Unsupervised Kernel Parameter Estimation by Constrained Nonlinear Optimization for Clustering Nonlinear Biological Data

Hyokyeong Lee and Rahul Singh*
Department of Computer Science, San Francisco State University, San Francisco, CA 94132
hyolee@sfsu.edu, *rahul@sfsu.edu (Corresponding author)

***Abstract*—Data on a wide-range of bio-chemical phenomena is often highly non-linear. Due to this characteristic, data analysis tasks, such as clustering can become non-trivial. In recent years, the use of kernel-based algorithms has gained popularity for data analysis and clustering to ameliorate the above challenges. In this paper, we propose a novel approach for kernel parameter estimation using constrained nonlinear programming and conditionally positive definite kernels. The central idea is to maximize the trace of the kernel matrix, which maximizes the variance in the feature space. Therefore, the parameter estimation process does not involve any user intervention or prior understanding of the data and the parameters are learned only from data. The results from the proposed method significantly improve upon results obtained with other leading non-linear analysis techniques.**

Keywords: expression analysis, kernel methods, constrained nonlinear optimization, sequential quadratic programming

I. INTRODUCTION

In the recent past, DNA microarray technology [6], has become commonplace for holistic investigation of the expression response of thousands of genes under multiple conditions. Clustering constitutes one of the basic methods for expression data analysis and a number of methods have been proposed in this area till date. One of the fundamental challenges associated with clustering gene expression data is that the clusters are often linearly inseparable; that is, they may be interconnected, may overlap, or may even be embedded one in another. Unfortunately, most existing clustering methods have difficulty in handling nonlinearly separable data.

Kernel methods constitute a promising class of techniques for non-linear classification tasks and have been successfully applied in multiple biological problems [2]. Their basic idea lies in mapping the input data points into a higher dimensional space and then search for linear relations that can be used to discriminate the data points. In spite of their advantages, challenges remain in the practical application of kernel methods. In the unsupervised data analysis context being considered in this paper, we distinguish two primary and interwoven challenges, namely, determining the kernel structure and kernel parameter estimation.

In kernel methods, the kernel matrix is required to be symmetric and positive-definite (PD). This ensures that the quadratic minimization problem is convex. Furthermore, any PD kernel can be written as a dot product of the function that maps the input feature space to a high dimensional space. The class of conditionally positive definite (CPD) kernels, which subsumes PD kernels has long been studied theoretically and has gained prominence in the recent past in applications to real-world problems. CPD kernels provide an alternative to the classical PD kernels in machine learning tasks and can be theoretically linked to PD kernels.

However, simply using a PD or CPD kernel does not guarantee an optimal kernel matrix, i.e. a kernel that leads to maximum linear separability in the feature space. Furthermore, a PD kernel cannot ensure a globally optimal solution if the kernel parameters are not optimal. Therefore, finding optimal parameters of the chosen kernel is critical to kernel methods.

In this paper, we propose a novel method to optimally estimate the parameters of CPD kernels based on line search sequential quadratic programming (LS-SQP) and apply it to non-linear biological data clustering problems. For purposes of specificity, we demonstrate the proposed LS-SQP-based parameter estimation approach using sigmoid kernels, which constitutes one of the commonly used kernel functions. However, the method can be extended to other kernel types also. The key advantages of this method include unsupervised learning of the parameters, the ability to incorporate linear or non-linear constraints in the search process, and high accuracy especially when compared to direct search methods [15].

II. PRIOR WORK

Clustering microarray gene expression is a widely studied problem and a large number of clustering methods have been developed to group either genes or conditions. A partial listing of the different classes of methods includes graph-based algorithms [1, 5, 13] such as Bayesian clustering and nearest neighbor network algorithm, systems theory [14], bi-clustering [4, 17], mixture models [18], and kernel methods [27].

