Validation techniques are used to assess and compare the performance of different clustering methods. These methods can also be employed for tuning the cluster settings, for example, optimizing the number of clusters or tuning the kernel parameters. A recent review of Handl et al. (2005) presents the state of the art in cluster validation on high-dimensional data, among others, on microarray data, referring to some previous important manuscripts in the field (Bolshakova & Azuaje, 2003; Halkidi, Batistakis, & Vazirgiannis, 2001). Two main kinds of validation techniques are internal and external validation. Internal validation

assesses the quality of a clustering result based on statistical properties, for example, assessing the compactness of a cluster or maximizing the intercluster distances while minimizing the intracluster distances. External validation reflects the level of agreement of a clustering result with an external partition, for example, existing diagnostic classes generally used by experts in clinical practice. The global silhouette index, the distortion score, and the Calinski-Harabasz index (F-statistic) are commonly used for internal validation, and the Rand index and adjusted Rand index for external validation.

This chapter describes classical *k*-means, kernel *k*-means, and spectral clustering algorithms and discusses their advantages and disadvantages in the context of clinical microarray data analysis. Since classical *k*-means clustering cannot handle high-dimensional microarray experiments for computational reasons, principal component analysis (PCA) is used as a preceding dimensionality-reduction step. Kernel *k*-means and spectral clustering are capable of directly handling the high-dimensional microarray experiments since they make use of the kernel trick, which allows them to work implicitly in the feature space. Several internal and external cluster-validation criteria commonly used in the input data space are described and extended for usage in the feature space. The advantages of nonlinear clustering techniques in case of clinical microarray data analysis are further demonstrated by means of the clustering results on several microarray data sets related to cancer.

Preprocessing

This chapter uses standardization as a preceding preprocessing step for all clustering methods. However, classical *k*-means clustering should not be directly applied to the high-dimensional microarray data as such. In all practical cases, the number of genes (the dimensionality) is much larger than the number of arrays (the data points) such that only a small subspace of the data space is actually spanned by the data. Therefore, in case of classical *k*-means, standardization is followed by principal component analysis to obtain a representation of the data with a reduced dimensionality (without any selection of principal components). In this section, we describe these unsupervised preprocessing steps, as well as filtering, which is also commonly used for that purpose.

Filtering

A set of microarray experiments, generating gene expression profiles (measurements of a single gene under several conditions), frequently contains a considerable number of genes that do not really contribute to the clinical process that is being studied. The expression values of these profiles often show little variation over the different experiments (they are called "constitutive" with respect to the clinical process studied). Moreover, these constitutive genes will have seemingly random and meaningless profiles after standardization (division by a small standard deviation resulting in noise inflation), which is a very common preprocessing step. Another problem with microarray data sets is the fact that these regularly contain highly unreliable expression profiles with a considerable number of missing values. Due

to their number, replacing these missing values in these expression profiles is not possible within the desired degree of accuracy.

If these data sets were passed to the clustering algorithms as such, the quality of the clustering results could significantly degrade. A simple solution (that can also be used in combination with other preprocessing steps) is to remove at least a fraction of the undesired genes from the data. This procedure is in general called filtering (Eisen, Spellman, Brown, & Botstein, 1998). Filtering involves removing gene expression profiles from the data set that do not satisfy one or possibly more criteria. Commonly used criteria include a minimum threshold for the standard deviation of the expression values in a profile (removal of constitutive genes) and a threshold on the maximum percentage of missing values. Another similar method for filtering takes a fixed number or fraction of genes best satisfying one criterion (like the criteria stated above).

Standardization or Rescaling

Biologists are mainly interested in the relative behavior instead of the absolute behavior of genes. Genes that are up- and down-regulated together should have the same weights in subsequent algorithms. Applying standardization or rescaling (sometimes also called normalization) to the gene expression profiles can largely achieve this (Quackenbush, 2001).Consider a gene expression profile, denoted by the column vector $\mathbf{g} = [g^1, g^2, ..., g^j, ..., g^\ell]$, measured for ℓ experiments. Rescaling is commonly done by replacing every expression level g^j in g by:

$$\frac{g^j - \mu}{\hat{\sigma}},$$

where μ is the average expression level of the gene expression profile and is given by

$$\mu = \frac{\sum_{j=1}^{\ell} g^j}{\ell},$$

and $\hat{\sigma}$ is the standard deviation given by

$$\hat{\sigma} = \sqrt{\frac{1}{\ell - 1}\sum_{j=1}^{\ell}(g^j - \mu)^2}.$$

This is repeated for every gene expression profile in the data set and results in a collection of expression profiles all having an average of zero and standard deviation of one (i.e., the absolute differences in expression behavior have been largely removed). The division by the standard deviation is sometimes omitted (rescaling is then called mean centering).

Principal Component Analysis

PCA looks for linear combinations of gene expression levels in order to obtain a maximal variance over a set of patients. In fact, those combinations are most informative for this set of patients and are called the principal components. One can either use all principal components or select only a subset for usage in subsequent analyses.

One formulation (Joliffe, 1986) to characterize PCA problems is to consider a given set of centered (zero mean) input data $\{\mathbf{x}_j\}_{j=1}^{\ell}$ as a cloud of points for which one tries to find projected variables $\mathbf{w}^T\mathbf{x}$ with maximal variance. This means:

$$\max_{\mathbf{w}} \ \mathrm{Var}(\mathbf{w}^T\mathbf{x}) = \mathbf{w}^T\mathbf{C}\mathbf{w},$$

where the covariance matrix $\mathbf{C}$ is estimated as

$$\mathbf{C} \cong \frac{1}{\ell - 1}\sum_{j=1}^{\ell}\mathbf{x}_j\mathbf{x}_j^T.$$

One optimizes this objective function under the constraint that $\mathbf{w}^T\mathbf{w} = 1$. Solving the constrained optimization problem gives the eigenvalue problem:

$$\mathbf{C}\mathbf{w} = \lambda\mathbf{w}.$$

The matrix $\mathbf{C}$ is symmetric and positive semidefinite. The eigenvector $\mathbf{w}$ corresponding to the largest eigenvalue determines the projected variable having maximal variance.

Kernel versions of principal component analysis and canonical correlation analysis among others have been formulated by Schölkopf, Smola, and Müller (1998). Primal-dual formulations of these methods are introduced by Suykens, Van Gestel, De Brabanter, De Moor, and Vandewalle (2002). This formulation for kernel principal component analysis has already extensively been tested as a dimensionality-reduction technique on microarray data by Pochet, De Smet, Suykens, and De Moor (2004).

Classical Clustering Methods

In a recent review, Handl et al. (2005) state that although there have recently been numerous advances in the development of improved clustering techniques for (biological and clinical) microarray data analysis (e.g., biclustering techniques [Madeira & Oliveira, 2004; Sheng, Moreau, & De Moor, 2003], adaptive quality-based clustering [De Smet, Mathys, Marchal, Thijs, De Moor, & Moreau, 2002], and gene shaving [Hastie et al., 2000]), traditional clustering techniques such as *k*-means (Rosen et al., 2005; Tavazoie, Hughes, Campbell, Cho, & Church, 1999) and hierarchical clustering algorithms (Eisen et al., 1998) remain

the predominant methods. According to this review, this fact is arguably more owing to their conceptual simplicity and their wide availability in standard software packages than to their intrinsic merits. In this context, this chapter focuses on a class of linear and nonlinear clustering techniques based on the traditional *k*-means clustering.

k-Means

The *k*-means clustering algorithm aims at partitioning the data set, consisting of ℓ expression patterns $\{\mathbf{x}_1, ..., \mathbf{x}_\ell\}$ in an n-dimensional space, into k disjoint clusters $\{C_i\}_{i=1}^k$, such that the expression patterns in each cluster are more similar to each other than to the expression patterns in other clusters (Dubes & Jain, 1988). The centers or centroids (i.e., prototypes) of all clusters $\mathbf{m}_1, ..., \mathbf{m}_k$ are returned as representatives of the data set, together with the cluster assignments of all expression patterns. The general objective of *k*-means is to obtain a partition that minimizes the mean squared error for a fixed number of clusters, where the mean squared error is the sum of the Euclidean distances between each expression pattern and its cluster center.

