Journal of Machine Learning Research 4 (2003) 67-81 Submitted 11/02; Revised 3/02; Published 5/03

The *em* Algorithm for Kernel Matrix Completion with Auxiliary Data

Koji Tsuda KOJI.TSUDA@TUEBINGEN.MPG.DE
Max Planck Institute for Biological Cybernetics
72076 Tübingen, Germany; and
AIST Computational Biology Research Center
Tokyo, 135-0064, Japan

Shotaro Akaho S.AKAHO@AIST.GO.JP
AIST Neuroscience Research Institute
Tsukuba, 305-8568, Japan

Kiyoshi Asai ASAI@K.U-TOKYO.AC.JP
Department of Computational Biology
Graduate School of Frontier Science
University of Tokyo
Kashiwa, 277-8562, Japan; and
AIST Computational Biology Research Center
Tokyo, 135-0064, Japan

Editor: Chris Williams

Abstract

In biological data, it is often the case that observed data are available only for a subset of samples. When a kernel matrix is derived from such data, we have to leave the entries for unavailable samples as missing. In this paper, the missing entries are completed by exploiting an auxiliary kernel matrix derived from another information source. The parametric model of kernel matrices is created as a set of spectral variants of the auxiliary kernel matrix, and the missing entries are estimated by fitting this model to the existing entries. For model fitting, we adopt the *em* algorithm (distinguished from the EM algorithm of Dempster et al., 1977) based on the information geometry of positive definite matrices. We will report promising results on bacteria clustering experiments using two marker sequences: 16S and gyrB.

Keywords: Information geometry, *em* algorithm, Kernel matrix completion, Bacteria clustering

1. Introduction

In kernel machines such as support vector machines (SVM) (Schölkopf and Smola, 2002), objects are represented as a kernel matrix, where n objects are represented as an $n \times n$ positive semidefinite matrix. Essentially the (i, j) entry of the kernel matrix describes the similarity between i-th and j-th objects. Due to positive semidefiniteness, the objects can be embedded as n points in an Euclidean feature space such that the inner product between two points equals to the corresponding entry of kernel matrix. This property enables us to apply diverse learning methods (for example, SVM or kernel PCA) without explicitly constructing a feature space (Schölkopf and Smola, 2002).

Biological data such as amino acid sequences, gene expression arrays and phylogenetic profiles are derived from expensive experiments (Brown, 2002). Typically initial experimental measurements are so noisy that they cannot be given to learning machines directly. Since high quality data are created by extensive work of human experts, it is often the case that good data are available only for a subset of samples. When a kernel matrix is derived from such incomplete data, we have to leave the entries for unavailable samples as *missing*. We call such a matrix an “incomplete matrix”. Our aim is to estimate the missing entries, but it is obviously impossible without additional information. So we make use of a *parametric model* of admissible matrices, and estimate missing entries by fitting the model to existing entries.

In this scheme, it is important to define a parametric model appropriately. For example, Graepel (2002) used the set of all positive definite matrices as a model. Although this model worked well when only a few entries are missing, this model is too general for our cases where whole columns and rows are missing. Thus we need another information source for constructing a parametric model. Fortunately, in biological data, it is common that one object is described by two or more representations. For example, genes are represented by gene networks and gene expression arrays at the same time (Vert and Kanehisa, 2003). Also a bacterium is represented by several marker sequences (Yamamoto et al., 2000). In this paper, we assume that *auxiliary data* are available from another information source. From the auxiliary data, a complete kernel matrix called the *auxiliary matrix* is derived and a parametric model is created by giving perturbations to this matrix. When creating a parametric model of admissible matrices from an auxiliary matrix, one typical way is to define the parametric model as all *spectral variants*[1] which have the same eigenvectors but different eigenvalues (Cristianini et al., 2002). When several auxiliary matrices are available, the weighted sum of these matrices would be a good parametric model as well (Lanckriet et al., 2002).

In order to fit a parametric model, the distance between two matrices has to be determined. A common way is to define the Euclidean distance between matrices (for example, the Frobenius norm) and make use of the Euclidean geometry. Recently Vert and Kanehisa (2003) tackled with the incomplete matrix approximation problem by means of kernel CCA. Also Cristianini et al. (2002) proposed a similarity measure called "alignment", which is basically the cosine between two matrices. In contrast that their methods are based on the Euclidean geometry, this paper will follow an alternative way: we will define the Kullback-Leibler (KL) divergence between two kernel matrices and make use of the Riemannian information geometry (Ohara et al., 1996). The KL divergence is derived by relating a kernel matrix to a covariance matrix of Gaussian distribution. The primal advantage is that the KL divergence allows us to use the *em* algorithm (Amari, 1995) to approximate an incomplete kernel matrix. The *e* and *m* steps are formulated as convex programming problems, and moreover they can be solved analytically when spectral variants are used as a parametric model.

We performed bacteria clustering experiments using two marker sequences: 16S and gyrB (Yamamoto et al., 2000). We derived the incomplete and auxiliary kernel matrices from gyrB and 16S, respectively. As a result, even when 50% of columns/rows are missing, the clustering performance of the completed matrix was better than that of the auxiliary matrix, which illustrates the effectiveness of our approach in real world problems.

This paper is organized as follows: Section 2 introduces the information geometry to the space of positive definite matrices. Based on geometric concepts, the *em* algorithm for matrix completion is presented in Section 3, where detailed computations are deferred in Section 4. In Section 5, the matrix completion problem is formulated as statistical inference and the equivalence between the *em* and EM algorithms (Dempster et al., 1977) is shown. Then the bacteria clustering experiment is described in Section 6. After seeking for possible extensions in Section 7, we conclude the paper in Section 8.

2. Information Geometry of Positive Definite Matrices

We first explain how to introduce the information geometry (Amari and Nagaoka, 2001) in the space of positive definite matrices. Let us define the set of all $d \times d$ positive definite matrices as $\mathcal{P}$. The first step is to relate a $d \times d$ positive definite matrix $P \in \mathcal{P}$ to the Gaussian distribution with mean 0 and covariance matrix P:

$$p(x|P) = \frac{1}{(2\pi)^{d/2}|P|^{1/2}} \exp(-\frac{1}{2}x^\top P^{-1}x). \tag{1}$$

1. The spectral variants model is so simple and has favorable geometrical properties as will be discussed in Section 3. Moreover, the complexity of this model has recently been analyzed by Bousquet and Herrmann (2003). For simplicity, we will exclusively focus on this model in the rest of this paper, but there remains much to explore about matrix models.