In the context of mapping or dimensionality reduction of non-linear data, manifold learning is a recently developed approach. The idea of manifold learning is that while the dimensionality of data is high, each data point may be described as a function of only a few underlying parameters [3]. Manifold learning algorithms attempt to discover these parameters so that the data can be represented in low dimensions. Examples of the manifold learning algorithms include isometric feature mapping (Isomap) [26] and locally linear embedding (LLE) [21].

978-1-4673-2560-8/12/$31.00 ©2012 IEEE

III. KERNEL METHODS

A. Kernel Functions

Kernel methods map input data using nonlinear mapping into a high dimensional space called feature space and then search for linear relations among the data points in the feature space [7, 23]. Suppose we are given a set of data $X = \{x_i | x_i \in \mathbf{R}^D,\ i = 1,\ldots,n\}$. Kernel methods map the data into the high dimensional space □

$$\Phi : X \to F, \qquad x \mapsto \Phi(x), \tag{1}$$

where Φ is a mapping. To avoid computing the mapping explicitly, we can use a nonlinear function in the input space, i.e. by using the kernel trick

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle . \tag{2}$$

Instead of specifying the actual form of Φ, a kernel function is chosen. The commonly used kernels are Gaussian kernel, polynomial kernel and sigmoid kernel. A polynomial kernel is suited for problems when the training data are normalized. The sigmoid kernel is popular for support vector machines due to its origin from neural networks. A question may arise as to which kernel function gives the best result; Gaussian, polynomial and sigmoid kernels can each lead to good results in different circumstances and there are no golden rules for choosing the best kernel among these three [11].

We next define the notions of positive definite (PD) kernels, conditionally positive definite (CPD) kernels and demonstrate the connection between them.

Definition 1: Let □ be a non-empty set. A symmetric function K: □ × □→□ is called a kernel. K is called positive definite (PD) kernel if $\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geq 0$, $x_1, \ldots x_n \in$ □ and $c_1, \ldots c_n \in R$. K is said to be strictly PD if for any $x_1, \ldots x_n \in$ □, the above inequality is strict. In such a case, the matrix $[K_{ij}]$ is positive definite and not just positive semidefinite.

Definition 2: The kernel K is called conditionally positive definite (CPD) of order 1, if $\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geq 0$ $x_1, \ldots x_n \in$ □ and $c_1, \ldots c_n \in R$ subject to the constraint $\sum_{i=1}^{n} c_i = 0$. K is strictly CPD, if $\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) > 0$.

The relationship between PD matrices and CPD matrices is given by the following result [10].

Theorem 1: The matrix $[K_{ij}]$ is CPD and Hermitian *iff* for $A_{ij}=exp(K_{ij})$, the matrix $[A_{ij}^{\beta}]$ obtained by element-wise exponentiation of the elements of A_{ij}, is PD for all $\beta > 0$

As mentioned earlier, an important problem with using kernel functions is that optimal parameters play the key role in the performance of kernels, i.e. how well the data becomes separable [12]. Improperly chosen parameters may give drastically poor results, so the parameters must be carefully chosen.

B. Behavior of Sigmoid Kernel for different parameter values

For the sake of specificity, in this paper we describe the proposed approach in the context of the sigmoid kernel, which is commonly used in kernel methods such as SVMs. We underline that the proposed approach can also be applied to other kernels. The sigmoid kernel is defined as:

$$K(x, x') = \tanh(\alpha x^T x' + \vartheta). \tag{3}$$

where α is a scaling parameter of the input data and ϑ is a shifting parameter. The sigmoid kernel is widely used even though it is known that that the kernel may not be PSD (this observation also holds true for the other kernel types). The sigmoid kernel is valid when its parameters take values in certain ranges. The practical viability of the sigmoid kernel was investigated in [16] whose conclusions can be summarized as under (see [16] for the proofs):

1. If $\alpha \geq 0$ and $\vartheta < 0$, then the kernel matrix is CPD.
2. If the sigmoid kernel is PD, then $\alpha \geq 0$ and $\vartheta \geq 0$. The inverse does not hold, so the practical viability is not clear.
3. If $\alpha < 0$ and $\vartheta > 0$, then the data in the feature space may not be separable using the kernel. Thus, this combination may not be a good choice.
4. Separability of the data is also not guaranteed if $\alpha < 0$ and $\vartheta < 0$.