Suppose a set of expression patterns $\mathbf{x}_j, j = 1,..., \ell$. The objective function, that is, the mean squared error criterion, is then defined as:

$$\text{se} = \sum_{i=1}^{k}\sum_{j=1}^{\ell} z_{C_i,\mathbf{x}_j} \left\|\mathbf{x}_j - \mathbf{m}_i\right\|^2,$$

where $z_{C_i,\mathbf{x}_j}$ is an indicator function defined as

$$z_{C_i,\mathbf{x}_j} = \begin{cases} 1 & \text{if } \mathbf{x}_j \in C_i \\ 0 & \text{otherwise} \end{cases}$$

with

$$\sum_{i=1}^{k} z_{C_i,\mathbf{x}_j} = 1 \qquad \forall j,$$

and m_i is the center of cluster C_i defined as

$$\mathbf{m}_i = \frac{1}{|C_i|}\sum_{j=1}^{\ell} z_{C_i\mathbf{x}_j}\mathbf{x}_j,$$

where $|C_i|$ is the cardinality (number of elements) of the set C_i. The Euclidean distance is often used as dissimilarity function $D(\mathbf{x}_j, \mathbf{m}_i)$ in the indicator function. The iterative *k*-means clustering algorithm first proposed by MacQueen (1967) optimizes this nonconvex objective function as follows.

k-Means Clustering Algorithm

1. Select k initial centroids $\mathbf{m}_1,...,\mathbf{m}_k$.
2. Assign each expression pattern $\mathbf{x}_j, 1 \le j \le \ell$, to cluster C_i with the closest centroid m_i based on the indicator function:

$$z_{C_i,\mathbf{x}_j} = \begin{cases} 1 & D(\mathbf{x}_j,\mathbf{m}_i) < D(\mathbf{x}_j,\mathbf{m}_h), \forall h \neq i, i, h = 1,...,k \\ 0 & \text{otherwise} \end{cases}$$

3. Calculate the new centers m_i of all clusters C_i as:

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{j=1}^{\ell} z_{C_i,\mathbf{x}_j} \mathbf{x}_j.$$

4. Repeat Steps 2 and 3 until convergence (no change).
5. Return mi, $1 \le i \le k$.

This algorithm can easily be implemented and works very well for compact and hyperspherically shaped clusters. Although convergence is always reached, k-means does not necessarily find the most optimal clustering (i.e., the global minimum for the objective function). The result of the algorithm is highly dependent on the number of clusters k and the initial selection of the k cluster centroids. Cluster-validation criteria are required in order to choose the optimal settings for k and the initialization. Finally, remember that one disadvantage of this classical k-means algorithm is that preprocessing is required in order to allow clustering.

Kernel Clustering Methods

Kernel clustering methods have already shown to be useful in text-mining applications (De Bie, Cristianini, & Rosipal, 2004; Dhillon et al., 2004) and image data analysis (Zhang & Rudnicky, 2002), among others. For example, Qin, Lewis, and Noble (2003) proposed a kernel hierarchical clustering algorithm on microarray data for identifying groups of genes that share similar expression profiles. Support vector clustering (Ben-Hur, Horn, Siegelmann, & Vapnik, 2001) is another clustering method based on the approach of support vector machines. These kernel clustering methods have recently emerged for clustering data in which the clusters are not linearly separable in order to find nonlinear relationships in the data. Moreover, these techniques allow for dealing with high-dimensional data, which makes it specifically interesting for application on microarray data. In this section, we focus on kernel k-means and spectral clustering.

Kernel *k*-Means

Kernel *k*-means clustering is an extension of the linear *k*-means clustering algorithm in order to find nonlinear structures in the data. Consider a nonlinear mapping $\phi(\cdot)$ from the input space to the feature space. No explicit construction of the nonlinear mapping $\phi(\cdot)$ is required since in this feature space inner products can easily be computed by using the kernel trick $\kappa(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T\phi(\mathbf{y})$. Mercer's condition (Vapnik, 1998) guaranties that each kernel function $\kappa(\mathbf{x}, \mathbf{y})$, that is, a positive semidefinite symmetric function, corresponds to an inner product in the feature space. This allows for the construction of an $\ell \times \ell$ symmetric and positive definite kernel matrix K holding all pairwise inner products of the input data $\mathbf{K}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j), \forall i, j = 1,..., \ell$. This kernel trick can be applied to each algorithm that can be expressed in terms of inner products. The kernel functions used in this chapter are thelinear kernel, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_j^T\mathbf{x}_i$, and the RBF kernel:

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2 / \sigma^2).$$

The polynomial kernel of degree d, $\kappa(\mathbf{x}_i, \mathbf{x}_j) = (\tau + \mathbf{x}_j^T\mathbf{x}_i)^d$, with $\tau > 0$, is another commonly used kernel function.

The objective function of kernel *k*-means clustering is exactly the same as the objective function of the classical *k*-means clustering stated earlier except for the fact that it is now rewritten in terms of inner products that can be replaced by a kernel function $\kappa(\mathbf{x}, \mathbf{y})$(Dhillon et al., 2004; Zhang & Rudnicky, 2002). By introducing the feature map $\phi(\cdot)$, the mean squared error function can be expressed in the feature space by:

$$\mathrm{se}^{\phi} = \sum_{i=1}^{k}\sum_{j=1}^{\ell} z_{C_i,\mathbf{x}_j} \left\|\phi(\mathbf{x}_j) - \mathbf{m}_i^{\phi}\right\|^2$$

with $\mathbf{m}_i^{\phi}$, the cluster center of cluster C_i, defined by

$$\mathbf{m}_i^{\phi} = \frac{1}{|C_i|}\sum_{j=1}^{\ell} z_{C_i,\mathbf{x}_j}\phi(\mathbf{x}_j)$$

The Euclidean distance between $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ can be written as:

$$\begin{aligned} D^2(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) &= \left\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\right\|^2 \\ &= \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_i) - 2\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) + \phi(\mathbf{x}_j)^T\phi(\mathbf{x}_j) \\ &= \kappa(\mathbf{x}_i, \mathbf{x}_i) - 2\kappa(\mathbf{x}_i, \mathbf{x}_j) + \kappa(\mathbf{x}_j, \mathbf{x}_j) \end{aligned}$$

The computation of distances in this feature space can then be carried out by:

$$D^2(\phi(\mathbf{x}_j),\mathbf{m}_i^{\phi}) = \left\|\phi(\mathbf{x}_j) - \mathbf{m}_i^{\phi}\right\|^2$$
$$= \left\|\phi(\mathbf{x}_j) - \frac{1}{|C_i|}\sum_{l=1}^{\ell} z_{C_i,\mathbf{x}_l}\phi(\mathbf{x}_l)\right\|^2$$
$$= \phi(\mathbf{x}_j)^T\phi(\mathbf{x}_j) - \frac{2}{|C_i|}\sum_{l=1}^{\ell} z_{C_i,\mathbf{x}_l}\phi(\mathbf{x}_j)^T\phi(\mathbf{x}_l) + \frac{1}{|C_i|^2}\sum_{l=1}^{\ell}\sum_{p=1}^{\ell} z_{C_i,\mathbf{x}_l} z_{C_i,\mathbf{x}_p}\phi(\mathbf{x}_l)'\phi(\mathbf{x}_p)$$

Application of the kernel trick results in:

$$D^2(\phi(\mathbf{x}_j),\mathbf{m}_i^{\phi}) = \mathbf{K}_{jj} + f(C_i,\mathbf{x}_j) + g(C_i),$$

with

$$f(C_i,\mathbf{x}_j) = -\frac{2}{|C_i|}\sum_{l=1}^{\ell} z_{C_i,\mathbf{x}_l}\mathbf{K}_{jl}$$
$$g(C_i) = \frac{1}{|C_i|^2}\sum_{l=1}^{\ell}\sum_{p=1}^{\ell} z_{C_i,\mathbf{x}_l} z_{C_i,\mathbf{x}_p}\mathbf{K}_{lp}.$$

This gives the following formulation of the mean squared error criterion in the feature space:

$$\mathrm{se}^{\phi} = \sum_{i=1}^{k}\sum_{j=1}^{\ell} z_{C_i,\mathbf{x}_j}(\mathbf{K}_{jj} + f(C_i,\mathbf{x}_j) + g(C_i)).$$

The kernel-based *k*-means algorithm solving the nonconvex optimization problem is then as follows.