It is well known that the Gaussian distribution belongs to the exponential family. The canonical form of an exponential family distribution is written as

$$p(x|\theta) = \exp(\theta^\top r(x) - \psi(\theta)),$$

where $r(x)$ is the vector of sufficient statistics, θ is the natural parameter and $\psi(\theta)$ is the normalization factor. When (1) is rewritten in the canonical form, we have the sufficient statistics as

$$r(x) = -\left(\frac{1}{2}x_1^2, \dots, \frac{1}{2}x_d^2, x_1x_2, \dots, x_{d-1}x_d\right)^\top,$$

and the natural parameter as

$$\theta = \left([P^{-1}]_{11}, \dots, [P^{-1}]_{dd}, [P^{-1}]_{12}, \dots, [P^{-1}]_{d-1,d}\right)^\top,$$

where $[M]_{ij}$ denotes the (i,j) entry of matrix M. The natural parameter θ provides a coordinate system to specify a positive definite matrix P, which is called the θ-coordinate system (or the e-coordinate system). On the other hand, there is an alternative representation for the exponential family. Let us define the mean of $r_i(x)$ as η_i: For example, when $r_i(x) = x_s x_t$,

$$\eta_i = \int x_s x_t p(x|\theta) dx = P_{st}.$$

This new set of parameters η_i provides another coordinate system, called η-coordinate system (or the m-coordinate system):

$$\eta = (P_{11}, \dots, P_{dd}, P_{12}, \dots, P_{d-1,d})^\top.$$

Let us consider the following curve $\theta(t)$ connecting two points θ_1 and θ_2 linearly in θ coordinates:

$$\theta(t) = t(\theta_2 - \theta_1) + \theta_1.$$

When written is the matrix form, this reads

$$P^{-1}(t) = t(P_2^{-1} - P_1^{-1}) + P_1^{-1}.$$

This curve is regarded as a straight line from the exponential viewpoint and is called an exponential geodesic or e-geodesic. In particular, each coordinate curve $\theta_i = t$, $\theta_j = c_j \ \ (j \neq i)$ is an e-geodesic. When the e-geodesic between any two points in a manifold $\mathcal{S} \subseteq \mathcal{P}$ is included in $\mathcal{S}$, the manifold $\mathcal{S}$ is said to be e-flat. On the other hand, the mixture geodesic or m-geodesic is defined as

$$\eta(t) = t(\eta_2 - \eta_1) + \eta_1.$$

In the matrix form, this reads

$$P(t) = t(P_2 - P_1) + P_1.$$

When the m-geodesic between any two points in $\mathcal{S}$ is included in $\mathcal{S}$, the manifold $\mathcal{S}$ is said to be m-flat.

In information geometry, the distance between probability distributions is defined as the Kullback-Leibler divergence (Amari and Nagaoka, 2001):

$$KL(p,q) = \int p(x) \log \frac{p(x)}{q(x)} dx.$$

By relating a positive definite matrix to the covariance matrix of Gaussian (1), we have the Kullback-Leibler (KL) divergence for two matrices P, Q:

$$KL(P,Q) = \mathrm{tr}(Q^{-1}P) + \log\det Q - \log\det P - d.$$

With respect to a manifold $\mathcal{S} \subseteq \mathcal{P}$ and a point $P \in \mathcal{P}$, the projection from P to $\mathcal{S}$ is defined as the point in $\mathcal{S}$ closest to P. Since the KL divergence is asymmetric, there are two kinds of projection:

- e-projection: $Q^* = \mathrm{argmin}_{Q\in\mathcal{S}} KL(Q,P)$.
- m-projection: $Q^* = \mathrm{argmin}_{Q\in\mathcal{S}} KL(P,Q)$.

It is proved that the m-projection to an e-flat submanifold is unique, and e-projection to an m-flat manifold is unique (Amari and Nagaoka, 2001). This uniqueness property means that the corresponding optimization problem is convex and so the global optimal solution is easily obtained by any reasonable method.

3. Completion of an Incomplete Kernel Matrix

In this section, we describe the *em* algorithm to complete an incomplete kernel matrix. Let $x_1,\dots,x_\ell \in \mathcal{X}$ be the set of samples of interest. In supervised learning cases, this set includes both training and test sets, thus we are considering the transductive setting (Vapnik, 1998). Let us assume that the data is available for the first n samples, and unavailable for the remaining $m := \ell - n$ samples. Denote by K_I an $n\times n$ kernel matrix, which is derived from the data for the first n samples. Then, an incomplete kernel matrix is described as

$$D = \begin{pmatrix} K_I & D_{vh} \\ D_{vh}^\top & D_{hh} \end{pmatrix}, \qquad (2)$$

where D_{vh} is an $n\times m$ matrix and D_{hh} is an $m\times m$ symmetric matrix. Since D has missing entries, it cannot be presented as a point in $\mathcal{P}$. Instead, all the possible kernel matrices form a manifold

$$\mathcal{D} = \{D \mid D_{vh} \in \Re^{n\times m},\ D_{hh}\in\Re^{m\times m},\ D_{hh} = D_{hh}^\top,\ D \succ 0\},$$

where $D \succ 0$ means that D is positive definite. We call it the *data manifold* as in the conventional EM algorithm (Ikeda et al., 1999). It is easy to verify that $\mathcal{D}$ is an m-flat manifold; hence, the e-projection to $\mathcal{D}$ is unique.