Based on the above observations, we use the condition 1 to guide our proposed optimal parameter estimation method in order to obtain a valid sigmoid kernel and the corresponding kernel matrix.

IV. KERNEL PARAMETER ESTIMATION AND CLUSTERING USING KERNEL *K*-MEANS

A. Kernel K-means and Kernel Matrix

Clustering involves partition a set of data points. Given a set of data points $S = \{x_1, x_2, ..., x_n\}$, in clustering we seek an assignment of the form:

$$f : S \to \{1, 2, ..., k\}, \tag{4}$$

where k indexes the clusters. A large number of clustering algorithms exist of which k-means clustering is a popular method. Kernel k-means is k-means clustering algorithm conducted in a feature space that overcomes the limitations of the classical k-means algorithm on linearly inseparable data. Before clustering, data points are mapped to a higher-dimensional feature space by a nonlinear function and then kernel k-means partitions the data points in the feature space. Each data point is assigned to the nearest centroid by a clustering function f defined on S as

$$f(x_i) = \arg\min_{1 \leq j \leq k} \| \Phi(x_i) - \mu_j \|, \tag{5}$$

where μ_j are cluster centroids. At each iteration, both f and the cluster centers are updated until a convergence is achieved at a solution that satisfies the optimization criterion

$$\arg\min_{f} \sum_{i,j:f_i=f_j} \| \Phi(x_i) - \Phi(x_j) \|^2 . \quad (6)$$

B. Line Search Sequential Quadratic Programming

We use the sigmoid kernel and seek to satisfy the two constraints from condition 1 in Section III, i.e. $\alpha > 0$ and $\vartheta < 0$ and sufficiently small. The goal of the parameter estimation is to find parameter values that enable the (sigmoid) kernel to be CPD *and* maximize the variance of the kernel matrix which is the input to the kernel *k*-means clustering algorithm. In the following, we make two critical observations and describe our formulation for maximizing the variance of the kernel matrix based on them.

Observation 1: Maximizing the trace of the kernel matrix leads to maximizing eigenvalues of the kernel matrix.

Observation 2: Maximizing the variance of the kernel matrix in the feature space requires maximizing the trace of the kernel matrix because the leading eigenvalues of the kernel matrix measure the variance along the principal components in the feature space

Based on the above observations, the objective function and constraints to find a CPD sigmoid kernel can be formulated as follows:

$$\text{maximize } Trace(M), \quad \text{subject to } \alpha > 0 \text{ and } \vartheta < 0, \quad (7)$$

where *Trace*(*M*) is the trace of the kernel matrix *M*. Since The two unknown parameters α and ϑ of the hyperbolic tangent are nonlinear parameters and the two constraints in (7) are imposed on these parameters, there is no analytical solution to find the unknown parameters that satisfy the constraints.

We therefore apply a form of constrained nonlinear programming called sequential quadratic programming (SQP) to solve the nonlinear optimization described above. Specifically, we use the line search SQP (LS-SQP), [19] to estimate the two parameters so that the sigmoid kernel becomes CPD. The line search method finds a search direction along which the objective function is reduced and then computes a step size that decides how far to move along that direction. At each iteration of LS-SQP, the solution is updated as

$$x_{k+1} = x_k + a_k p_k , \quad (8)$$

where p_k is the search direction and a_k is the step size at k^{th} iteration. The search direction must be a descent direction in order to find a local minimum. In this framework, SQP solves a quadratic programming (QP) sub-problem subject to a linearization of constraints to define the search direction. The general form of SQP is given as under:

$$\text{minimize } f(x) \quad (9)$$

$$\text{subject to } a_m^T x = b_m \quad (m \in \mathcal{E}), \quad (10)$$

$$a_m^T x \geq b_m \quad (m \in \mathcal{I}), \quad (11)$$

where x is a set of unknown parameters, $f(x)$ is an objective function, a_m are elements of a Jacobian matrix for the equality and inequality constraints, b_m are elements of a corresponding bound vector, and $\mathcal{E}$ and $\mathcal{I}$ are finite sets of indices m to which equality and inequality constraints on a_m apply.

