Efficient Formulations for 1-SVM and their Application to Recommendation Tasks

Yasutoshi Yajima and Tien-Fang Kuo
Department of Industrial Engineering and Management, Tokyo Institute of Technology, Japan
Email: {yasutosi,kuo}@me.titech.ac.jp

Abstract— **The present paper proposes new approaches for recommendation tasks based on one-class support vector machines (1-SVMs) with graph kernels generated from a Laplacian matrix. We introduce new formulations for the 1-SVM that can manipulate graph kernels quite efficiently. We demonstrate that the proposed formulations fully utilize the sparse structure of the Laplacian matrix, which enables the proposed approaches to be applied to recommendation tasks having a large number of customers and products in practical computational times. Results of various numerical experiments demonstrating the high performance of the proposed approaches are presented.**

Index Terms— **support vector machine, Laplacian matrix, graph kernel, quadratic programming problem, collaborative filtering, recommender system**

I. Introduction

Recently, the importance of recommender systems has increased rapidly with the growing availability of online information on the Web. Customers visiting the largest e-commerce sites often have difficulty in finding a particular item among the enormous number of products for sale. Many recommender systems [1], [2] have been installed to filter out irrelevant products and locate products that might be of interest to individual customers.

Collaborative filtering is one of the most successful technologies for recommendation tasks, in which customer ratings on products or historical records of purchased products are exploited to extract the preferences of individuals. Collaborative filtering calculates similarities between customers based on the customer rating, or the purchased products patterns of each individual. Collaborative filtering then finds a set of the most similar patterns, and recommends products for a particular individual. In the present paper, we provide new approaches for recommendation tasks using *kernels* defined on a graph where the nodes correspond to data items such as the products, and the weights of the edges correspond the relations between the products. There have been several graph based kernels which can be used to obtain similarities between data points.

Very recently, Fouss *et al.* [3] introduced a graph kernel, referred to as the commute time kernel and directly applied the kernel-based dissimilarities to the recommendation task. More precisely, they defined the kernel over a bipartite graph with two sets of nodes corresponding to a set of customers and products. They placed edges between the customer nodes and the product nodes when the customer has purchased the product. They defined a random walk model over this graph by assigning the transition probabilities over the edges. They showed that the average commute time between the two nodes is given by the kernel and that it can be used as a distance measure between the corresponding customer and product.

In the present paper, we use the 1-SVM with graph-based kernels to select relevant products for each customer. We introduce new formulations for the 1-SVM that can efficiently manipulate several recently developed graph kernels, such as [4], [5], [6], [7]. In addition, we show that a special case of our formulation does not require any optimization calculations. More importantly, the new kernel matrix is significantly smaller than that of the method reported in [3], which enables us to apply the present approach to large e-commerce sites with a practical amount of computation.

In Sect. II, we briefly review the standard formulation of the 1-SVM and its basic settings for recommendation tasks. In Sect. III, we describe various graph kernels, and in Sect. IV, we introduce new formulations for the 1-SVM. Experiments using a movie dataset are presented in Sect. V, and conclusions are presented in Sect. VI.

II. 1-SVM for Recommendation

The SVM was originally designed as a method for two-class classification problems where both positive and negative examples are required to learn discriminate functions. In this section, we will describe a variant of the SVM, called the one-class SVM (1-SVM) which can handle problems that consider a single class of data points. In [8], Schölkopf *et al.* have proposed a method for adapting the conventional two-class SVM formulation to the one-class problems.

Suppose that we have a set of N-dimensional data points $\boldsymbol{x}_j \in \mathbf{R}^N$ $(j = 1, 2, \ldots, l)$. Also, assume that we have a function $\phi(\cdot) : \mathbf{R}^N \mapsto \mathcal{F}$ that maps the data points into a higher-dimensional *feature space*, denoted by $\mathcal{F}$. Hereinafter, for simplicity, we denote the mapped image $\phi(\boldsymbol{x}_j)$ as ϕ_j. Let $\boldsymbol{w} \in \mathcal{F}$ and $\rho \in \mathbf{R}$. Also, the inner product in $\mathcal{F}$ is denoted as $\langle \cdot\,, \cdot \rangle$. The purpose of the 1-SVM is to calculate a hyperplane that holds most of the data points in its positive side, i.e., $\langle \boldsymbol{w}\,, \phi_j \rangle - \rho > 0$.

This paper is based on "One-Class Support Vector Machines for Recommendation Tasks," by Y. Yajima, which appeared in the Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2006), Singapore, April 9-12, 2006.