Next let us define the parametric model to approximate D. Here the model is derived as the spectral variants of K_B, which is an $\ell\times\ell$ auxiliary kernel matrix. Let us define λ_i and v_i as the i-th eigenvalue and eigenvector of K_B, respectively.[2] Then K_B is decomposed as follows:

$$K_B = \sum_{i=1}^{\ell} \lambda_i v_i v_i^\top .$$

Define

$$M_i = v_i v_i^\top, \qquad (3)$$

then all the spectral variants are represented as[3]

$$\mathcal{M} = \{M \mid M = \sum_{j=1}^{\ell} \beta_j M_j,\ \beta\in\Re^\ell,\ M\succ 0\}.$$

We call it the model manifold (Ikeda et al., 1999). Since the eigenvectors $\{v_i\}_{i=1}^{\ell}$ are orthonormal, $\mathcal{M}$ can be reparametrized as follows:

$$\mathcal{M} = \{M \mid M = (\sum_{j=1}^{\ell} b_j M_j)^{-1},\ b\in\Re^\ell,\ M\succ 0\}, \qquad (4)$$

where $b_j = 1/\beta_j$. It is easily seen that the manifold $\mathcal{M}$ is e-flat and m-flat at the same time. Such a manifold is called dually-flat.

2. When ℓ is large, the minor eigenvalues of K_B may be extremely small and the numerical computation of eigenvectors could be instable. For stabilizing eigenvectors, it is effective to add a small positive constant to all the diagonal elements.
3. In some cases, it would also be good to construct $\mathcal{M}$ only with major eigenvectors, i.e. $M = \sum_{i=1}^{k} \beta_j M_j + \beta_0 I$, where $k < \ell, \beta_0 \in \Re_+$ and I is the unit matrix.

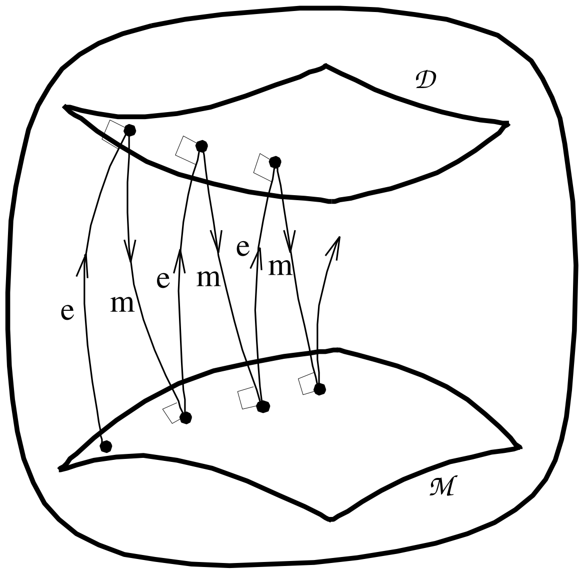

Figure 1: Information geometric picture of the *em* algorithm. The data manifold $\mathcal{D}$ corresponds to the set of all completed matrices, whereas the model manifold $\mathcal{M}$ corresponds to the set of all spectral variants of an auxiliary matrix. The nearest points are found by gradually minimizing the KL divergence by repeating *e* and *m* projections.

Our approximation problem is formulated as finding the nearest points in two manifolds: Find $D \in \mathcal{D}$ and $M \in \mathcal{M}$ to minimize $KL(D,M)$. In geometric terms, this problem is to find the nearest points between *e*-flat and *m*-flat manifolds. It is well known that such a problem is solved by an alternating procedure called the *em* algorithm (Amari, 1995). The *em* algorithm gradually minimizes the KL divergence by repeating *e*-step and *m*-step alternately (Figure 1).

In the *e*-step, the following optimization problem is solved with fixing M: Find $D \in \mathcal{D}$ that minimizes $KL(D,M)$. This is rewritten as follows: Find D_{vh} and D_{hh} that minimize

$$L_e = \mathrm{tr}(DM^{-1}) - \log\det D, \tag{5}$$

subject to the constraint that $D \succ 0$. As indicated by information geometry, this is a convex problem, which can readily be solved by any reasonable optimizer. Moreover the solution is obtained in a closed form: Let us partition M^{-1} as

$$M^{-1} = \begin{pmatrix} S_{vv} & S_{vh} \\ S_{vh}^\top & S_{hh} \end{pmatrix}. \tag{6}$$

The solution of (5) is described as

$$D_{vh} = -K_I S_{vh} S_{hh}^{-1}, \tag{7}$$

$$D_{hh} = S_{hh}^{-1} + S_{hh}^{-1} S_{vh} K_I S_{vh} S_{hh}^{-1}. \tag{8}$$

The derivation of (7) and (8) will be described in Section 4.1.

In the *m*-step, the following optimization problem is solved with fixing D: Find $M \in \mathcal{M}$ that minimizes $KL(D,M)$. This is rewritten as follows: Find $b \in \Re^\ell$ that minimizes

$$L_m = \sum_{j=1}^{\ell} b_j \,\mathrm{tr}(M_j D) - \log\det(\sum_{j=1}^{\ell} b_j M_j) \tag{9}$$

subject to the constraint that $M \succ 0$. When $\{M_j\}_{j=1}^{\ell}$ are defined as (3), the closed form solution of (9) is obtained as

$$b_i = 1/\operatorname{tr}(M_i D), \qquad i = 1,\ldots,\ell. \tag{10}$$

The derivation of (10) will be described in Section 4.2.

4. Computing Projections

This section presents the derivation of e and m-projections in detail.

4.1 e-projection

In the following, we will derive the e-projection of M as (7) and (8). The optimization problem (5) is firstly solved without regarding the constraint $D \succ 0$, and then the solution is verified to satisfy the constraint. It is well known that the log determinant of a partitioned matrix is described as

$$\log\det D = \log\det K_I + \log\det(D_{hh} - D_{vh}^\top K_I^{-1} D_{vh}).$$

Thus, L_e in (5) is rewritten as follows:

$$\begin{aligned} L_e &= \operatorname{tr}(DM^{-1}) - \log\det D \\ &= \operatorname{tr}(DM^{-1}) - \log\det K_I - \log\det(D_{hh} - D_{vh}^\top K_I^{-1} D_{vh}). \end{aligned}$$