The objective function and constraints from (7) are represented in the form for SQP as described above by Eqs. (9) – (11). The objective is to maximize trace of the kernel matrix M to be constructed by using the sigmoid function with the estimated $x = (\alpha, \vartheta)^T$. Accounting for the general form of the SQP, which minimizes the objective function, we get the following objective function and constraints:

$$\text{minimize } (-Trace(KM)) \text{ , subject to } x \geq B_l \text{ and } x \leq B_u , \quad (12)$$

where B_l is a lower bound vector and B_u is an upper bound vector. Since the constraints in (12) force α to be nonnegative, the lower bound of α is a very small positive number and the upper bound is ∞ . The lower bound of ϑ is $-\infty$ and the upper bound is a very large negative number. With these bound values, the Jacobian matrix A and the bound vector B for (10) and (11) are

$$A^T = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (22) \qquad B = \begin{pmatrix} B_l \\ -B_u \end{pmatrix}. \quad (13)$$

Thus, the inequality constraints can be represented as $A^T x \geq B$. The Hessian for the QP subproblem is updated by quasi-Newton approximation. We use the Broyden-Fletcher-Goldfarb-Shanno method [19] for this purpose. At each SQP iteration, the merit function shown in Eq. (14) [20] was used as a penalty function in finding the appropriate step size α_k .

$$\psi(x) = f(x) + \sum_{i=1}^{m_e} r_i c_i(x) + \sum_{i=m_e+1}^{m} r_i \max\{0, c_i(x)\} , \quad (14)$$

In Eq.(14), m_e is the last index of the equality constraints, m is the last index of the inequality constraints, and r_i is the penalty parameter.

V. Experiments

The proposed method is applied to the yeast gene microarray data [25] and compared with five state-of-the-art linear and nonlinear dimensionality reduction methods. These methods are: (1) principal component analysis (PCA), (2) correspondence analysis (CA), (3) Multidimensional Scaling (MDS), (4) Locally Linear Embedding (LLE), and (5) Isometric Feature Mapping (Isomap). After the input data is represented in a low-dimensional space using the aforementioned methods, the clusters are determined using k-means clustering and the accuracy of the clusters compared across all the methods.

A. Data

The input data is a well-known gene expression microarray data set called *Saccharomyces cerevisiae* [25], which is publicly available (http://genome-www.stanford.edu/cellcycle). For each gene, the ratio of expression was measured in each time point and the magnitude of the ratio of expression was color-coded. The genes were synchronized by four synchronization methods, i.e. α-factor, CDC15, CDC28 and elutriation. Spellman *et al.*

classified the genes by their pattern of expression into five groups termed G1, S, G2, M and M/G1. These genes were sorted according to the phase of expression, i.e. M/G1, G1, S, G2 and then M. Each row in the input data represents each gene and each column represents a time point in an experiment. There are 113 genes in M/G1 group, 300 genes in G1 group, 71 genes in S group, 121 genes in G2 group and 195 genes in M group. The expression patterns of the first gene in G2 group and the last gene in the M group are virtually same but the genes are separated into the two different groups [25]. These genes are challenging to cluster because such similar expression patterns cause the non-trivial distribution and non-linear separability of the data.