Introducing additional variables $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_l)^T$, $\boldsymbol{w}$ and ρ are obtained by solving the following quadratic programming problem:

$$\left| \begin{array}{ll} \text{Min.} & \frac{1}{2} \langle \boldsymbol{w}, \boldsymbol{w} \rangle + \frac{1}{\nu l} \sum_{j=1}^{l} \xi_j - \rho \\ \text{s.t.} & \langle w, \phi_j \rangle + \xi_j \geq \rho, \quad \xi_j \geq 0, \ j = 1, 2, \ldots, l, \end{array} \right. \tag{1}$$

where $\nu \in (0, 1]$ is a predetermined positive parameter. Let $(\boldsymbol{w}^*, \rho^*)$ denote an optimal solution of the problem (1). When a data point, the mapped image of which is denoted by ϕ, belongs to the negative side of the hyperplane, i.e., $\langle \boldsymbol{w}^*, \phi \rangle - \rho^* < 0$, the pattern can be considered to be different from the given single class of data points.

The objective of the recommendation task is to find products that have not yet been purchased but that would likely be purchased by a specific customer, hereinafter referred to as an *active* customer. Suppose that we are given a set of products $P = \{1, 2, \ldots, M\}$ and that, for each product $j \in P$, the associated feature vectors $\phi_j \in \mathcal{F}$ are obtained. In addition, let $P(a) \subseteq P$ be a subset of indices that are rated as preferable products, or that have actually been purchased by the active customer a. For simplicity, let us assume that $P(a)$ consists of l products and is denoted as $P(a) = \{1, 2, \ldots, l\}$, which is treated as a set of the single class of data points in the problem (1). Let $(\boldsymbol{w}^*, \rho^*)$ denote an optimal solution of (1). Then, for each product i that has not been purchased, i.e., $i \in P \setminus P(a)$, the distance from the hyperplane calculated as $(\langle \boldsymbol{w}^*, \phi_i \rangle - \rho^*) / \langle \boldsymbol{w}^*, \boldsymbol{w}^* \rangle$ can be used as a preference score of the product i. Ignoring the constants, one can use the inner product $\langle \boldsymbol{w}^*, \phi_i \rangle$ as a *score* to rank the product i for the specific active customer a. It has been shown that the parameter ν enjoys the ν-*property* [8] described below:

Theorem 1: ν is an upper bound on the fraction of the data points lying in the negative side of the hyperplane. Also, ν is a lower bound on the fraction of the data points lying in the nonnegative side of the hyperplane, i.e., $\langle \boldsymbol{w}^*, \phi_i \rangle - \rho^* \geq 0$.

Generating a nonlinear map $\phi(\cdot)$ is quite important in SVM. Usually, this is done implicitly by *kernels* that are naturally introduced by the following dual formulation of the problem (1).

$$\left| \begin{array}{ll} \text{Max.} & -\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \langle \phi_i, \phi_j \rangle \alpha_i \alpha_j \\ \text{s.t.} & \sum_{j=1}^{l} \alpha_j = 1, \quad 0 \leq \alpha_j \leq \frac{1}{\nu l}, \ j = 1, 2, \ldots, l, \end{array} \right. \tag{2}$$

where $\alpha_1, \alpha_2, \cdots, \alpha_l$ are dual variables. Note that the dual formulation can be defined using only the values of the inner products, without knowing the mapped image ϕ_i, explicitly. In addition, let $(\alpha_1^*, \alpha_2^*, \cdots, \alpha_l^*)$ be an optimal solution of the dual problem. Then, the associated optimal primal solution is given as $\boldsymbol{w}^* = \sum_{j=1}^{l} \alpha_j^* \phi_j$, which immediately implies that the score of the product i is given by $\langle \boldsymbol{w}^*, \phi_i \rangle = \sum_{j=1}^{l} \alpha_j^* \langle \phi_i, \phi_j \rangle$.

Let $K = \{K_{ij}\}$ be a symmetric matrix called a kernel matrix, which consists of the inner products $\langle \phi_i, \phi_j \rangle$ as the $i - j$ element. Any positive semidefinite matrices can be used as a kernel matrix K. It has been shown that positive semidefiniteness ensures the existence of the mapped points, ϕ_i's (see, for example, [9]). More precisely, let U be an orthonormal matrix whose columns correspond to the eigenvectors of K, and Λ be a diagonal matrix whose diagonal elements correspond to the eigenvalues of K. When the matrix K is positive semidefinite, one can obtain the eigendecomposition $K = U \Lambda U^T$. Then, the mapped image ϕ_i is explicitly given as the ith column vector of the matrix $\Lambda^{\frac{1}{2}} U^T$, i.e.,

$$\Lambda^{\frac{1}{2}} U^T = [\phi_1 \phi_2 \cdots \phi_l]. \tag{3}$$

In the next section, we will introduce recently developed kernel matrices defined on the graph.