When we partition M^{-1} as (6), it turns out that

$$L_e = \operatorname{tr}(K_I S_{vv}) + 2\operatorname{tr}(D_{vh} S_{vh}) + \operatorname{tr}(D_{hh} S_{hh}) - \log\det K_I - \log\det(D_{hh} - D_{vh}^\top K_I^{-1} D_{vh}). \tag{11}$$

The saddle point equation with respect to D_{hh} is obtained as

$$\frac{\partial L_e}{\partial D_{hh}} = S_{hh} - (D_{hh} - D_{vh}^\top K_I^{-1} D_{vh})^{-1}, \tag{12}$$

because $\frac{\partial}{\partial C}\log\det C = C^{-1}$ for any symmetric matrix C. Solving (12) with respect to D_{hh}, we have

$$D_{hh} = S_{hh}^{-1} + D_{vh}^\top K_I^{-1} D_{vh}. \tag{13}$$

Substituting (13) into (11), we have

$$L_e = \operatorname{tr}(K_I S_{vv}) + 2\operatorname{tr}(D_{vh} S_{vh}) + \operatorname{tr}(I + S_{hh} D_{vh}^\top K_I^{-1} D_{vh}) - \log\det K_I - \log\det S_{hh}.$$

Now the saddle point equation with respect to D_{vh} is obtained as

$$\frac{\partial L_e}{\partial D_{vh}} = 2S_{vh} + 2K_I^{-1} D_{vh} S_{hh} = 0.$$

Solving this equation, we have the solution (7) for D_{vh}. By substituting (7) into (13), we have the solution (8) for D_{hh}.

Next, let us verify that the solution D is positive definite ($D \succ 0$), that is,

$$c^\top D c > 0, \tag{14}$$

for any $c \in \Re^{\ell}$, $c \neq 0$. The left hand side of (14) is written as

$$c^\top D c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}^\top \begin{pmatrix} K_I & D_{vh} \\ D_{vh}^\top & D_{hh} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}.$$

By using the relation (13), we have the following:

$$c^\top D c = \|K_I^{1/2} c_1 - K_I^{-1/2} D_{vh} c_2\|^2 + c_2 S_{hh}^{-1} c_2. \tag{15}$$

When $M \succ 0$, its inverse is also positive definite ($M^{-1} \succ 0$). As S_{hh} is a submatrix of M^{-1} as shown in (6), we have $S_{hh}^{-1} \succ 0$. Accordingly, the second term of (15) is strictly positive. Since the first term is nonnegative, we finally prove that $c^\top D c > 0$.

4.2 *m*-projection

Here we will show the derivation of the *m*-projection (10). Firstly let us derive the solution b to minimize

$$L_m = \sum_{j=1}^{\ell} b_j \operatorname{tr}(M_j D) - \log\det(\sum_{j=1}^{\ell} b_j M_j).$$

without the constraint $M \succ 0$. Since $\partial \log\det Q^{-1}/\partial Q = Q$, the saddle point equations are described as

$$\operatorname{tr}(M_i(\sum_{j=1}^{\ell} b_j M_j)^{-1}) = \operatorname{tr}(M_i D), \qquad i = 1,\ldots,\ell. \tag{16}$$

Remembering that $M_j = v_j v_j^\top$, we have

$$(\sum_{j=1}^{\ell} b_j M_j)^{-1} = \sum_{j=1}^{\ell} \frac{1}{b_j} v_j v_j^\top.$$

The left hand side of (16) is rewritten as

$$\operatorname{tr}(M_i(\sum_{j=1}^{\ell} b_j M_j)^{-1}) = \operatorname{tr}(v_i v_i^\top \sum_{j=1}^{\ell} \frac{1}{b_j} v_j v_j^\top).$$

Since $v_i^\top v_j = 0$ $(i \neq j)$, we have the following:

$$\operatorname{tr}(M_i(\sum_{j=1}^{\ell} b_j M_j)^{-1}) = \operatorname{tr}(\frac{1}{b_i} v_i v_i^\top) = \frac{1}{b_i}.$$

Therefore, the solution of (16) is analytically obtained as

$$b_i = 1/\operatorname{tr}(M_i D) = \left(v_i^\top D v_i\right)^{-1}, \qquad i = 1,\ldots,\ell.$$

When D is positive definite, we have $v_i^\top D v_i > 0$ and thus $b_i > 0$. Since $\{b_i^{-1}\}_{i=1}^{\ell}$ correspond to the eigenvalues of M, M is proven to be positive definite.

We have shown that the *m*-projection is obtained analytically when the model manifold corresponds to spectral variants of a matrix. However, it is not always the case. For example, consider we have c auxiliary matrices $N_1,\ldots,N_c$ and the model manifold is constructed as the harmonic mixture of them:

$$\mathcal{M} = \{M \mid M = (\sum_{j=1}^{c} b_j N_j)^{-1},\ b \in \Re^c,\ M \succ 0\}. \tag{17}$$

This is an *e*-flat manifold, so the optimization problem is convex, but the analytical solvability depends on geometric properties of auxiliary matrices $\{N_i\}_{i=1}^{c}$(Ohara, 1999). We will briefly discuss this issue in the Appendix.

5. Relation to the EM algorithm

In statistical inference with missing data, the EM algorithm (Dempster et al., 1977) is commonly used. By posing the matrix completion problem as statistical inference, the EM algorithm can be applied, and—as shown later—it eventually leads to the same procedure. To describe the matrix completion problems, it is sufficient to define the geometry of matrices. So, in a sense, it is cumbersome to relate matrix completion to statistical concepts—random variables, observations and so on—and then use the geometry of probability distributions. Nevertheless it would be meaningful to rewrite our method in terms of statistical concepts for establishing connections to other methods such as the probabilistic PCA (Tipping and Bishop, 1999), the Gaussian processes (Williams, 1999, Csató and Opper, 2002) and the Bayesian covariance kernel (Seeger, 2002).

Let v and h be the n and m dimensional visible and hidden variables. From observed data,[4] the covariance matrix of v is known as

$$E_o[vv^\top] = K_I,$$

where E_o denotes the expectation with respect to observed data. However, we do not know the covariances $D_{vh} = E_o[vh^\top]$ and $D_{hh} = E_o[hh^\top]$. Our purpose is to obtain the maximum likelihood estimate of parameter b of the following Gaussian model:

$$p(v,h|b) = \frac{1}{(2\pi)^{d/2}|M|^{1/2}} \exp\left(-\frac{1}{2}\begin{pmatrix} v \\ h \end{pmatrix}^\top M^{-1}\begin{pmatrix} v \\ h \end{pmatrix}\right),$$

where M is described as (4).