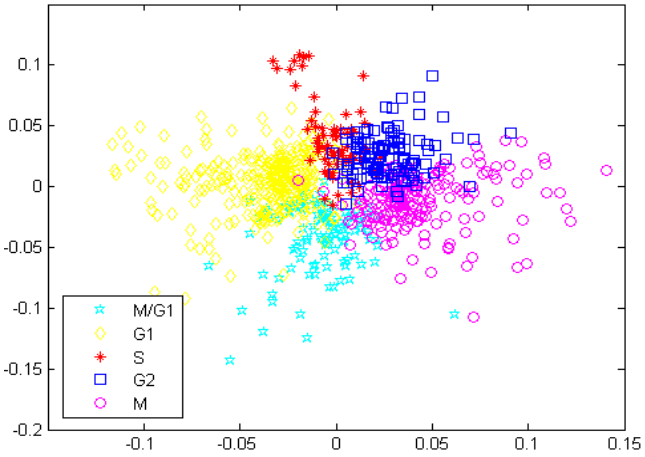

Figure 1. Two-dimensional gene representation by correspondence analysis. The five classes are of the reference classes in (Spellman 1998).

To have an initial insight about the data, we employ correspondence analysis (CA). CA is a statistical approach that represents the structure of the original high-dimensional data in two dimensions. We use CA because it was developed specifically for providing the summary of multivariate data sets in lower dimensions [9]. CA has been applied to S. *cerevisiae* data [8, 22] to help understand the relationship of data. We use the particular CA method proposed in [22]. Figure 1 shows the gene distribution in two-dimensions by CA. The figure shows that the five clusters are closely located and the clusters highly intersect with each other. Intuitively, it is hard to find a dimension that maximizes the separability of the clusters.

B. Clustering Accuracy

The Clustering accuracy of our proposed method was compared with the clustering accuracies by applying the standard *k*-means algorithm to the low-dimensional representations obtained by PCA, CA, MDS LLE and Isomap. The kernel *k*-means algorithm used by us was implemented based on the pseudo code in [24]. We used the publicly available source code for LLE (http://www.cs.nyu.edu/~roweis/lle/code.html) and Isomap (http://isomap.stanford.edu).

The true number of clusters (five) was used as a parameter for the *k*–means clustering. Parameters for the other methods were determined through empirical testing. The evaluation metric used by us was clustering accuracy by identifying incorrectly clustered genes using Spellman *et al.*'s classification from [25] as reference. For MDS, the dimension of the highest clustering accuracy was chosen as the best dimension for the data. For LLE and Isomap, the combination of the dimension and the number of the nearest neighbors with the highest clustering accuracy was selected as the best combination of the two parameters for the data.

Table I. Clustering accuracy (%). KKmeans: kernel k-means with line search SQP. Agglom: agglomerative hierarchical clustering, Divisive: divisive hierarchical clustering.

Clustering	PCA	MDS	LLE	Isomap	KKmeans
K-means	65.38	64.25	67.88	67.75	77.63
Agglom	37.75	37.50	38.38	39.00	□
Divisive	37.88	42.25	38.63	60.63	□

Table II. Number of dimensions and number of nearest neighbors of the highest clustering accuracy, clustering method: *k*-means

Clustering	Parameter	MDS	LLE	Isomap
K-means	Dimension	3	5	2
	k-neighbors	□	6	5
Agglom	Dimension	3	4	2
	k-neighbors	□	21	11
Divisive	Dimension	3	3	3
	k-neighbors	□	6	13

Table I shows the highest clustering accuracy of each method. In addition to the *k*-means algorithm, the two hierarchical clustering algorithms (agglomerative and divisive) are compared. As the table shows, our proposed method achieved the highest clustering accuracy among the six methods using *k*-means clustering method. Other five methods showed the similar clustering accuracy. The two hierarchical clustering methods showed the lower clustering accuracies than the *k*-means clustering. Table II shows the pair of parameters of the highest clustering accuracy in Table I for MDS, LLE and Isomap.