III. Laplacian of a Graph and Associated Kernel

Recently, several studies [4], [5], [6], [7] have reported the development of kernels using weighted graphs. In this section, we will review such kernels.

First, let us introduce a weighted graph $G(V, E)$ having a set of nodes V and a set of undirected edges E. The set of nodes V corresponds to a set of data items such as products in a recommendation task. For each edge $(i, j) \in E$, a positive weight $b_{ij} > 0$ representing the similarity between the two nodes $i, j \in V$ is assigned. We assume that the larger the weight b_{ij}, the greater the similarity between the two nodes. Note that if there exists no edge between i and j, then we set $b_{ij} = 0$.

The edge weights b_{ij}'s are defined in several ways. For example, the following exponential function is often used

$$b_{ij} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2} \right), \tag{4}$$

where σ is a bandwidth hyperparameter. One can also use a k-nearest-neighbor graph where we put an edge between the node i and j when the data point $\boldsymbol{x}_i$ is among the k nearest neighbors of $\boldsymbol{x}_j$ or vice versa, and assign a weight b_{ij} as (4), or, simply, $b_{ij} = 1$ for each edge (i, j). Let M be the number of nodes in V, and let B be an $M \times M$ symmetric matrix with elements b_{ij} for $(i, j) \in E$.

Next, let us introduce the Laplacian matrix L of the graph $G(V, E)$ as $L = D - B$, where D is a diagonal matrix, the diagonal elements d_{ii} of which are the sum of the ith row of B, i.e., $d_{ii} = \sum_j b_{ij}$. Throughout this paper, we assume that the graph $G(V, E)$ is connected. This implies that the rank of the matrix L is $M - 1$, and that the null space of L is the one-dimensional space spanned by the vector of all ones, i.e., $L\boldsymbol{e} = 0$. Also, It has been known that L is positive semidefinite (see [10] for further details).

There are several methods for generating kernel matrices based on L. Fouss *et al.* [3] considered a random walk model on the graph G, in which, for each edge (i, j), the transition probability p_{ij} is defined as $p_{ij} = b_{ij} / \sum_{k=1}^{M} b_{ik}$. Intuitively, at each node i, the transition probability to the node j is proportional to the weight b_{ij}. They considered the average commute time $n(i, j)$,

which represents the average number of steps that a random walker, starting from node i, will take to enter node j for the first time and then return to node i. They indicated that the average commute time $n(i,j)$ can be used as a dissimilarity measure between any two data points corresponding to the nodes of the graph, and that $n(i,j)$ is given as $n(i,j) = V_G\left(l_{ii}^+ + l_{jj}^+ - 2l_{ij}^+\right)$, where $V_G = \sum_{i,j} b_{ij}$ and l_{ij}^+ is the $i-j$ element of the Moore-Penrose pseudoinverse of L, which is denoted by L^+. Fouss *et al.* [3] also showed that as long as the graph is connected, the pseudoinverse L^+ is explicitly given as follows:

$$L^+ = \left(L - \boldsymbol{e}\boldsymbol{e}^T/M\right)^{-1} + \boldsymbol{e}\boldsymbol{e}^T/M, \tag{5}$$

where $\boldsymbol{e}$ is a vector of all ones. Since L is positive semidefinite [10], so is its pseudoinverse L^+, which implies that L^+ can act as a kernel matrix [3].

Here, L and L^+ share the common eigenvectors. Let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M$ and $\lambda_1, \lambda_2, \ldots, \lambda_M$ be the eigenvectors and the corresponding eigenvalues of L, respectively. It is well-known that L is decomposed into $L = \sum_{i=1}^M \lambda_i(\boldsymbol{v}_i\boldsymbol{v}_i^T)$, and that the pseudoinverse is also given as

$$L^+ = \sum_{i=1}^M \lambda_i^+(\boldsymbol{v}_i\boldsymbol{v}_i^T), \quad \text{where} \quad \lambda^+ = \begin{cases} \lambda^{-1} & \text{if } \lambda \neq 0 \\ 0 & \text{if } \lambda = 0. \end{cases} \tag{6}$$

We note that

$$L^+\boldsymbol{e} = 0. \tag{7}$$

Several variants of the above equation have been proposed. Zhu *et al.* [11] introduced the following regularized Laplacian kernel matrix

$$\sum_{i=1}^M (1+t\lambda_i)^{-1}\boldsymbol{v}_i\boldsymbol{v}_i^T = \sum_{k=0}^\infty t^k\left(-L\right)^k = \left(I+tL\right)^{-1}. \tag{8}$$