In the course of maximum likelihood estimation, we have to estimate the observed covariances D_{vh} and D_{hh} in an appropriate way. The EM algorithm consists of the following two steps.

- E-Step: Fix b and update D_{vh} and D_{hh} by conditional expectation.
- M-Step: Fix D and update b by maximum likelihood estimation.

It is shown that the likelihood of observed data increases monotonically by repeating these two steps (Dempster et al., 1977).

The M-step maximizes the likelihood, which is easily seen to be equivalent to minimizing the KL divergence (Amari, 1995). So the M-step is equivalent to the m-step (9). However, the equivalence between E-step and e-step is not obvious, because the former is based on conditional expectation and the latter minimizes the KL divergence. In the E-step, the covariance matrices are computed from the conditional distribution described as

$$p(h|v,b) = \frac{1}{(2\pi)^{m/2}|S_{hh}^{-1}|^{1/2}} \exp\left(-\frac{1}{2}(h + S_{hh}^{-1}S_{vh}^\top v)^\top S_{hh}(h + S_{hh}^{-1}S_{vh}^\top v)\right),$$

where S matrices are derived as (6). Taking expectation with this distribution, we have

$$\begin{aligned} E_b[vh^\top \mid v] &= -vv^\top S_{vh}S_{hh}^{-1}, \\ E_b[hh^\top \mid v] &= S_{hh}^{-1} + S_{hh}^{-1}S_{vh}^\top vv^\top S_{vh}S_{hh}^{-1}. \end{aligned}$$

Then the covariance matrices are estimated as

$$\begin{aligned} D_{vh} = E_oE_b[vh^\top \mid v] &= -K_I S_{vh}S_{hh}^{-1}, \\ D_{hh} = E_oE_b[hh^\top \mid v] &= S_{hh}^{-1} + S_{hh}^{-1}S_{vh}^\top K_I S_{vh}S_{hh}^{-1}. \end{aligned}$$

Since these solutions are equivalent to (7) and (8), respectively, the E-step is shown to be equivalent to the e-step in this case. It is well known that the conditional expectation leads to the minimization of the KL divergence in most cases (Csiszár and Tusnady, 1984, Neal and Hinton, 1999). However, there are special cases that the EM and *em* algorithms are different (Amari, 1995).

4. In fact, we do not have observed data in any sense. However, we assumed them as a matter of form for relating *em* and EM.

6. Bacteria Classification Experiment

In this section, we perform unsupervised classification experiments for bacteria based on two marker sequences: 16S and gyrB. Basically we would like to identify the genus of a bacterium by means of extracted entities from the cell. It is known that several specific proteins and RNAs can be used for genus identification (Kasai et al., 1998). Among them, we especially focus on 16S rRNA and gyrase subunit B (gyrB) protein. 16S rRNA is an essential constituent in all living organisms, and the existence of many conserved regions in the rRNA genes allows the alignment of their sequences derived from distantly related organisms, while their variable regions are useful for the distinction of closely related organisms. GyrB is a type II DNA topoisomerase which is an enzyme that controls and modifies the topological states of DNA supercoils. This protein is known to be well preserved over evolutional history among bacterial organisms thus is supposed to be a better identifier than the traditional 16S rRNA (Kasai et al., 1998). Notice that 16S is represented as a nucleotide sequence with 4 symbols, and gyrB is an amino acid sequence with 20 symbols. Since gyrB has been found to be useful more recently than 16S (Yamamoto et al., 2000), gyrB sequences are available only for a limited number of bacteria. Thus, it is considered that gyrB is more "expensive" than 16S.

Our dataset has 52 bacteria of three genera (*Corynebacterium*: 10, *Mycobacterium*: 31, *Rhodococcus*: 11), each of which has both 16S and gyrB sequences. For simplicity, let us call these genera as class 1-3, respectively. For 16S and gyrB, we computed the second order count kernel, which is the dot product of bimer counts (Tsuda et al., 2002). Each kernel matrix is normalized such that the norm of each sample in the feature space becomes one. The kernel matrices of gyrB and 16S can be seen in Figure 2 (b) and (c), respectively. For reference, we show an ideal matrix as Figure 2 (a), which indicates the true classes. In our scenario, for a considerable number of bacteria, gyrB sequences are not available as in Figure 2 (d). We will complete the missing entries by the *em* algorithm with the spectral variants of the 16S matrix. When the *em* algorithm converges, we end up with two matrices: the *completed matrix* on data manifold $\mathcal{D}$ and the *estimated matrix* on model manifold $\mathcal{M}$. The completed and estimated matrices are shown in Figure 2 (e) and (f), respectively. These two matrices are in general not the same, because the two manifolds may not have intersection.

In order to evaluate the quality of completed and estimated matrices, K-means clustering is performed in the feature space of each kernel. Clustering is repeated 20 times with different initial centers selected randomly from the samples. We have chosen the clustering result with the smallest squared error. In evaluation, we use the mutual information score between the obtained clusters and the true classes (Vaithyanathan and Dom, 1999).[5] Let $U_1,\ldots,U_c$ be the obtained clusters and $T_1,\ldots,T_c$ be the true classes. Let p_{ij} be the fraction of the samples which belong to both U_i and T_j. Also let $p_{i.}$ and $p_{.j}$ be the fractions of the samples in U_i and T_j, respectively. The mutual information is defined as

$$\sum_{i=1}^{c}\sum_{j=1}^{c} p_{ij}\log\frac{p_{ij}}{p_{i.}p_{j.}}.$$

The clustering experiment is performed by randomly removing samples from the gyrB data. The ratio of missing samples is changed from 0% to 90%. The mutual information scores of completed and estimated matrices averaged over 20 trials are shown in Figures 3 and 4, respectively. Comparing the two matrices, the estimated matrix performed significantly worse than the complete matrix. It is because the completed matrix maintains existing entries unchanged, and so the class information in the gyrB matrix is well preserved. We especially focus on the comparison between the completed matrix and the 16S matrix, because there is no point in performing the *em* algorithm when the 16S matrix works better than the completed matrix. According to the plot, the mutual information score of the completed matrix was larger than that of the 16S matrix up to 50% missing ratio. It implies that the matrix completion is meaningful even in quite hard situations—50% sample loss implies 75% loss in entries. This result encourages us (and hopefully readers) to apply the *em* algorithm to other data such as gene networks (Vert and Kanehisa, 2003).