Kernel *k*-Means Clustering Algorithm

1. Assign an initial clustering value to each sample, $z_{C_i,\mathbf{x}_j}, 1 \le i \le k, 1 \le j \le \ell$, forming k initial clusters $C_1, \ldots, C_k$.
2. For each cluster C_i, compute $|C_i|$ and $g(C_i)$.
3. For each training sample $\mathbf{x}_j$ and cluster C_i compute $f(C_i, \mathbf{x}_j)$.
4. Assign $\mathbf{x}_j$ to the closest cluster by computing the value of the indicator function

$$z_{C_i,\mathbf{x}_j} = \begin{cases} 1 & f(C_i,\mathbf{x}_j) + g(C_i) < f(C_h,\mathbf{x}_j) + g(C_h), \forall h \neq i, i, h = 1,\ldots,k \\ 0 & \text{otherwise} \end{cases}.$$

5. Repeat Steps 2, 3, and 4 until convergence is reached.
6. For each cluster C_i select the sample that is closest to the center as the representative centroid of cluster C_i by computing:

$$\mathbf{m}_i = \min_{\mathbf{x}_j \text{ s.t. } z_{C_i,\mathbf{x}_j}=1} D(\phi(\mathbf{x}_j), \mathbf{m}_i^{\phi}),\ 1 \leq i \leq k.$$

Note also that in this algorithm, the factor $\mathbf{K}_{jj}$ is ignored because it does not contribute to determine the closest cluster. Remember that the term $g(C_i)$ needs to be computed only once for each cluster in each iteration, while the term $f(C_i, \mathbf{x}_j)$ is calculated once per data point.

The objective function of the kernel k-means algorithm (see distortion score in the feature space) monotonically decreases in each iteration, which also holds for the classical k-means algorithm. The general structure of the traditional algorithm is thus preserved in its nonlinear version. Nevertheless, there are two main differences between both algorithms, namely, the nonlinear mapping via the kernel trick and the lack of an explicit centroid in the feature space. The mapping from the feature space back to the input space is called the pre-image problem and is nontrivial. Typically, the exact pre-image does not exist and can therefore only be approximated, which is typically considered with respect to kernel principal component analysis (Schölkopf et al., 1998). In this algorithm, a pseudocentroid is calculated instead. However, there exist iterative nonlinear optimization methods that attempt to solve this problem (Mika, Schölkopf, Smola, Müller, Scholz, & Rätsch, 1999).

This algorithm, unfortunately, is prone to local minima since the optimization problem is not convex. Considerable effort has been devoted to finding good initial guesses or inserting additional constraints in order to limit the effect of this fact on the quality of the solution obtained. The spectral clustering algorithm is a relaxation of this problem for which it is possible to find the global solution.

Spectral Clustering

Spectral clustering techniques have emerged as promising unsupervised learning methods to group data points that are similar. These methods have been successfully applied to machine learning, data analysis, image processing, pattern recognition, and very large-scale integration (VLSI) design. These methods can be regarded as relaxations of graph-cut problems on a fully connected graph. In this graph, each node represents a data point, and the edges between the data points are assigned weights that are equal to the affinities. Clustering then corresponds to partitioning the nodes in the graph into groups. Such a division of the graph nodes in two disjoint sets is called a graph cut.

In order to achieve a good clustering, one can see that it is undesirable to separate two nodes into different clusters if they are connected by an edge with a large weight (meaning that they have a large affinity). To cast this into an optimization problem, several graph-cut cost functions for clustering have been proposed in the literature, among which are the cut cost, the average cut cost, and the normalized cut cost (Shi & Malik, 2000). The cut cost is immediately computationally tractable (Blum & Chawla, 2001), but it often leads to degenerate results (where all but one of the clusters is trivially small; see Figure 2 (right) and Joachims,

2003, for an instructive artificially constructed example). This problem can largely be solved by using the average or normalized cut-cost functions, of which the average cut cost seems to be more vulnerable to outliers (distant samples, meaning that they have low affinity to all other points). Unfortunately, both optimizing the average and normalized cut costs are NP-complete problems. To get around this, spectral relaxations of these optimization problems have been proposed (Cristianini et al., 2002; Ng, Jordan, & Weiss, 2002; Shi & Malik, 2000). These spectral relaxations are known as spectral clustering algorithms.

Given an undirected graph $G = (V, E)$ where V is the set of ℓ nodes and E is the set of edges, the problem of graph partitioning consists of separating the graph into two sets A and B by eliminating edges connecting the two sets. The sets should be disjoint such that $A \cup B = V$ and $A \cap B = \varnothing$. The total weight of the edges that have to be eliminated is called the cut

$$cut(\mathcal{A},\mathcal{B}) = \sum_{a\in\mathcal{A},b\in\mathcal{B}} w(a,b) = \sum_{i,j} w(i,j)(q_i - q_j)^2,$$

where $w(a,b)$ is the associating weight between nodes a and b, and q_i is a cluster membership indicator of the form:

$$q_i = \begin{cases} 1, & \text{if } i \in A \\ -1, & \text{if } i \in B. \end{cases}$$

Minimizing the cut cost is equivalent to the following problem:

$$\min_{\mathbf{q}} \mathbf{q}^T(\mathbf{D}-\mathbf{W})\mathbf{q}$$

such that $\mathbf{q} \in \{-1,1\}^{\ell}$,

where $\mathbf{D}$ is an $\ell\times\ell$ diagonal matrix with

$$d_i = \sum_j w(i,j)$$

on its diagonal, that is, the total connection from node i to all other nodes, and $\mathbf{W}$ is an $\ell\times\ell$ symmetric with ijth entry equal to $w(i,j)$. This problem, however, is NP hard due to the constraint on $\mathbf{q}$. A suboptimal solution can be found by relaxing the constraint and allowing real values for $\mathbf{q}$. The solution to the relaxed problem with constraint $\tilde{\mathbf{q}}^T\tilde{\mathbf{q}} = 1$ is given by the following eigenvalue problem:

$$\mathbf{L}\tilde{\mathbf{q}} = \lambda\tilde{\mathbf{q}}.$$

The matrix $\mathbf{L}$ is the Laplacian of the graph, and it is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, though other definitions may be found in the literature. The suboptimal solution $\tilde{\mathbf{q}}$ is the eigenvector corresponding to the second smallest eigenvalue (also called the Fiedler vector). The cut-cost

criterion, however, has a bias for separating small sets of points. This is due to the fact that there are no restrictions related to the size or the balance of the clusters.

The normalized-cut criterion (Shi & Malik, 2000), defined as:

$$ncut(\mathcal{A},\mathcal{B}) = \frac{cut(\mathcal{A},\mathcal{B})}{d_{\mathcal{A}}} + \frac{cut(\mathcal{A},\mathcal{B})}{d_{\mathcal{B}}}$$

with $d_{\mathcal{A}} = \sum_{i \in \mathcal{A}} d_i$,

penalizes small sets or isolated points by taking into account the total weight of each cluster. Minimizing the normalized cut is equivalent to:

$$\min_{\mathbf{q}} \frac{\mathbf{q}^T\mathbf{L}\mathbf{q}}{\mathbf{q}^T\mathbf{D}\mathbf{q}}$$

such that $\mathbf{q} \in \{-1,1\}^{\ell}, \mathbf{q}^T\mathbf{D}\mathbf{1} = 0,$

where $\mathbf{1}$ is a $\ell \times 1$ vector of ones. However, this problem is NP complete in the same manner as the cut cost; an approximate solution can be found efficiently by relaxing the discrete constraint on $\mathbf{q}$. If $\mathbf{q}$ can take real values, then the normalized cut corresponds to the Rayleigh quotient of the following generalized eigenvalue problem:

$$\mathbf{L}\tilde{\mathbf{q}} = \lambda\mathbf{D}\tilde{\mathbf{q}}.$$

The constraint $\tilde{\mathbf{q}}^T\mathbf{D}\mathbf{1} = 0$ is automatically satisfied in the generalized eigenproblem. The relaxed solution to the normalized cut is the Fiedler vector. Figure 2 illustrates a comparison between the normalized-cut and cut costs.

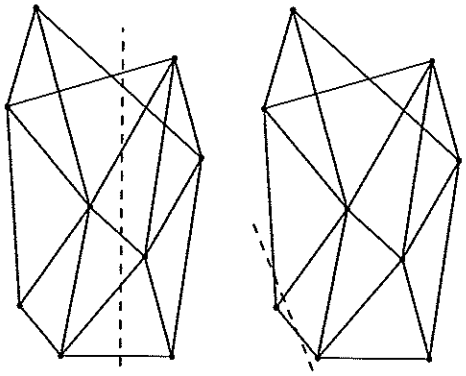
A different spectral algorithm was proposed in Ng et al. (2002), where the affinity matrix is first normalized using symmetric divisive normalization. The resulting eigenvalue problem is:

$$\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}\tilde{\mathbf{q}} = \lambda\tilde{\mathbf{q}}.$$

Whereas standard clustering methods often assume Gaussian class distributions, spectral clustering methods do not. In order to achieve this goal, one avoids the use of the Euclidean distance (as a dissimilarity measure) or the inner product (as a similarity or affinity measure). Instead, often an affinity measure based on an RBF kernel is used:

$$\mathbf{A}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2).$$

Figure 2. Partitioning for two different fully connected graphs. (Left) Normalized-cut cost of the graph attempts to produce balanced (similar-sized) clusters, while the cut cost (number of edges) is minimized. (Right) Cut cost favors cutting small sets of isolated nodes in the graph as the cut increases with the number of edges going across the two partitioned parts.