Moreover, by introducing the modified Laplacian $L_r = rD - B$ with a parameter $0 \leq r \leq 1$, Ito *et al.* [7] defined the modified Laplacian regularized kernel matrix as

$$\left(I + tL_r\right)^{-1}. \tag{9}$$

In particular, when $r = 0$ this kernel matrix is the von Neumann diffusion kernel, which is defined as

$$\sum_{k=0}^\infty t^k B^k = \left(I - tB\right)^{-1}. \tag{10}$$

Furthermore, introducing the *normalized Laplacian* $\tilde{L} \equiv D^{-1/2}LD^{-1/2}$, Smola and Kondor [5] proposed several kernel matrices such as the diffusion kernel

$$\exp(-t\tilde{L}) \tag{11}$$

and a normalized variant of the regularized Laplacian kernel defined as follows:

$$\left(I + t\tilde{L}\right)^{-1}. \tag{12}$$

IV. Learning 1-SVMs with Graph Kernels

Next, we will describe recommendation methods based on the 1-SVM using the kernel matrices described in the previous section. Recall that we are given a set of M products $P = \{1, 2, \ldots, M\}$ and a subset $P(a) \subseteq P$, which have been purchased by the active customer a. We assume that $P(a) = \{1, 2, \ldots, l\}$. In addition, the elements of the kernel matrix K represent the inner products of the feature vectors corresponding to the products.

Let us first rewrite the primal formulation. To this end, introducing M variables $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_M)^T$, let us assume that $\boldsymbol{w} \in \mathcal{F}$ is given as a linear combination of M points as follows:

$$\boldsymbol{w} = \sum_{j=1}^M \alpha_j\phi_j.$$

Substituting this equation into the primal problem (1), the following formulation is obtained:

$$\left|\begin{array}{ll} \text{Min.} & \frac{1}{2}\boldsymbol{\alpha}^T K\boldsymbol{\alpha} + \frac{1}{\nu l}\sum_{i=1}^l \xi_i - \rho \\ \text{s.t.} & \left\langle \sum_{i=1}^M \alpha_i\phi_i\,,\, \phi_j \right\rangle + \xi_j \geq \rho,\ j = 1, 2, \ldots, l, \\ & \xi_j \geq 0,\ j = 1, 2, \ldots, l. \end{array}\right. \tag{13}$$

Note that the norm of $\boldsymbol{w}$ can be written as follows:

$$\left\langle \boldsymbol{w}\,,\, \boldsymbol{w} \right\rangle = \left\langle \sum_{j=1}^M \alpha_j\phi_j\,,\, \sum_{j=1}^M \alpha_j\phi_j \right\rangle = \boldsymbol{\alpha}^T K\boldsymbol{\alpha}.$$

Let $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_M^*)^T$ be an optimal solution of this problem. Then, the preference score of the product i is given as the ith element of the vector $K\boldsymbol{\alpha}^*$, i.e.,

$$\left\langle \sum_{j=1}^M \alpha_j^*\phi_i\,,\, \phi_j \right\rangle = \left(K\boldsymbol{\alpha}^*\right)_i. \tag{14}$$

Here, generating the kernel matrices given in Sect. III requires calculation of the inverse of the matrices as described in (5) and (8). The inverse operations require a significant computational effort, which prevents us from using these kernel matrices for the recommendation tasks when the number of products is large. Moreover, in general, these kernel matrices become fully dense, which causes difficulty in holding the kernel matrices in memory during the time required for solving the problem (13). In the subsequent subsections, however, we will propose new formulations of 1-SVMs with kernel matrices the inverse of which are readily available. Exploiting the special structures of the kernel matrices, we will derive simpler formulations for solving 1-SVM with the kernel matrices (5) and (8).

A. Regularized Laplacian Kernel

Suppose that the kernel matrix K is the regularized Laplacian kernel matrix (8). Let us first introduce a new vector of variables $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_M)^T \in \mathbf{R}^M$, and define $\boldsymbol{\beta} \equiv K\boldsymbol{\alpha}$. Note that

$$\beta_j = \left(K\boldsymbol{\alpha}\right)_j = \left\langle \sum_{i=1}^M \alpha_i\phi_i\,,\, \phi_j \right\rangle$$

holds for each j. It follows that

$$\boldsymbol{\alpha} = K^{-1}\boldsymbol{\beta} = (I + tL)\,\boldsymbol{\beta}$$

holds. Furthermore, a straightforward calculation reveals that

$$\boldsymbol{\alpha}^T K \boldsymbol{\alpha} = \boldsymbol{\beta}^T (I + tL)\,\boldsymbol{\beta}.$$

Therefore, the problem (13) can be equivalently formulated with respect to the new variable $\boldsymbol{\beta}$ as follows:

$$\left| \begin{array}{ll} \text{Min.} & \frac{1}{2}\boldsymbol{\beta}^T (I + tL)\,\boldsymbol{\beta} - \rho + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \\ \text{s.t.} & \beta_j + \xi_j \geq \rho, \;\; \xi_j \geq 0, \;\; j = 1, 2, \ldots, l. \end{array} \right. \tag{15}$$

Associated with this formulation, we have the following theorem.