5. As another evaluation measure, you can use the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985, Yeung and Ruzzo, 2001). In this experiment, we computed the ARI as well, and the similar results were obtained.

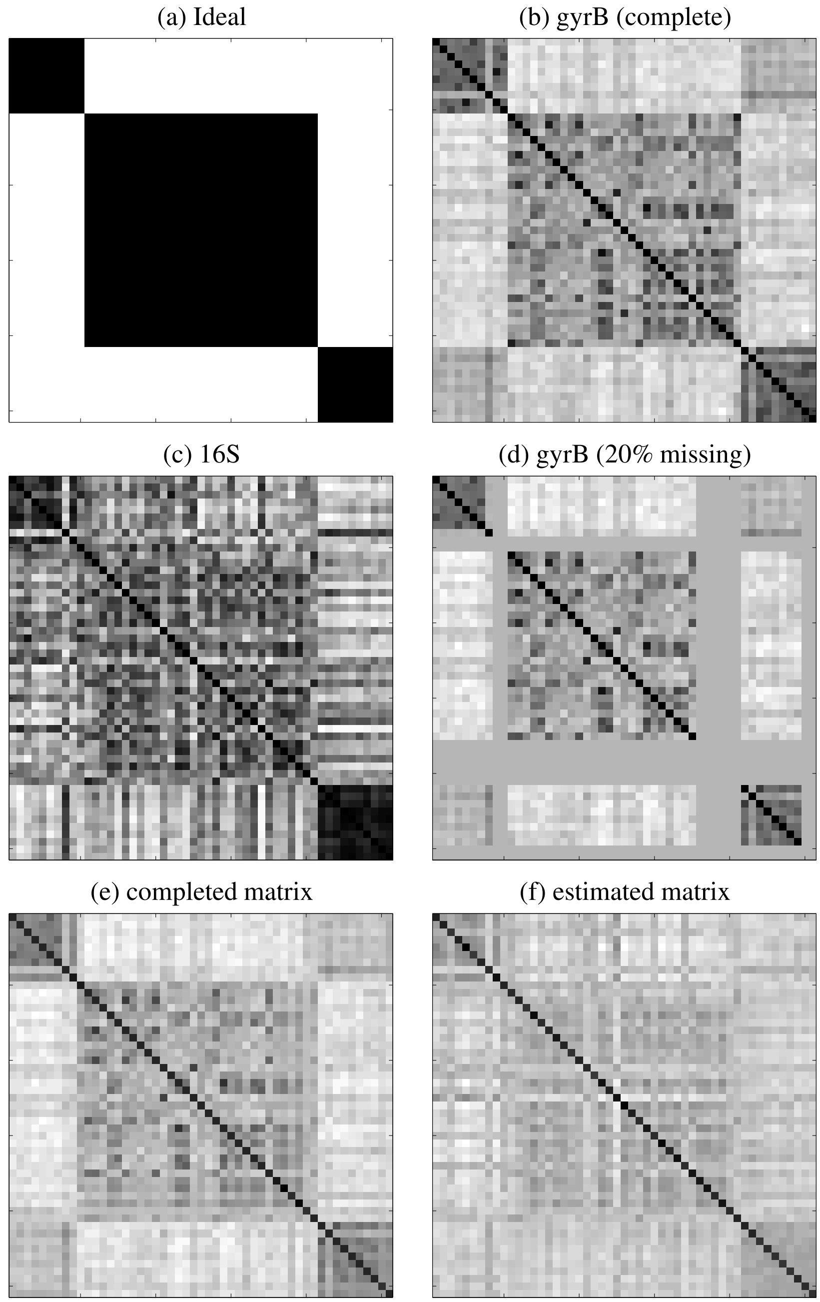

Figure 2: An example of kernel matrix completion. Suppose that we have an incomplete kernel matrix like (d). Our purpose is to complete the missing part by using a complete auxiliary kernel matrix (c). As the result of the *em* algorithm, we end up with two matrices. One is the completed matrix (e) on the data manifold, and the other is the estimated matrix (f) on the model manifold. Notice that (a) describes the correct class labels and (b) shows the complete version of (d).

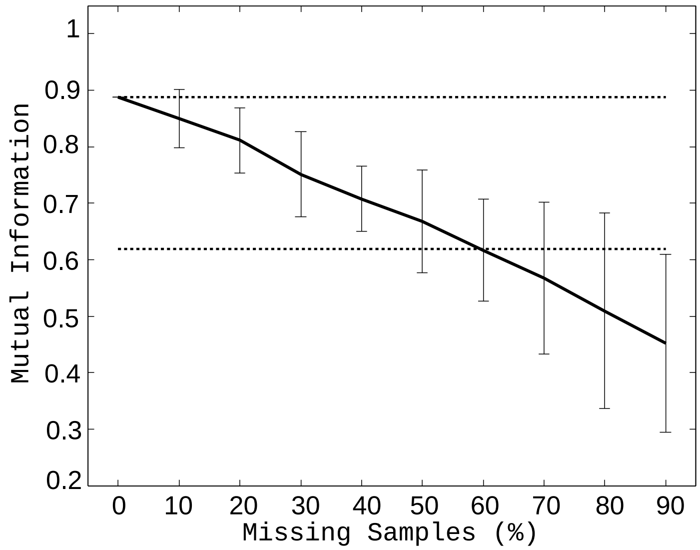

Figure 3: Clustering performance of the completed matrix. The solid curve shows the averaged mutual information score of the completed matrix, and the error bar describes the standard deviation. The upper and lower flat lines show the mutual information scores of the complete gyrB and 16S kernel matrices, respectively.