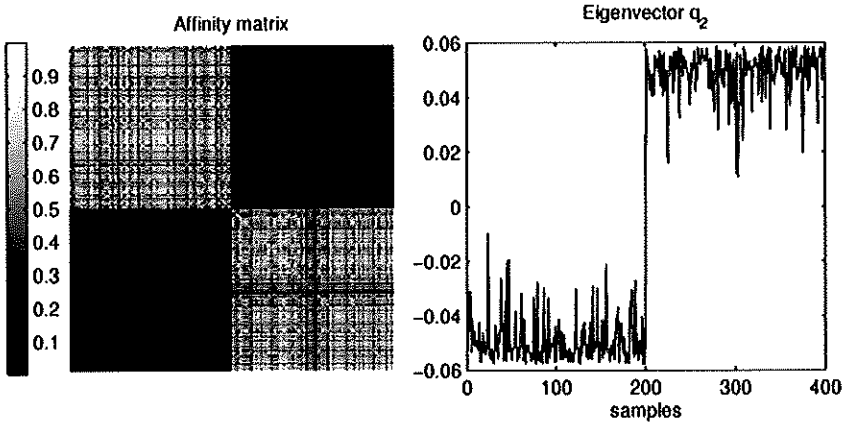
The width σ^2 of the kernel function controls how rapidly the affinity matrix $\mathbf{A}_{ij}$ falls off with the distance between x_i and x_j. Even though in spectral clustering often (though not exclusively) an RBF kernel is used, the positive definiteness of the RBF kernel is in fact not a requirement. On the other hand, affinities should be symmetric and all entries must be positive (which is the case indeed for the RBF kernel). The matrix containing the affinities between all pairs of samples should therefore be referred to as the affinity matrix in a spectral clustering context (and not as the kernel matrix).

Suppose a data set that contains samples $\mathbf{x}_1, ..., \mathbf{x}_\ell$. A well-known instance of spectral clustering, proposed by Ng et al. (2002), finds k clusters in the data as follows.

Spectral Clustering Algorithm

1. Form the affinity matrix A (with dimensions $\ell \times \ell$) defined by $\mathbf{A}_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2)$ if $i \neq j$, and $\mathbf{A}_{ii} = 0$.
2. Define $\mathbf{D}$ to be the diagonal matrix of which the element (i, i) is the sum of row i of $\mathbf{A}$, and construct the matrix $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$.
3. Find $\mathbf{u}_1, ..., \mathbf{u}_k$ the k largest eigenvectors of $\mathbf{L}$ (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_k]$ (with dimensions $\ell \times k$) by stacking the eigenvectors in columns.
4. Form the matrix $\mathbf{V}$ from $\mathbf{U}$ by renormalizing each row of $\mathbf{U}$ to have unit length, that is, $\mathbf{V}_{ij} = \mathbf{U}_{ij} / ((\sum_j \mathbf{U}_{ij}^2)^{1/2})$.
5. Treating each row of $\mathbf{V}$ as a sample with dimension k, cluster these into k clusters via k-means or any other algorithm (that attempts to minimize the distortion).
6. Finally, assign the original sample $\mathbf{x}_i$ to cluster C_j if and only if row i of the matrix $\mathbf{V}$ is assigned to cluster C_j.

Figure 3. (Left) The block diagonal structure of the affinity matrix clearly shows that two dense clusters with sparse connections between them are present. (Right) Eigenvector $\tilde{q}_2$ holds the information of the true partition of the data.



Conditions in which the algorithm is expected to do well are described by Ng et al. (2002). Once the samples are represented by rows of **V** (with dimension k), tight clusters are formed. An artificial example is illustrated in Figure 3.

Cluster-Validation Methods

Validation of the clustering results can be done internally, that is, by assessing the quality of a clustering result based on statistical properties, and externally, that is, reflecting the level of agreement of a clustering result with an external partition, for example, existing diagnostic classes generally used in clinical practice (Bolshakova et al., 2005; Halkidi & Vazirgiannis, 2005; Handl et al., 2005; Jain & Dubes, 1988; Milligan & Cooper, 1985). Moreover, internal cluster-validation techniques can also be used for selecting the best clustering result when comparing different clustering methods, several random initializations, a different number of clusters, a range of kernel parameters (e.g., the width σ^2 of the RBF kernel), and so forth. In this section, a formulation of three well-known internal validation methods in the input space (global silhouette index, Calinski-Harabasz index [F-statistic], and distortion score) and two external validation methods (Rand index and adjusted Rand index) are given first (applied in the input space) for reason of completeness. However, in order to be useful for kernel k-means clustering (and eventually other kernel clustering methods as well), we also derive the internal validation criteria for usage in the feature space.

Internal Validation

Global Silhouette Index

An expression pattern from a patient can be considered to be well clustered if its distance to the other expression patterns of the same cluster is small and the distance to the expression patterns of other clusters is larger. This criterion can be formalized by using the silhouette index (Kaufman & Rousseeuw, 1990), that is, for testing the cluster coherence. This measure validates the cluster result on statistical grounds only (statistical validation). Clinical information is not used here.

Suppose $\mathbf{x}_j$ is an expression pattern that belongs to cluster C_i. Call $v(\mathbf{x}_j)$ (also called the within dissimilarity) the average distance of x_j to all other expression patterns from C_i. Suppose C_h is a cluster different from C_i. Define $w(\mathbf{x}_j)$ (also called the between dissimilarity) as the minimum over all clusters C_h different from C_i of the average distance from $\mathbf{x}_j$ to all expression patterns of C_h. The silhouette width $s(\mathbf{x}_j)$ of expression patterns $\mathbf{x}_j$ is now defined as follows:

$$s(\mathbf{x}_j) = \frac{w(\mathbf{x}_j) - v(\mathbf{x}_j)}{\max(v(\mathbf{x}_j), w(\mathbf{x}_j))},$$

with $\mathbf{x}_j \in C_i$, and

$$v(\mathbf{x}_j) = \frac{1}{|C_i| - 1} \sum_{\substack{\mathbf{x}_l \in C_i \\ \mathbf{x}_l \neq \mathbf{x}_j}} \left\| \mathbf{x}_j - \mathbf{x}_l \right\|^2$$

$$w(\mathbf{x}_j) = \min_{h=1,\ldots,i-1,i+1,\ldots,k} \left(\frac{1}{|C_h|} \sum_{\mathbf{x}_l \in C_h} \left\| \mathbf{x}_j - \mathbf{x}_l \right\|^2 \right).$$

Note that $-1 \leq s(\mathbf{x}_j) \leq 1$. Consider two extreme situations now. First, suppose that the within dissimilarity $v(\mathbf{x}_j)$ is significantly smaller than the between dissimilarity $w(\mathbf{x}_j)$. This is the ideal case and $s(\mathbf{x}_j)$ will be approximately equal to one. This occurs when $\mathbf{x}_j$ is well clustered and there is little doubt that $\mathbf{x}_j$ is assigned to an appropriate cluster. Second, suppose that $v(\mathbf{x}_j)$ is significantly larger than $w(\mathbf{x}_j)$. Now $s(\mathbf{x}_j)$ will be approximately -1 and $\mathbf{x}_j$ has in fact been assigned to the wrong cluster (worst-case scenario).

Two other measures can now be defined: the average silhouette width of a cluster and the average silhouette width of the entire data set. The first is defined as the average of $s(\mathbf{x}_j)$ for all expression patterns of a cluster, and the second is defined as the average of $s(\mathbf{x}_j)$ for all expression patterns in the data set. This last value can be used to mutually compare different cluster results and can be used as an inherent part of clustering algorithms if its value is optimized during the clustering process.

When using this validation measure in combination with kernel clustering methods performing the actual clustering in the feature space, for example, kernel k-means clustering, using

this definition of the silhouette index leads to wrong results since the distances between the expression patterns are computed in the input space. We therefore derive the definition of the silhouette index for computation in the feature space.