Theorem 2: The problem (15) has an optimal solution $(\boldsymbol{\beta}^*, \boldsymbol{\xi}^*, \rho^*)$ which satisfies

$$0 \leq \beta_j^* \leq \rho^*, \quad j = 1, 2, \ldots, M.$$

Proof: By the Karush–Kuhn–Tucker (KKT) conditions, there exist nonnegative Lagrangian multipliers v_j $(j = 1, \ldots, l)$ such that the optimal solution $(\boldsymbol{\beta}^*, \rho^*, \boldsymbol{\xi}^*)$ satisfies

$$\left((I + tL)\,\boldsymbol{\beta}^*\right)_j = \left\{ \begin{array}{ll} v_j, & \forall j = 1, 2, \ldots, l, \\ 0, & \forall j = l+1, l+2, \ldots, M, \end{array} \right. \tag{16}$$

and the complementarity conditions

$$v_j \left(\beta_j^* + \xi_j^* - \rho^*\right) = 0, \quad \forall j = 1, 2, \ldots, l. \tag{17}$$

Moreover, we note that any optimal solutions must satisfy the following

$$\xi_i^* = \max\left\{0, \rho^* - \beta_i^*\right\}, \quad \forall i = 1, 2, \ldots, l. \tag{18}$$

It follows from the definition of the Laplacian matrix L that the left hand side of (16) can be written in the following way:

$$\left((I + tL)\,\boldsymbol{\beta}^*\right)_j = \beta_j^* + t\sum_{i=1}^{M} b_{ji}\left(\beta_j^* - \beta_i^*\right),$$

which is nonnegative. This holds true for an index k attaining the minimum of β_j^*'s

$$k \equiv \text{argmin}\{\beta_j^* | j = 1, 2, \ldots, M\}.$$

Therefore, since t is a positive parameter,

$$\beta_k^* \geq -t\sum_{i=1}^{M} b_{ki}\left(\beta_k^* - \beta_i^*\right) \geq 0,$$

which implies that β_j^* is nonnegative for all $j = 1, 2, \ldots, M$, and that $\rho^* \geq 0$.

Next, on the contrary, let us assume that there exists an index h such that $\beta_h^* > \rho^*$. Without loss of generality, we also assume that

$$h \equiv \text{argmax}\{\beta_j^* | j = 1, 2, \ldots, M\}.$$

It follows that

$$\left((I + tL)\,\boldsymbol{\beta}^*\right)_h = \beta_h^* + t\sum_{i=1}^{M} b_{hi}\left(\beta_h^* - \beta_i^*\right) > 0. \tag{19}$$

From the KKT conditions (16), the index should be $h \leq l$, which implies that the associated Lagrangian multiplier is positive $\left((I + tL)\,\boldsymbol{\beta}^*\right)_h = v_h > 0$.

It follows from (18) that, however, we have $\xi_h^* = 0$, which results in

$$\beta_h^* + \xi_h^* - \rho^* > 0.$$

This contradicts the complementarity conditions (17), which completes the proof. ■

It follows from this theorem and the conditions (18) that the optimal solution satisfies the following equalities:

$$\xi_j = \rho - \beta_j, \;\; j = 1, 2, \ldots, l.$$

Substituting ξ_j into the problem (15), we then have a simpler formulation given below:

$$\left| \begin{array}{ll} \text{Min.} & \frac{1}{2}\boldsymbol{\beta}^T (I + tL)\,\boldsymbol{\beta} + \frac{1-\nu}{\nu}\rho - \frac{1}{\nu l}\sum_{i=1}^{l}\beta_i \\ \text{s.t.} & \beta_j \leq \rho, \;\; j = 1, 2, \ldots, l. \end{array} \right. \tag{20}$$

In many practical situations, the Laplacian matrix L is very sparse and can be stored in main memory even if the number of data points is very huge. Moreover, it has been shown that this problem can be equivalently optimized by solving an *unconstrained minimization problem* using an implicit Lagrangian function. A more detailed description can be found in [12].

B. Commute Time Kernel

Let us consider the case when we use the commute time kernel matrix L^+ as K in (13), i.e.,

$$\left| \begin{array}{ll} \text{Min.} & \frac{1}{2}\boldsymbol{\alpha}^T L^+ \boldsymbol{\alpha} + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i - \rho \\ \text{s.t.} & \left(L^+\boldsymbol{\alpha}\right)_j + \xi_j \geq \rho, \;\; j = 1, 2, \ldots, l, \\ & \xi_j \geq 0, \;\; j = 1, 2, \ldots, l. \end{array} \right. \tag{21}$$

We will also show that a simpler formulation can be derived.