Figure 2 shows the five classes identified by Spellman *et al.* as reference and the clusters found by each of the five methods. The 800 genes were sorted in the order of M/G1, G1, S, G2 and M groups and the five classes were color-coded by the five colors, i.e. cyan for M/G1, yellow for G1, red for S, blue for G2 and magenta for M. Each vertical line stands for each gene and the lines of the same color indicate that those genes were assigned to the same cluster. There are three interesting observations. First, the four methods (PCA, MDS, LLE and Isomap) failed in separating S and G2 groups. But our method (Figure 4(f)) identified the two clusters better than the four methods even though the two clusters were not exactly separated. We further investigate the S and G2 clusters in detail in the next section. Second, LLE failed in separating M/G1 and M groups (Figure 4(d)). However, LLE identified the genes in M group the most clearly compared to the other four methods. These facts imply that LLE overfitted the least among the five methods but it made LLE lose the discriminative power in this data set. Third, the common phenomenon in the clusters found by the five methods is that genes that were in groups that were immediate neighbors affected each other, e.g. some of the

genes in G1 group were assigned to S group according to the distributions of red vertical lines (Figure 2(b) – (f)).

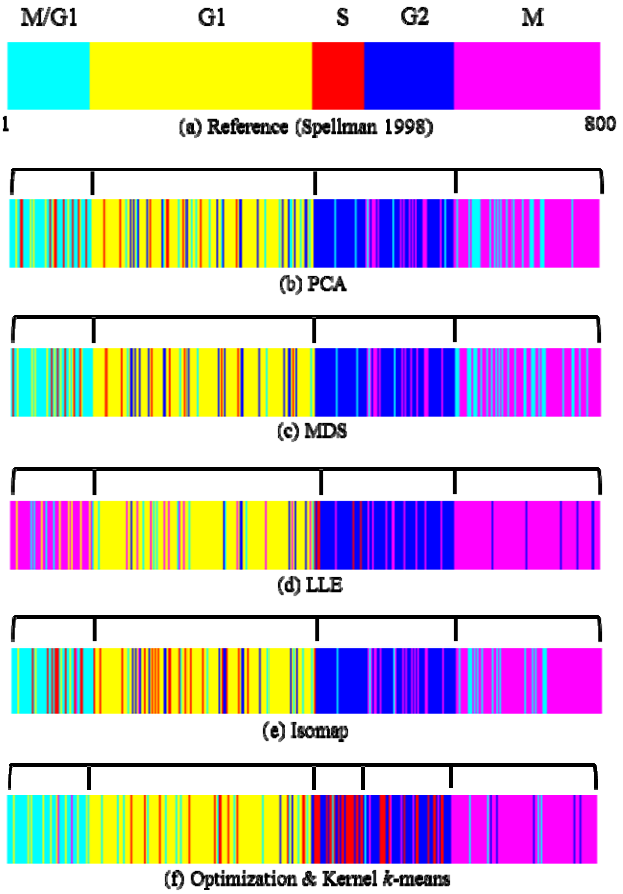

Figure 2. Clusters found by the five methods. (a) five classes identified by Spellman *et al*. (Spellman 1998) (b) - (f) clusters identified by the five methods.

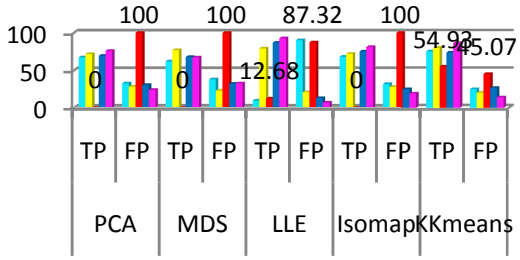

Figure 3. Percentage (%) of true positive (TPs) and false positive (FPs) in each reference group. Clustering algorithm: *k*-means. Cyan: M/G1, yellow: G1, red: S, blue: G2, magenta: M

Figure 3 is the histogram representation of the data from Figure 2. Each bar in Figure 3 indicates the percentage of true positive (TP) and false positive (FP) in each cluster. The percentage of TP in the S group by kernel *k*-means is the highest among the five groups and the percentage of FP in the S group by the same method is the lowest. The common observation applicable to all the four methods (PCA, MDS, LLE and Isomap) is that the S group was poorly delineated.