By introducing the feature map $\phi(\cdot)$, $v(\mathbf{x}_j)$ and $w(\mathbf{x}_j)$ can be expressed in the feature space as:

$$v^{\phi}(\mathbf{x}_j) = \frac{1}{(|C_i|-1)} \sum_{\substack{\mathbf{x}_l \in C_i \\ \mathbf{x}_l \neq \mathbf{x}_j}} \left\| \phi(\mathbf{x}_j) - \phi(\mathbf{x}_l) \right\|^2$$

$$w^{\phi}(\mathbf{x}_j) = \min_{h=1,\ldots,i-1,i+1,\ldots,k} \left(\frac{1}{|C_h|} \sum_{\mathbf{x}_j \in C_h} \left\| \phi(\mathbf{x}_j) - \phi(\mathbf{x}_l) \right\|^2 \right), \text{ for } \mathbf{x}_j \in C_i.$$

Replacing all the dot products by a kernel function $\kappa(\cdot,\cdot)$ results in:

$$v^{\phi}(\mathbf{x}_j) = \frac{1}{(|C_i|-1)} \kappa(\mathbf{x}_j, \mathbf{x}_j) - \frac{2}{(|C_i|-1)} \sum_{\substack{\mathbf{x}_l \in C_i \\ \mathbf{x}_l \neq \mathbf{x}_j}} \kappa(\mathbf{x}_j, \mathbf{x}_l) + \frac{1}{(|C_i|-1)} \sum_{\substack{\mathbf{x}_l \in C_i \\ \mathbf{x}_l \neq \mathbf{x}_j}} \kappa(\mathbf{x}_l, \mathbf{x}_l)$$

$$w^{\phi}(\mathbf{x}_j) = \min_{h=1,\ldots,i-1,i+1,\ldots,k} \left(\frac{1}{|C_h|} \kappa(\mathbf{x}_j, \mathbf{x}_j) - \frac{2}{|C_h|} \sum_{\mathbf{x}_l \in C_h} \kappa(\mathbf{x}_j, \mathbf{x}_l) + \frac{1}{|C_h|} \sum_{\mathbf{x}_l \in C_h} \kappa(\mathbf{x}_l, \mathbf{x}_l) \right), \text{ for } \mathbf{x}_j \in C_i.$$

Consequently, the silhouette index can be computed in the feature space as:

$$s^{\phi}(\mathbf{x}_j) = \frac{w^{\phi}(\mathbf{x}_j) - v^{\phi}(\mathbf{x}_j)}{\max(v^{\phi}(\mathbf{x}_j), w^{\phi}(\mathbf{x}_j))}.$$

Calinski-Harabasz Index

The Calinski-Harabasz index (Calinski & Harabasz, 1974; Milligan & Cooper, 1985), also called *F*-statistic, is a measure of intercluster dissimilarity over intracluster dissimilarity. For ℓ expression patterns and k clusters, the Calinski-Harabasz index CH is defined as:

$$CH = \frac{trB/(k-1)}{trW/(\ell-k)},$$

with B and W the between- and within-cluster scatter matrices (measures of dissimilarity), respectively. A larger value for CH indicates a better clustering since the between-cluster dissimilarity is then supposed to be large, while the within-cluster dissimilarity is then supposed to be small. Maximum values of the CH index are often used to indicate the correct

number of partitions in the data. The trace of the between-cluster scatter matrix B can be written as:

$$trB = \sum_{i=1}^{k} |C_i| \|\mathbf{m}_i - \mathbf{m}\|^2 = \sum_{i=1}^{k} |C_i| \left\| \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j - \frac{1}{\ell} \sum_{\mathbf{x}_l \in S} \mathbf{x}_l \right\|^2,$$

where $|C_i|$ denotes the number of elements in cluster C_i with centroid m_i, and m the centroid of the entire data set S. The trace of the within-cluster scatter matrix W can be written as:

$$trW = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \mathbf{m}_i\|^2 = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \left\| \mathbf{x}_j - \frac{1}{|C_i|} \sum_{\mathbf{x}_l \in C_i} \mathbf{x}_l \right\|^2.$$

Therefore, the CH index can be written as:

$$CH = \frac{\sum_{i=1}^{k} |C_i| \left\| \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j - \frac{1}{\ell} \sum_{\mathbf{x}_l \in S} \mathbf{x}_l \right\|^2 \Big/ k-1}{\sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \left\| \mathbf{x}_j - \frac{1}{|C_i|} \sum_{\mathbf{x}_l \in C_i} \mathbf{x}_l \right\|^2 \Big/ \ell - k}.$$

By introducing the feature map $\phi(\cdot)$, the traces of the between-cluster scatter matrix B and of the within-cluster scatter matrix W can be expressed as:

$$trB^{\phi} = \sum_{i=1}^{k} |C_i| \left\| \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} \phi(\mathbf{x}_j) - \frac{1}{\ell} \sum_{\mathbf{x}_l \in S} \phi(\mathbf{x}_l) \right\|^2$$

and

$$trW^{\phi} = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in C_i} \left\| \phi(\mathbf{x}_j) - \frac{1}{|C_i|} \sum_{\mathbf{x}_l \in C_i} \phi(\mathbf{x}_l) \right\|^2.$$

After applying the kernel trick (as done for the global silhouette index), the CH index can be calculated in feature space by:

$$CH^{\phi} = \frac{trB^{\phi}/(k-1)}{trW^{\phi}/(\ell - k)}.$$

Distortion Score

The mean squared error criterion, which is the objective function in both classical and kernel *k*-means clustering, can be used for internal validation. In this context, the mean squared error criterion is called the distortion score.

For a set of expression patterns $\mathbf{x}_j, j = 1, ..., \ell$, the distortion score is formulated as:

$$\text{se} = \sum_{i=1}^{k}\sum_{j=1}^{\ell} z_{C_i,\mathbf{x}_j} \left\|\mathbf{x}_j - \mathbf{m}_i\right\|^2,$$

with the indicator function $z_{C_i,\mathbf{x}_j}$ defined as

$$z_{C_i,\mathbf{x}_j} = \begin{cases} 1 & \text{if } \mathbf{x}_j \in C_i \\ 0 & \text{otherwise} \end{cases}$$

with

$$\sum_{i=1}^{k} z_{C_i,\mathbf{x}_j} = 1 \qquad \forall j$$

and the centroid (or prototype) $\mathbf{m}_i$ of cluster C_i defined as

$$\mathbf{m}_i = \frac{1}{|C_i|}\sum_{j=1}^{\ell} z_{C_i\mathbf{x}_j}\mathbf{x}_j.$$

In the feature space, the distortion score can be expressed by:

$$\begin{aligned}\text{se}^{\phi} &= \sum_{i=1}^{k}\sum_{j=1}^{\ell} z_{C_i,\mathbf{x}_j} \left\|\phi(\mathbf{x}_j) - \mathbf{m}_i^{\phi}\right\|^2 \\ &= \sum_{i=1}^{k}\sum_{j=1}^{\ell} z_{C_i,\mathbf{x}_j} (\mathbf{K}_{jj} + f(C_i,\mathbf{x}_j) + g(C_i)),\end{aligned}$$

with $f(C_i, \mathbf{x}_j)$, $g(C_i)$, and $\mathbf{m}_i^{\phi}$ defined as in the kernel *k*-means algorithm.

External Validation

Rand Index

The Rand index (Rand, 1971; Yeung, Fraley, Murua, Raftery, & Ruzzo, 2001; Yeung, Haynor, & Ruzzo, 2001) is a measure that reflects the level of agreement of a cluster result with an external partition, that is, an existing partition of a known cluster structure of the data. This external criterion could, for example, be the existing diagnostic classes generally used by experts in clinical practice (e.g., groups of patients with a similar type of cancer, groups of patients responding to therapy in a similar way, or groups of patients with a similar kind of diagnosis), a predefined cluster structure if one is clustering synthetic data where the clusters are known in advance, or another cluster result obtained using other parameter settings for a specific clustering algorithm or obtained using other clustering algorithms. Note that the latter could be used to investigate how sensitive a cluster result is to the choice of the algorithm or parameter setting. If this result proves to be relatively stable, one could assume that pronounced structures are present in the data possibly reflecting subcategories that are clinically relevant.