First, let $(\boldsymbol{\alpha}^*, \rho^*, \boldsymbol{\xi}^*)$ be an optimal solution of (21). It follows from (7) that, for any real number $\theta \in \mathbf{R}$, we have

$$L^+\left(\boldsymbol{\alpha}^* + \boldsymbol{e}\theta\right) = L^+\boldsymbol{\alpha}^*.$$

This implies that the solution $(\boldsymbol{\alpha}^* + \boldsymbol{e}\theta, \rho^*, \boldsymbol{\xi}^*)$ is also optimal. Thus, there exists an optimal solution satisfying the following equality constraint

$$\boldsymbol{e}^T\boldsymbol{\alpha} = 1, \tag{22}$$

which can be added to the problem (21).

Next, as in the previous section, let us introduce a vector of variables $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_M)^T$, and let us define

$$\boldsymbol{\beta} \equiv \left(L^+ - \boldsymbol{e}\boldsymbol{e}^T/M\right)\boldsymbol{\alpha} + \boldsymbol{e}/M. \tag{23}$$

For each j, if $\boldsymbol{\alpha}$ satisfies the added constraint (22), then

$$\beta_j = \left(L^+\boldsymbol{\alpha} - \frac{\boldsymbol{e}}{M}\boldsymbol{e}^T\boldsymbol{\alpha} + \frac{\boldsymbol{e}}{M}\right)_j = \left(L^+\boldsymbol{\alpha}\right)_j = \left\langle \sum_{i=1}^{M}\alpha_i\phi_i\,,\, \phi_j \right\rangle$$

holds. Therefore, it follows from (5) and (23) that

$$\boldsymbol{\alpha} = \left(L - \frac{\boldsymbol{e}\boldsymbol{e}^T}{M}\right)\left(\boldsymbol{\beta} - \frac{\boldsymbol{e}}{M}\right) \tag{24}$$

holds. In addition, we can easily verify that the constraint (22) is written as

$$\boldsymbol{e}^T\boldsymbol{\alpha} = \boldsymbol{e}^T\left(L - \frac{\boldsymbol{e}\boldsymbol{e}^T}{M}\right)\left(\boldsymbol{\beta} - \frac{\boldsymbol{e}}{M}\right) = -\boldsymbol{e}^T\left(\boldsymbol{\beta} - \frac{\boldsymbol{e}}{M}\right) = 1.$$

Then the variable $\boldsymbol{\beta}$ has to satisfy $\boldsymbol{e}^T\boldsymbol{\beta} = 0$. Furthermore, $\boldsymbol{\alpha}^T L^+ \boldsymbol{\alpha} = \boldsymbol{\beta}^T L \boldsymbol{\beta}$ holds if $\boldsymbol{\beta}$ satisfies $\boldsymbol{e}^T\boldsymbol{\beta} = 0$. Therefore, the problem (21) with the constraint (22) can be equivalently formulated as follows:

$$\left|\begin{array}{ll} \text{Min.} & \frac{1}{2}\boldsymbol{\beta}^T L\boldsymbol{\beta} - \rho + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i \\ \text{s.t.} & \beta_j + \xi_j \geq \rho, \;\; \xi_j \geq 0, \;\; j = 1, 2, \ldots, l, \\ & \boldsymbol{e}^T\boldsymbol{\beta} = 0. \end{array}\right. \tag{25}$$

Let $(\boldsymbol{\beta}^*, \boldsymbol{\xi}^*, \rho^*)$ be an optimal solution of the problem (25). We have the following theorem.

Theorem 3: The optimal solution $(\boldsymbol{\beta}^*, \boldsymbol{\xi}^*, \rho^*)$ satisfies $\beta_j^* \leq \rho^*$ for all $j = 1, \ldots, M$.

Proof: Let

$$\bar{\beta} \equiv \max\left\{\beta_j^* \mid j = 1, 2, \ldots, M\right\}.$$

For the purpose of contradiction, let us assume that $\bar{\beta} > \rho^*$. We will show that a better solution can be constructed.