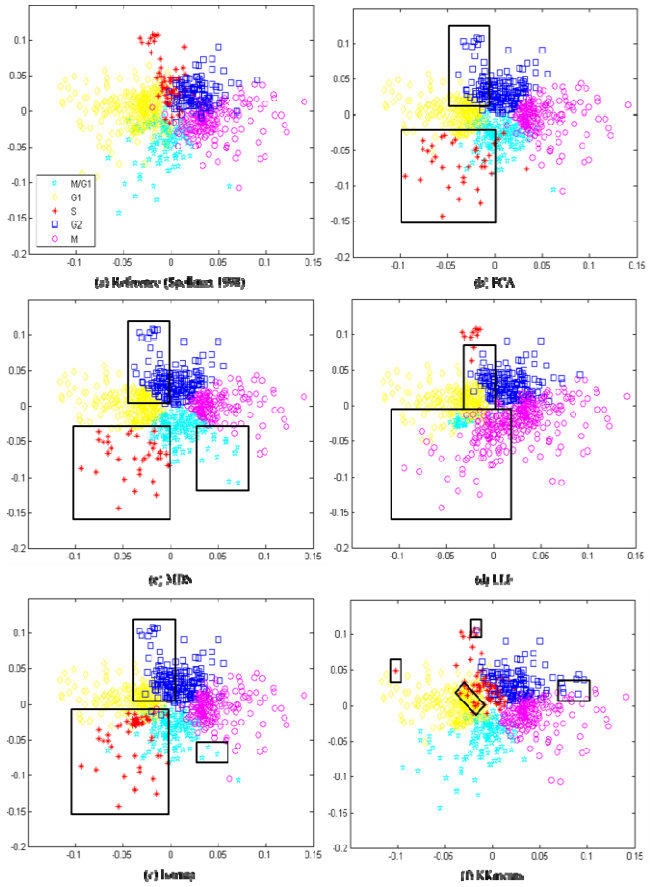

Figure 4. Gene distributions found by each method using two-dimensional CA representation. Genes in black rectangles are incorrectly clustered genes based on the reference clusters in [25]. X-axis: dimension 1, Y-axis: dimension 2 (a) Gene distribution based on the reference classification in (Spellman 1998)

C. A closer look at the S and G2 clusters

The genes in S and G2 groups turned out to be the most difficult to accurately cluster. We look into these two groups in detail in the following. Figure 4 is the two-dimensional representation of the 800 genes obtained by using CA. The color scheme used in Figure 2 is used here to distinguish the five classes of the genes discovered as identified in [25]. The black rectangles indicate the genes that were incorrectly clustered. Overall, the five clusters are not clearly separated and neighboring clusters intersect each other. The number of genes in the S group is the smallest and some of the genes from this group overlap with the two neighboring groups, G1 and G2. It is clear from this figure that the four methods failed to identify the rest of the non-overlapping genes in S group from the two neighboring groups (G1 and G2) but the proposed method identified most of the genes (Figure 4(f)) from the two neighboring groups correctly.

VI. Discussions and Conclusions

In this paper we have proposed the use of conditionally positive definite kernels coupled with line search sequential quadratic programming for optimal kernel parameter estimation. Compared to manifold learning techniques, the proposed method does not require user assignment of complex parameters such as the number of nearest neighbors or the number of dimensions required in the mapping. Finally, the approach is computationally efficient and leads to optimal parameter estimation.

Acknowledgment

This research was supported in part by the National Science Foundation grant IIS- 0644418.

References

[1] A.E. Bayá and P.M. Granitto, "Clustering gene expression data with a penalized graph-based metric," BMC Bioinformatics, 12:2, 2011.

[2] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Scholkopf, and G. Ratsch, "Support vector Machines and Kernels for Computational Biology", PLoS Computational Biology, 4(10), 2008

[3] L. Cayton, "Algorithms for manifold learning," Technical report CS2008-0923. University of California, San Diego, 2008.

[4] Y. Cheng and G.M. Church, "Biclustering of Expression Data," Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology, pp. 93–103, 2000.

[5] E.J. Cooke, R.S. Savage, P.D. Kirk, R. Darkins and D.L. Wild, "Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements," BMC Bioinformatics, 12:399, 2011.