Suppose one wants to compare two partitions (the cluster result at hand and the external criterion) of a set of ℓ expression patterns. Suppose that a is the number of expression pattern pairs that are placed in the same subset (or cluster) in both partitions. Suppose that d is the number of expression pattern pairs that are placed in different subsets in both partitions. The Rand index is then defined as the fraction of agreement between both partitions:

$$r = \frac{a+d}{M}$$

with M as the maximum number of all expression pattern pairs in the data set, that is, $M = \ell(\ell - 1)/2$. This can also be rewritten by $M = a + b + c + d$, with b as the number of expression pattern pairs that are placed in the same cluster according to the external criterion but in different clusters according to the cluster result, and c as the number of expression pattern pairs that are placed in the same cluster according to the cluster result but in different clusters according to the external criterion. The Rand index lies between zero and one (one if both partitions are identical), and can be viewed as the proportion of agreeing expression pattern pairs between two partitions.

Adjusted Rand Index

One disadvantage of the Rand index is that the expected value of two random partitions is not a constant value (Yeung, Haynor, et al., 2001) and depends on the number of clusters k. In order to compare clustering results with different numbers of clusters k, the adjusted Rand index was proposed by Hubert and Arabie (1985).

The general form of an index with a constant expected value is:

$$\frac{\text{index} - \text{expected index}}{\text{maximum index} - \text{expected index}},$$

which is bounded above by 1 and below by -1, and has an expected value of 0 for random clustering.

Let n_{ij} be the number of expression patterns that are in both cluster u_i (according to the external criterion) and cluster v_j (according to the cluster result). Let n_i and n_j be the number of expression patterns in cluster u_i and cluster v_j, respectively. According to Hubert and Arabie (1985), the adjusted Rand index can be expressed in a simple form by:

$$AR = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \left(\sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}\right)\Big/\binom{n}{2}}{\frac{1}{2}\left(\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right) - \left(\sum_i \binom{n_{i\cdot}}{2}\sum_j \binom{n_{\cdot j}}{2}\right)\Big/\binom{n}{2}}.$$

Experiments

In this section, the clustering and cluster-validation methods described are demonstrated on acute leukemia data (Golub et al., 1999) and colon cancer data (Alon et al., 1999). Golub et al. studied microarray data obtained from the bone marrow or peripheral blood of 72 patients with ALL or AML using an Affymetrix chip. Although the structure of this data set is simple and the separation between the two conditions is more pronounced than in most other cases, it can still be considered as a frequently used benchmark. The data set contains 47 patients with ALL and 25 patients with AML. The expression matrix contains 7,129 genes. Alon et al. studied 40 tumor and 22 normal colon tissue samples using an Affymetrix chip. The array contained probes for more than 6,500 genes, but the data that can be downloaded includes only the 2,000 genes with the highest minimal intensity across the 62 tissues.

Preprocessing of the acute leukemia data set is done by thresholding and log transformation, similar to how it was done in the original publication. Thresholding is achieved by restricting gene expression levels to be larger than 20; that is, expression levels that are smaller than 20 will be set to 20. Concerning the log transformation, the natural logarithm of the expression levels is taken. For the colon cancer data set, only log transformation is done, as in the original publication. Further preprocessing of both data sets is done by standardization (normalization). For classical *k*-means, this is followed by principal component analysis (without the selection of principal components). Although kernel clustering techniques are capable of handling high-dimensional data, one should not forget the possible benefits of performing preprocessing steps that remove noise before using any clustering technique.

Tuning of the hyperparameters is an important issue, discussed previously in a large number of publications. We therefore only refer to some of these studies (Halkidi & Vazirgiannis, 2005; Handl et al., 2005). However, since kernel clustering methods require tuning of the kernel parameters as well, some research effort still needs to be performed on this subject.

Note that classical *k*-means clustering and kernel *k*-means clustering with a linear kernel require the optimization of a number of clusters and the random initialization. Kernel *k*-means with an RBF kernel and spectral clustering, however, also require the additional optimization of the kernel parameter σ^2. Tuning these hyperparameters needs to be performed based on internal validation criteria.

Results

Since both data sets contain two given diagnostic categories, we restrict the number of clusters *k* to be equal to two. The initialization is optimized by repeating each *k*-means or kernel *k*-means algorithm 100 times, selecting the best result based on the distortion score within these algorithms (note that this is done for each value of σ^2). Optimization of this kernel parameter σ^2 is done based on the global silhouette index. Note that only intervals for σ^2 with meaningful cluster results are considered. For the optimal value of σ^2, both external validation indices (i.e., the Rand and adjusted Rand index) are reported as well.

Silhouette plots (representing for each cluster the sorted silhouette indices for all samples) and tuning curves (tuning the kernel parameter σ^2 based on the global silhouette index), followed by a table presenting global silhouette, Rand, and adjusted Rand indices for the optimal kernel parameter σ^2, are first shown for the acute leukemia data and then for the colon cancer data.

From these results, we can conclude that spectral clustering, unlike the other clustering algorithms, gives very good and consistent clustering results in terms of the global silhouette index (internal validation) for both data sets. Note that the results obtained by any optimally tuned clustering algorithm (classical *k*-means, kernel *k*-means with linear or RBF kernel, and spectral clustering) are not correlated to the given diagnostic categories (external partitions). However, this does not mean that these clustering results are clinically or biologically irrelevant; that is, these could correspond to other known or unknown diagnostic categories.

Figure 4. Silhouette plots of classical k-means (left) and kernel k-means clustering (right) on the acute leukemia data. These plots show the sorted silhouette indices (x-axis) for all samples in each cluster (y-axis).

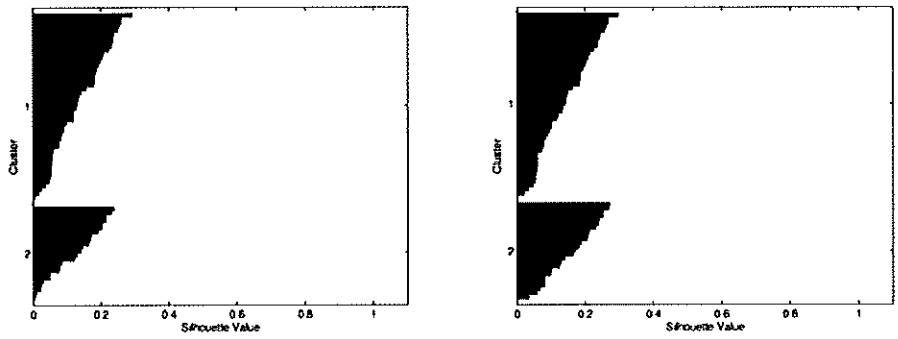


Figure 5. Tuning curve (left) and silhouette plot (right) of kernel k-means clustering on the acute leukemia data. The tuning curve shows the global silhouette index (y-axis) for a range of values for kernel parameter σ^2 (x-axis). The silhouette plot shows the sorted silhouette indices (x-axis) for all samples in each cluster (y-axis).

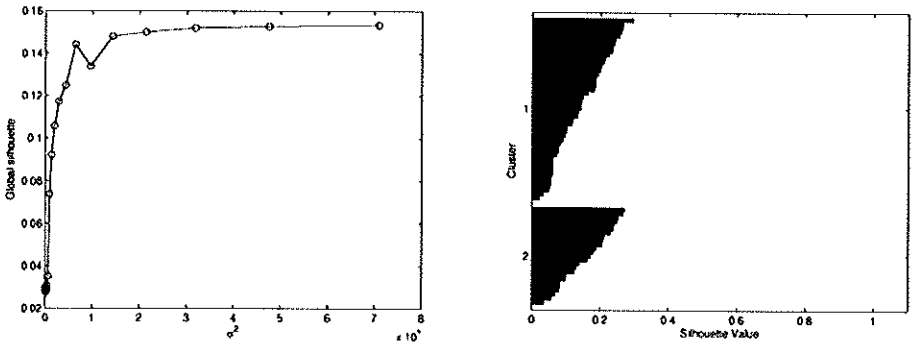


Figure 6. Tuning curve (left) and silhouette plot (right) of spectral clustering on the acute leukemia data. See Figure 5 for more detailed information on the tuning curve and silhouette plot.