Let I be a set of indices of the vector $\boldsymbol{\beta}^*$ defined as

$$I \equiv \left\{i | \beta_i^* = \bar{\beta}\right\}.$$

Note that any optimal solutions must satisfy the following

$$\xi_i^* = \max\left\{0, \rho^* - \beta_i^*\right\}, \quad \forall i = 1, 2, \ldots, l. \tag{26}$$

This implies that $\xi_i^* = 0$ for any $i \in I$. Then, for a sufficiently small $\epsilon > 0$, let us define a new solution $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \hat{\beta}_2, \cdots, \hat{\beta}_M)$, where

$$\hat{\beta}_i = \begin{cases} \bar{\beta} - \epsilon & \text{if } i \in I \\ \beta_i^* + \frac{|I|}{M - |I|}\epsilon & \text{otherwise} \end{cases}$$

It is obvious to see that $\hat{\boldsymbol{\beta}}$ satisfies $\boldsymbol{e}^T\hat{\boldsymbol{\beta}} = 0$. Also, when ϵ satisfies

$$0 < \epsilon \leq \rho^* - \bar{\beta},$$

$\hat{\beta}_i + \xi_i^* \geq \rho^*$ still hold true for all $i = 1, 2, \ldots, l$. Therefore, the point $(\hat{\boldsymbol{\beta}}, \boldsymbol{\xi}^*, \rho^*)$ is a feasible solution of the problem (25).

On the other hand, it follows from the definition of the matrix L, we have

$$\frac{1}{2}\boldsymbol{\beta}^{*T} L \boldsymbol{\beta}^* - \frac{1}{2}\hat{\boldsymbol{\beta}}^T L \hat{\boldsymbol{\beta}} = \sum_{i \in I}\sum_{j \notin I} b_{ij}\left\{(\bar{\beta} - \beta_j^*)^2 - (\bar{\beta} - \beta_j^* - \frac{M}{M - |I|}\epsilon)^2\right\} > 0,$$

which implies that the solution $(\hat{\boldsymbol{\beta}}, \boldsymbol{\xi}^*, \rho^*)$ yields a better objective function value than that of $(\boldsymbol{\beta}^*, \boldsymbol{\xi}^*, \rho^*)$ when ϵ satisfies

$$0 < \epsilon < \min\left\{\rho^* - \bar{\beta}, \frac{M - |I|}{M}(\bar{\beta} - \beta_j^*)\right\}, \; j \notin I,$$

which is a contradiction. ■

From Theorem 3 and (26), we can claim that the optimal solution $(\boldsymbol{\beta}^*, \boldsymbol{\xi}^*, \rho^*)$ of the problem (25) satisfies

$$\beta_j^* + \xi_j^* = \rho^*, \quad \forall j = 1, 2, \ldots, l.$$

Consequently, by substituting $\xi_j = \rho - \beta_j$, the problem (25) can be simplified as follows:

$$\left|\begin{array}{ll} \text{Min.} & \frac{1}{2}\boldsymbol{\beta}^T L\boldsymbol{\beta} + \frac{1-\nu}{\nu}\rho - \frac{1}{\nu l}\sum_{i=1}^{l}\beta_i \\ \text{s.t.} & \beta_j \leq \rho, \; j = 1, 2, \ldots, l, \\ & \boldsymbol{e}^T\boldsymbol{\beta} = 0, \end{array}\right. \tag{27}$$

which is also reduced into an *unconstrained minimization problem* using an implicit Lagrangian function [12].

C. Some Special Cases

It has been shown that the 1-SVM formulation given in (1) can be solved analytically when $\nu = 1.0$. This is also true for our formulation given in (13) with any types of kernel matrices given in Sect. III. We have the following theorem:

Theorem 4: Let $(\boldsymbol{\alpha}^*, \boldsymbol{\xi}^*, \rho^*)$ be an optimal solution of (13) with $\nu = 1.0$, i.e.,

$$\left|\begin{array}{ll} \text{Min.} & \frac{1}{2}\boldsymbol{\alpha}^T K\boldsymbol{\alpha} + \frac{1}{l}\sum_{j=1}^{l}\xi_j - \rho \\ \text{s.t.} & \left\langle \sum_{i=1}^{M}\alpha_i\phi_i \, , \, \phi_j \right\rangle + \xi_j \geq \rho, \; j = 1, 2, \ldots, l, \\ & \xi_j \geq 0, \; j = 1, 2, \ldots, l. \end{array}\right. \tag{28}$$

Then, the following inequalities hold true.

$$\left\langle \sum_{i=1}^{M}\alpha_i^*\phi_i \, , \, \phi_j \right\rangle \leq \rho^*, \quad \forall j = 1, 2, \ldots, l.$$

Proof: Let us assume, to the contrary, that there exists an index k such that $\left\langle \sum_{i=1}^{M}\alpha_i^*\phi_i \, , \, \phi_k \right\rangle > \rho^*$. It should be noted that $\xi_k^* = 0$.