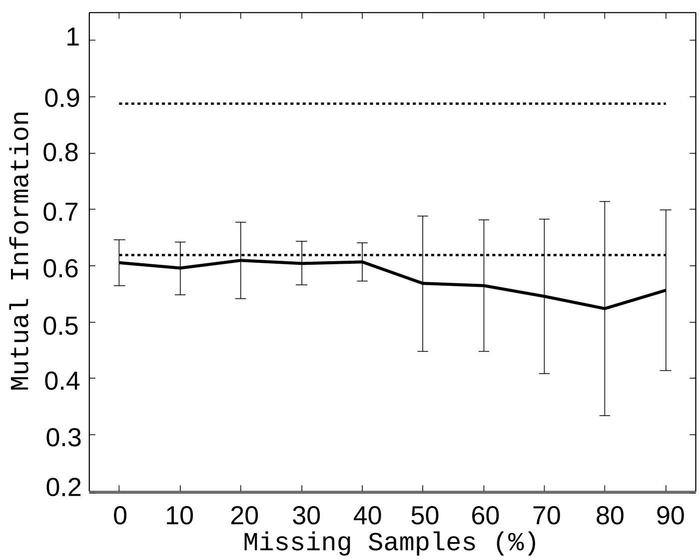

Figure 4: Clustering performance of the estimated matrix. The solid curve shows the averaged mutual information score of the estimated matrix, and the error bar describes the standard deviation.

7. Possible Extension

As we related the *em* algorithm to maximum likelihood inference in Section 5, it is straightforward to generalize it to the maximum *a posteriori* (MAP) inference or more generally the Bayes inference (Robert, 1994). For example, we are going to modify the *em* algorithm to obtain the MAP estimate. The MAP estimation amounts to minimizing the KL divergence penalized by a prior,

$$KL(D,M) - \log \pi(M), \qquad D \in \mathcal{D}, \quad M \in \mathcal{M},$$

where $\pi(M)$ is a prior distribution for M. Since the additional term $-\log \pi(M)$ depends only M, only the *m*-step is changed so as to minimize the above objective function with respect to M.

Let us give a simple example of the MAP estimation in the spectral variants case. In Bayesian inference, it is common to take a *conjugate prior*, so that the posterior distribution remains as a member of the exponential family. Since the model parameter b is related to a covariance matrix, we choose the Gamma distribution, which works as a conjugate prior for the variance of Gaussian distribution (Robert, 1994). The prior distribution is defined independently for each b_j as

$$\pi(b_j; \nu_j, \alpha_j) = \frac{1}{\Gamma(\nu_j)\alpha_j^{\nu_j}} \exp\left\{ -\frac{b_j}{\alpha_j} + (\nu_j - 1) \log b_j \right\},$$

where ν_j and α_j denote hyperparameters, by which the mean and the variance are specified by $E(b_j) = \alpha_j \nu_j$ and $V(b_j) = \alpha_j^2 \nu_j$, respectively. The m step for the MAP estimation is to minimize

$$L_m^{\mathrm{MAP}} = L_m - \sum_{j=1}^{\ell} \log \pi(b_j; \nu_j, \alpha_j),$$

which leads to the equation

$$\mathrm{tr}(M_i(\sum_{j=1}^{\ell} b_j M_j)^{-1}) + \frac{\nu_i - 1}{b_i} = \mathrm{tr}(M_i D) + \frac{1}{\alpha_i}, \qquad i = 1, \ldots, \ell.$$

In the spectral variants case, the left hand side is reduced to ν_i / b_i, thus we obtain the MAP solution in a closed form as

$$b_i = \frac{\nu_i}{\mathrm{tr}(M_i D) + 1/\alpha_i}, \qquad i = 1, \ldots, \ell.$$

When we construct the spectral variants model, the eigenvalues λ_j of the auxiliary matrix K_B are discarded. It may cause the undesirable solution, when λ_j carries significant information. To solve the problem, we can use the MAP estimation by constraining the parameter values around the corresponding eigenvalues. A typical choice is that ν_j is set to a certain constant ν_0 and α_j is determined so that the mean value $\alpha_j \nu_j$ is λ_j, i.e. $\alpha_j = \lambda_j / \nu_0$.

8. Conclusion

In this paper, we introduced the information geometry in the space of kernel matrices, and applied the *em* algorithm to matrix completion. The main difference from other Euclidean methods is that we use the KL divergence. In general, we cannot determine which distance is better, because it is highly data dependent. Nevertheless our method has a great utility, because it can be implemented with algebraic computation only and we do not need any specialized optimizer such as semidefinite programming (Graepel, 2002, Cristianini et al., 2002, Lanckriet et al., 2002). However, one disadvantage of our method is that the *em* algorithm may get stuck at local minima. Though each step is stated as a convex optimization problem, the overall optimization problem is not convex (Amari, 1995).

In Section 5, we related matrix completion to statistical inference. In future works, it would be interesting to involve advanced methods in statistical inference, such as generalized EM (Dempster et al., 1977) and variational Bayes (Attias, 1999). Also we are looking forward to apply our method to diverse kinds of real data which are not limited to bioinformatics.

Acknowledgments

The authors gratefully acknowledge that the bacterial *gyrB* amino acid sequences are offered by courtesy of Identification and Classification of Bacteria (ICB) database team of Marine Biotechnology Institute, Kamaishi, Japan. The authors would like to thank T. Kin, Y. Nishimori, T. Tsuchiya and J.-P. Vert for fruitful discussions. Many thanks to the anonymous reviewers for their helpful comments.