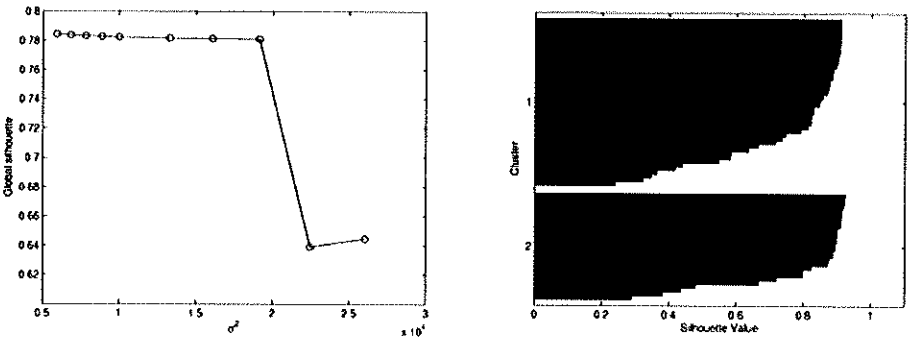


Table 1. Global silhouette, Rand, and adjusted Rand indices for the optimal kernel parameter σ^2 are given for all clustering methods on the acute leukemia data. There are two conclusions: (a) Spectral clustering clearly gives the best results in terms of internal validation, and (2) external validation shows that the clustering results are not correlated to the given diagnostic categories. However, these results could correspond to other known or unknown diagnostic categories.

	Kernel parameter σ^2	Global silhouette index	Adjusted Rand index	Rand index
k-means clustering	-	0.12988	-0.021418	0.49335
Kernel *k*-means clustering with linear kernel	-	0.15456	-0.017564	0.49452
Kernel *k*-means clustering with RBF kernel	709220.0	0.15337	-0.017564	0.49452
Spectral clustering	5913.0	0.78436	0.00258	0.49656

Figure 7. Silhouette plots of classical k-means (left) and kernel k-means clustering (right) on the colon cancer data. See Figure 4 for more detailed information on the silhouette plot.

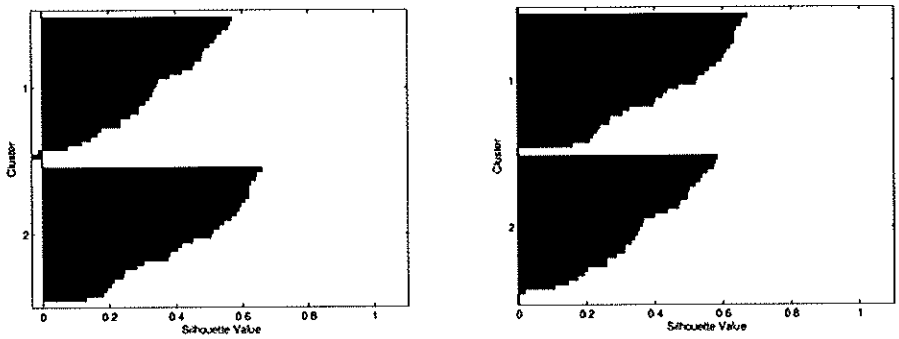


Figure 8. Tuning curve (left) and silhouette plot (right) of kernel k-means clustering on the colon cancer data. See Figure 5 for more detailed information on the tuning curve and silhouette plot.

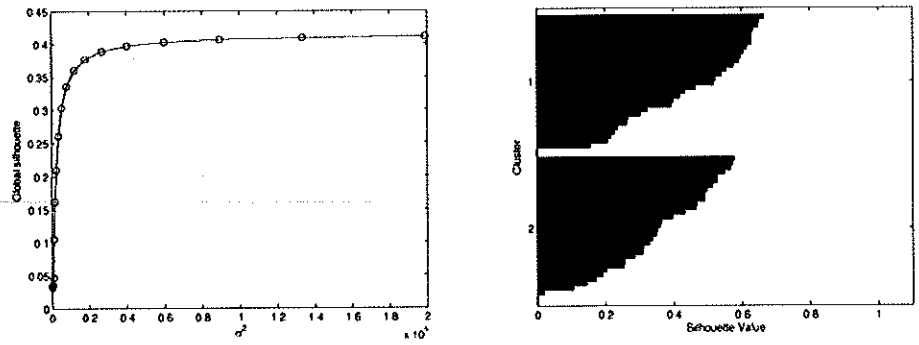


Figure 9. Tuning curve (left) and silhouette plot (right) of spectral clustering on the colon cancer data. See Figure 5 for more detailed information on the tuning curve and silhouette plot.

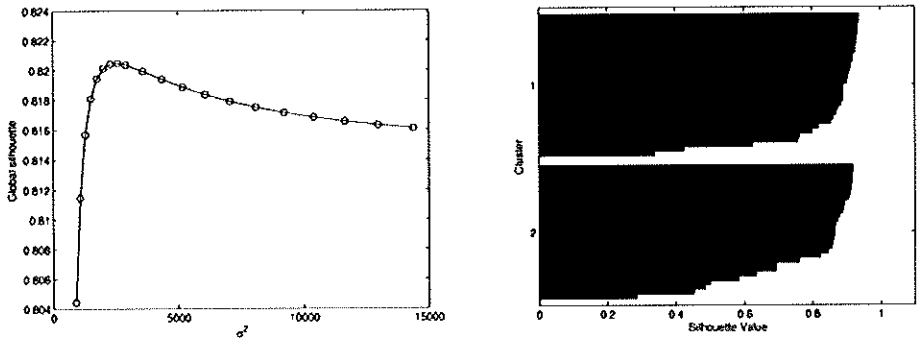


Table 2. Global silhouette, Rand, and adjusted Rand indices for the optimal kernel parameter σ^2 are given for all clustering methods on the colon cancer data. There are two conclusions: (a) Spectral clustering clearly gives the best results in terms of internal validation, and (b) external validation shows that the clustering results are not correlated to the given diagnostic categories. However, these results could correspond to other known or unknown diagnostic categories.

	Kernel parameter σ^2	Global silhouette index	Adjusted Rand index	Rand index
k-means clustering	-	0.3948	-0.0058061	0.49656
Kernel *k*-means clustering with linear kernel	-	0.41423	-0.0058	0.49656
Kernel *k*-means clustering with RBF kernel	198970.0	0.41073	-0.0058061	0.49656
Spectral clustering	2596.4	0.82046	-0.0058	0.49656

Conclusion

Most of the classical techniques that have previously been applied to analyze microarrays rely on specifically designed procedures in order to deal with the particular challenges posed by the gene expression data at hand (Alon et al., 1999; Golub et al., 1999). Therefore, these procedures are not guaranteed to perform well on other microarray data sets and cannot be considered as a general approach. In most publications, some way of input selection of a subset of relevant genes is performed first instead of systematically analyzing thousands of genes simultaneously. Although performing gene selection before clustering may improve the discrimination between the known groups, the choice for the best gene-selection method is highly dependent on the data set considered. The techniques presented here do not make any assumptions neither on the distribution of the data nor on the relevance on the input variables (genes), providing a more general approach that can be systematically extended to other microarray data sets. The model-selection step, that is, the choice of the kernel width and the choice of the optimal number of clusters, is often skipped for both kernel *k*-means and spectral clustering in most of the related publications. In this chapter, at least three variants for tuning this parameter based on different internal quality measures of the clusters have been proposed. Since these are internal measures, however, high correlations with external partitions are not ensured. Consequently, more sophisticated methods rather than the silhouette index (e.g., kernel alignment [Cristianini et al., 2002] and bounds derived from the formulation [Shi & Malik, 2000]) need to be considered or even defined for model selection. This way, the results and correlation with external partitions could be further improved.

In summary, kernel clustering methods like kernel *k*-means and spectral clustering are especially designed for clustering data that contain clusters that are not linearly separable in order to handle nonlinear relationships in the data. Moreover, these techniques allow for dealing with high-dimensional data. It was shown in this chapter that these properties make

kernel clustering methods specifically interesting for application on the high-dimensional microarray data (with or without preprocessing steps). Using these techniques for knowledge discovery in clinical microarray data analysis may therefore allow the discovery of new clinically relevant groups in the future.

Acknowledgments

Nathalie Pochet is a research assistant of the IWT at the Katholieke Universiteit Leuven, Belgium. Fabian Ojeda is a research assistant at the Katholieke Universiteit Leuven, Belgium. Frank De Smet is a postdoctoral research assistant at the Katholieke Universiteit Leuven, Belgium, and a medical advisor at the National Alliance of Christian Mutualities, Belgium. Tijl De Bie is a postdoctoral research assistant at the Katholieke Universiteit Leuven, Belgium. Johan Suykens is an associate professor at the Katholieke Universiteit Leuven, Belgium. Bart De Moor is a full professor at the Katholieke Universiteit Leuven, Belgium. Research was supported by GOA-Mefisto 666, GOA-Ambiorics, IDO (IOTA), and several PhD, postdoc, and fellow grants of the Research Council KUL; PhD and postdoc Grants and Projects G.0240.99, G.0115.01, G.0407.02, G.0197.02, G.0141.03, G.0491.03, G.0120.03, G.0413.03, G.0388.03, G.0229.03, G.0241.04, G.0452.04, and G.0499.04, and research communities ICCoS, ANMMM, and MLDM of the Flemish government (FWO); Bil. Int. Collaboration Hungary/Poland (AWI); PhD Grants STWW-Genprom, GBOU-McKnow, GBOU-SQUAD, and GBOU-ANA of IWT; DWTC: IUAP V-22 (2002-2006) and PODO-II (CP/01/40) of the Belgian federal government; FP5 CAGE of the European Union; ERNSI; NoE Biopattern; NoE E-tumours; Eureka 2063-IMPACT; Eureka 2419-FliTE; and contracts and research agreements with ISMC/IPCOS, Data4s, TML, Elia, LMS, IPCOS, and Mastercard.