[6] M.B. Eisen and P.O. Brown, "DNA arrays for analysis of gene expression," Methods Enzymol, vol. 303, pp. 170–205, 1999.

[7] G.E. Fasshauer, "Positive definite kernels: past, present and future," Dolomites Research Notes on Approximation, vol. 4, pp. 21–63, 2011.

[8] K. Fellenberg, N.C. Hauser, B. Brors, A. Neutzner, J.D. Hoheisel, and M. Vingron, "Correspondence analysis applied to microarray data," Proc. Natl Acad Sci USA, vol. 98, no. 19, pp. 10781–10786, Epub., 2001.

[9] M. Greenacre, "Correspondence Analysis in Practice. Chapman & Hall/CRC," Taylor & Francis Group, Boca Raton, 2007.

[10] A. Guichardet. Symmetric Hilbert Spaces and Related Topics. Springer, 1972.

[11] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2006.

[12] T. Howley and M.G. Madden, "The Genetic Kernel Support Vector Machine: Description and Evaluation," Artificial Intelligence Review, vol. 24, Issue: 3-4, Kluwer Academic Publishers, 2005, pp. 379-395, doi: 10.1007/s10462-005-9009-3.

[13] C. Huttenhower, A.I. Flamholz, J.N. Landis, S. Sahi, C.L. Myers, K.L. Olszewski, M.A. Hibbs, N.O. Siemers, O.G. Troyanskaya and H.A. Coller, "Nearest Neighbor Networks: clustering expression data based on gene neighborhoods," BMC Bioinformatics, 8:250, 2007.

[14] C.S. Kim, C.S. Bae and H.J. Tcha, "A phase synchronization clustering algorithm for identifying interesting groups of genes from cell cycle expression data," BMC Bioinformatics, 9:56, 2008.

[15] H. Lee, K. Yao, O. Okpani, A. Nakano and I. Ershaghi, "Hybrid Constrained Nonlinear Optimization to Infer Injector-Producer Relationships in Oil Fields," Int'l J. Computational Science, vol. 4, no. 1, pp. 1–22, 2010.

[16] H. Lin and C. Lin, "A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods," Neural Computation, pp. 1–32, 2003.

[17] S.C. Madeira and A.L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey," IEEE Trans. Computational Biology and Bioinformatics, vol. 1, no. 1, 2004.

[18] S.K. Ng, G.J. McLachlan, K. Wang, L. B.-T. Jones and S.-W. Ng, "A Mixture model with random effects components for clustering correlated gene-expression profile," Bioinformatics, vol. 22, no. 14, pp. 1745–1752, 2006.

[19] J. Nocedal and S.J. Wright, Numerical Optimization, Springer-Verlag New York Berlin Heidelberg, 1999.

[20] J.D. Powell, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations, Numerical Analysis," vol. 630, pp. 144–157, 1978.

[21] S.T. Roweis and L.K Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, vol. 290, pp. 2323–2326, 2000.

[22] A. Sasho, S. Zhu and R. Singh, "Identification and Analysis of Cell Cycle Phase Genes by Clustering In Correspondence Subspaces," Int'l Conf. Advances in Computing and Communications, vol. 190, Part 4, pp. 340–350, 2011.

[23] B. Schölkopf and A.J. Smola, Learning with Kernels. The MIT Press, 2002.

[24] J. Shawe-Taylor and N. Cristianini, Kernel Methods for Pattern Analysis. Cambridge University Press, 2004.

[25] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein and B. Futcher, "Comprehensive Identification of Cell Cycle–regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization," Molecular Biology of the Cell, vol. 9, pp. 3273–3297, 1998.

[26] J.B. Tenenbaum, V. Silva and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, vol. 290, pp. 2319–2323, 2000.

[27] H. Xiong and X-W Chen, "Data Dependent Kernel Machines for Micorarray Data Classification", IEEE/ACM Transactions on Computational Biology and Bioinformatics, 4(4), 2007