Appendix A. Analytical Solvability of the *m*-step

In this appendix, we discuss the solvability of the *m*-step. The left hand side of (16) is the *m*-coordinate of the submanifold $\mathcal{M}$, while b_j denote the *e*-coordinate of $\mathcal{M}$. The *e*-coordinate and *m*-coordinate are connected by the Legendre transform (Amari and Nagaoka, 2001). In the mother manifold $\mathcal{P}$, the Legendre transform is easily obtained as the inverse of the matrix. In the submanifold $\mathcal{M}$ of $\mathcal{P}$, however, it is difficult to obtain the Legendre transform in general. The difficulty is caused by the difference of geodesics defined in $\mathcal{M}$ and $\mathcal{P}$. When the geodesic defined by a coordinate system of a submanifold $\mathcal{S} \subseteq \mathcal{P}$ coincides with the geodesic defined by the corresponding global coordinate system of $\mathcal{P}$, the submanifold is called *autoparallel*. In our case, $\mathcal{M}$ is autoparallel for the *e*-coordinate, but it is not always autoparallel for the *m*-coordinate. When the submanifold is autoparallel for the both coordinate systems, the submanifold is called doubly autoparallel.

Let us consider when a submanifold becomes doubly autoparallel. To begin with, let us define the product $*$ between two $d \times d$ symmetric matrices $X, Y \in Sym(d)$,

$$X * Y = \frac{1}{2}(XY + YX). \tag{18}$$

The algebra equipped with the usual matrix sum and the product (18) is called the Jordan algebra of the vector space of $Sym(d)$. The following theorem provides the necessary and sufficient condition for a submanifold to be doubly autoparallel (Ohara, 1999).

Theorem 1 *Assume the identity matrix I is an element of the submanifold $\mathcal{M}$. Then $\mathcal{M}$ is doubly autoparallel if and only if the tangent space of $\mathcal{M}$ is a Jordan subalgebra of $Sym(d)$.*

Ohara (1999) has also shown that, if $\mathcal{M}$ is doubly autoparallel, the *m*-projection can be solved analytically, that is, the optimal solution is obtained by one Newton step.

For example, let us consider a submanifold $\mathcal{M} \subseteq \mathcal{P}$ is determined as (17), the *m*-coordinate on $\mathcal{M}$ is defined by

$$\eta_i = \mathrm{tr}(N_i(\sum_{j=1}^{c} b_j N_j)^{-1}).$$

Although the *m*-projection onto $\mathcal{M}$ from D is explicitly given by $\eta_i = \mathrm{tr}(N_i D)$, the Legendre transform from η_i to the *e*-coordinate b_j is not obtained in a closed form in general. However, if N_i forms the Jordan subalgebra, that is, the following holds for all i, j:

$$N_i * N_j \in \mathrm{span}(\{N_1, \ldots, N_c\}),$$

then M is doubly autoparallel and the Legendre transform can be obtained in a closed form. In the spectral variants case that we have considered, $N_i = v_i v_i^\top$ and

$$N_i * N_j = 0 \in \mathrm{span}(\{N_1, \ldots, N_c\}).$$

Thus the *m*-projection is obtained analytically in this case.

References

S. Amari. Information geometry of the EM and em algorithms for neural networks. *Neural Networks*, 8(9): 1379–1408, 1995.

S. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. American Mathematical Society, 2001.

H. Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Uncertainty in Artificial Intelligence: Proceedings of the Fifteenth Conference (UAI-1999)*, pages 21–30, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

O. Bousquet and D.J.L. Herrmann. On the complexity of learning the kernel matrix. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003. to appear.

T.A. Brown. *Genomes*. Wiley, 2nd edition, 2002.

N. Cristianini, J. Shawe-Taylor, J. Kandola, and A. Elisseeff. On kernel-target alignment. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 367–373. MIT Press, 2002.

L. Csató and M. Opper. Sparse online gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

I. Csiszár and G. Tusnady. Information geometry and alternating minimization procedures. *Statistics and Decisions*, 1:205–237, 1984.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Stat. Soc. B*, 39:1–38, 1977.

T. Graepel. Kernel matrix completion by semidefinite programming. In J.R. Dorronsoro, editor, *Artificial Neural Networks – ICANN 2002*, pages 687–693. Springer Verlag, 2002.

L. Hubert and P. Arabie. Comparing partitions. *J. Classif.*, pages 193–218, 1985.

S. Ikeda, S. Amari, and H. Nakahara. Convergence of the wake-sleep algorithm. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 239–245. MIT Press, 1999.

H. Kasai, A. Bairoch, K. Watanabe, K. Isono, S. Harayama, E. Gasteiger, and S. Yamamoto. Construction of the gyrB database for the identification and classification of bacteria. In *Genome Informatics 1998*, pages 13–21. Universal Academic Press, 1998.

G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the kernel matrix with semi-definite programming. In C. Sammut and A.G. Hoffmann, editors, *Proceedings of the 19th International Conference on Machine Learning*, pages 323–330. Morgan Kaufmann, 2002.

R.M. Neal and G.E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1999.

A. Ohara. Information geometric analysis of an interior point method for semidefinite programming. In O.E. Barndorff-Nielsen and E.B. Vedel Jensen, editors, *Geometry in Present Day Science*, pages 49–74. World Scientific, 1999.

A. Ohara, N. Suda, and S. Amari. Dualistic differential geometry of positive definite matrices and its applications to related problems. *Linear Algebra and Its Applications*, 247:31–53, 1996.

C.P. Robert. *The Bayesian Choice: A Decision-Theoretic Motivation*. Springer Verlag, 1994.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

M. Seeger. Covariance kernels from Bayesian generative models. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 905–912, Cambridge, MA, 2002. MIT Press.

M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18(Suppl. 1): S268–S275, 2002.

S. Vaithyanathan and B. Dom. Model selection in unsupervised learning with applications to document clustering. In I. Bratko and S. Dzeroski, editors, *Proceedings of 16th International Conference on Machine Learning (ICML)*, pages 433–443. Morgan Kaufmann, 1999.

V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

J.-P. Vert and M. Kanehisa. Graph-driven features extraction from microarray data using diffusion kernels and kernel CCA. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003. to appear.

C.K.I. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 599–621. MIT Press, 1999.

S. Yamamoto, H. Kasai, D.L. Arnold, R.W. Jackson, A. Vivian, and S. Harayama. Phylogeny of the genus Pseudomonas: intrageneric structure reconstructed from the nucleotide sequences of gyrB and rpoD genes. *Microbiology*, 146:2385–2394, 2000.

K.Y. Yeung and W.L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.