References

Alon, A., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., et al. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. In *Proceedings of the National Academy of Science USA, 96*, (pp. 6745-6750).

Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. *Journal of Machine Learning Research, 2*, 125-137.

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning* (pp. 19-26).

Bolshakova, N., & Azuaje, F. (2003). Cluster validation techniques for genome expression data. *Signal Processing, 83*, 825-833.

Bolshakova, N., Azuaje, F., & Cunningham, P. (2005). An integrated tool for microarray data clustering and cluster validity assessment. *Bioinformatics, 21*(4), 451-455.

Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics, 3*(1), 1-27.

Cristianini, N., Shawe-Taylor, J., & Kandola, J. (2002). Spectral kernel methods for clustering. *Advances in Neural Information Processing Systems, 14.*

De Bie, T., Cristianini, N., & Rosipal R. (2004). Eigenproblems in pattern recognition. In E. Bayro-Corrochano (Ed.), *Handbook of computational geometry for pattern recognition, computer vision, neurocomputing and robotics.* Springer-Verlag.

De Smet, F., Mathys, J., Marchal, K., Thijs, G., De Moor, B., & Moreau, Y. (2002). Adaptive quality based clustering of gene expression profiles. *Bioinformatics, 18*(5), 735-746.

Dhillon, I. S., Guan, Y., & Kulis, B. (2004a). Kernel *k*-means, spectral clustering and normalized cuts. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 551-556).

Dhillon, I. S., Guan, Y., & Kulis, B. (2004b). *A unified view of kernel k-means: Spectral clustering and graph partitioning* (Tech. Rep. No. TR-04-25). UTCS.

Dubes, R., & Jain, A. K. (1979). Validity studies in clustering methodologies. *Pattern Recognition Letters, 11*, 235-254.

Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. In *Proceedings of the National Academy of Science USA, 95* (pp. 14863-14868).

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gassenbeck, M., Mesirov, J. P., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science, 286*, 531-537.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems, 17*(2-3), 107-145.

Halkidi, M., & Vazirgiannis, M. (2005). Quality assessment approaches in data mining. In *The data mining and knowledge discovery handbook* (pp. 661-696).

Handl, J., Knowles, J., & Kell, D. B. (2005). Computational cluster validation in post-genomic data analysis. *Bioinformatics, 21*, 3201-3212.

Hastie, T., Tibshirani, R., Eisen, M. B., Alizadeh, A., Levy, R., Staudt, L., et al. (2000). Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology, 1*, 1-21.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 193-218.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data.* Englewood Cliffs, NJ: Prentice Hall.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning* (pp. 290-297).

Joliffe, I. T. (1986). *Principal component analysis.* Springer-Verlag.

Kannan, R., Vempala, S., & Vetta, A. (2004). On clusterings: Good, bad and spectral. *Journal of the ACM, 51*(3), 497-515.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 281-297).

Madeira, S. C., & Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics, 1*, 24-45.

Mika, S., Schölkopf, B., Smola, A. J., Müller, K. R., Scholz, M., & Rätsch, G. (1999). Kernel PCA and de-noising in feature spaces. *Advances in Neural Information Processing Systems, 11*.

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters. *Psychometrika, 50*, 159-179.

Ng, A., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems, 14*.

Pochet, N., De Smet, F., Suykens, J. A. K., & De Moor, B. L. R. (2004). Systematic benchmarking of microarray data classification: Assessing the role of nonlinearity and dimensionality reduction. *Bioinformatics, 20*, 3185-3195.

Qin, J., Lewis, D. P., & Noble, W. S. (2003). Kernel hierarchical gene clustering from microarray expression data. *Bioinformatics, 19*, 2097-2104.

Quackenbush, J. (2001). Computational analysis of microarray data. *Nature Reviews Genetics*, 418-427.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association, 66*, 846-850.

Rosen, J. E., Costouros, N. G., Lorang, D., Burns, A. L., Alexander, H. R., Skarulis, M. C., et al. (2005). Gland size is associated with changes in gene expression profiles in sporadic parathyroid adenomas. *Annals of Surgical Oncology, 12*(5), 412-416.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 53-65.

Schölkopf, B., Smola, A., & Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*, 1299-1319.

Sheng, Q., Moreau, Y., & De Moor, B. (2003). Biclustering microarray data by Gibbs sampling. *Bioinformatics, European Conference on Computational Biology Proceedings*, 19, ii196-ii205.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 808-905.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least squares support vector machines.* Singapore: World Scientific Publishing.

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., & Church, G. M. (1999). Systematic determination of genetic network architecture. *Nature Genetics*, 22, 281-285.

Van 't Veer, L. J., Dai, H., Van De Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature, 415*, 530-536.

Vapnik, V. N. (1998). *Statistical learning theory.* New York: Wiley-Interscience.

Yeung, K., Fraley, C., Murua, A., Raftery, A., & Ruzzo, W. (2001). Model-based clustering and data transformations for gene expression data. *Bioinformatics, 17*(10), 977-987.

Yeung, K., Haynor, D., & Ruzzo, W. (2001). Validating clustering for gene expression data. *Bioinformatics, 17*(4), 309-318.

Zhang, R., & Rudnicky, A. I. (2002). A large scale clustering scheme for kernel *k*-means. In *Proceedings of the International Conference on Pattern Recognition.*

Gustavo Camps-Valls, Jose Luis Rojo-Alvarez & Manel Martinez-Ramon

Kernel Methods in Bioengineering, Signal and Image

Kernel Methods in Bioengineering, Signal and Image Processing

Gustavo Camps-Valls, Universitat de València, Spain

José Luis Rojo-Álvarez, Universidad Rey Juan Carlos, Spain

Manel Martínez-Ramón, Universidad Carlos III de Madrid, Spain

IDEA GROUP PUBLISHING
Hershey • London • Melbourne • Singapore

Acquisition Editor:	Kristin Klinger
Senior Managing Editor:	Jennifer Neidig
Managing Editor:	Sara Reed
Assistant Managing Editor:	Sharon Berger
Development Editor:	Kristin Roth
Copy Editor:	Shanelle Ramelb
Typesetter:	Jamie Snavely
Cover Design:	Lisa Tosheff
Printed at:	Integrated Book Technology

Published in the United States of America by
Idea Group Publishing (an imprint of Idea Group Inc.)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@idea-group.com
Web site: http://www.idea-group.com

and in the United Kingdom by
Idea Group Publishing (an imprint of Idea Group Inc.)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 3313
Web site: http://www.eurospan.co.uk

Library of Congress Cataloging-in-Publication Data

Kernel methods in bioengineering, signal and image processing / Gustavo Camps-Valls, José Luis Rojo-Álvarez and Manel Martínez-Ramón, editors.
p. cm.
Summary: "This book presents an extensive introduction to the field of kernel methods and real world applications. The book is organized in four parts: the first is an introductory chapter providing a framework of kernel methods; the others address Bioegineering, Signal Processing and Communications and Image Processing"--Provided by publisher.
Includes bibliographical references and index.
ISBN 1-59904-042-5 (hardcover) -- ISBN 1-59904-043-3 (softcover) -- ISBN 1-59904-044-1 (ebook)
1. Engineering mathematics. 2. Biomedical engineering--Mathematics. 3. Signal processing--Mathematics. 4. Cellular telephone systems--Mathematics. 5. Image processing--Mathematics. 6. Kernel functions. I. Camps-Valls, Gustavo, 1972- II. Rojo-Alvarez, Jose Luis, 1972- III. Martínez-Ramón, Manel, 1968-
TA335.K47 2006
610.28--dc22

2006027728

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Kernel Methods in Bioengineering, Signal and Image Processing

Table of Contents