Next, let $\Delta \equiv \left\langle \sum_{i=1}^{M}\alpha_i^*\phi_i \, , \, \phi_k \right\rangle - \rho^* > 0$. Then, we can define a new solution $\hat{\boldsymbol{\xi}} = (\hat{\xi}_1, \ldots, \hat{\xi}_l)$ and $\hat{\rho}$ as follows:

$$\hat{\xi}_j = \begin{cases} \xi_j^* + \Delta & \text{if } j \neq k, \\ \xi_k^* & \text{if } j = k, \end{cases} \quad \text{and} \quad \hat{\rho} = \rho^* + \Delta.$$

It is straightforward to verify that the solution $(\boldsymbol{\alpha}^*, \hat{\boldsymbol{\xi}}, \hat{\rho})$ also satisfies the constraints of the problem (28). In particular, we note that the equality

$$\left\langle \sum_{i=1}^{M}\alpha_i^*\phi_i \, , \, \phi_k \right\rangle + \hat{\xi}_k = \hat{\rho}$$

holds true because $\hat{\xi}_k = 0$.

The objective value of the solution $(\boldsymbol{\alpha}^*, \hat{\boldsymbol{\xi}}, \hat{\rho})$ is calculated as

$$\begin{aligned} &\frac{1}{2}\boldsymbol{\alpha}^{*T} K\boldsymbol{\alpha}^* + \frac{1}{l}\sum_{j=1}^{l}\hat{\xi}_j - \hat{\rho} \\ &= \frac{1}{2}\boldsymbol{\alpha}^{*T} K\boldsymbol{\alpha}^* + \frac{1}{l}\sum_{j \neq k}(\xi_j^* + \Delta) + \frac{1}{l}\xi_k^* - \rho^* - \Delta \\ &= \frac{1}{2}\boldsymbol{\alpha}^{*T} K\boldsymbol{\alpha}^* + \frac{1}{l}\sum_{j=1}^{l}\xi_j^* - \rho^* - \frac{\Delta}{l}, \end{aligned}$$

which contradicts the optimality of the solution $(\boldsymbol{\alpha}^*, \boldsymbol{\xi}^*, \rho^*)$. This completes the proof. ■

This theorem also ensures that

$$\xi_j^* = \rho^* - \left\langle \sum_{i=1}^{M} \alpha_i^* \phi_i \, , \, \phi_j \right\rangle$$

holds for each $j = 1, 2, \ldots, l$. Then, substituting these equations into the objective function of the problem (28), the following unconstrained minimization is obtained:

$$\Big| \text{ Min.} \quad W(\boldsymbol{\alpha}) = \tfrac{1}{2}\boldsymbol{\alpha}^T K \boldsymbol{\alpha} - \tfrac{1}{l}\boldsymbol{y}^T K \boldsymbol{\alpha} \tag{29}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^T$ is an M-dimensional vector whose elements are defined as follows:

$$y_j = \begin{cases} 1 & \text{if } j = 1, 2, \ldots, l, \\ 0 & \text{if } j = l+1, l+2, \ldots, M. \end{cases}$$

Note that $\boldsymbol{y}$ is a binary vector representing the purchased products by the active customer.

The problem (29) can be solved analytically. Since the gradient of the objective function $W(\boldsymbol{\alpha})$ is described as

$$\nabla W(\boldsymbol{\alpha}) = K\boldsymbol{\alpha} - \frac{1}{l} K \boldsymbol{y},$$

$\boldsymbol{\alpha}^* = \frac{1}{l}\boldsymbol{y}$ is an optimal solution of the problem (29).

Recall that the optimal preference score of the product i is given in (14). Also, for each active customer a, let $\boldsymbol{y}_a \in \mathbf{R}^M$ be an M-dimensional binary vector representing the purchased products. Then, when we use the regularized Laplacian kernel matrix (8), for instance, a vector of the optimal scores is given by

$$(I + tL)^{-1} \boldsymbol{y}_a. \tag{30}$$

Also, when we use the commute time kernel matrix (5), the associated score is expressed as

$$\left(L - \frac{\boldsymbol{e}\boldsymbol{e}^T}{M} \right)^{-1} \boldsymbol{y}_a. \tag{31}$$

Note that the constant terms are omitted in the above two expressions.

Especially, when we use the normalized variant of the regularized Laplacian kernel given in (12), the score is expressed as follows:

$$\begin{aligned} \left(I + t\tilde{L} \right)^{-1} \boldsymbol{y}_a &= \left(I + t(I - D^{-1/2} B D^{-1/2}) \right)^{-1} \boldsymbol{y}_a \\ &= \frac{1}{1+t} \left(I - \frac{t}{1+t} D^{-1/2} B D^{-1/2} \right)^{-1} \boldsymbol{y}_a, \end{aligned} \tag{32}$$

which is equivalent to the method studied in [13].

V. Computational Experiments

To evaluate the performances of the proposed approaches, numerical experiments are conducted using a real-world dataset. We use the MovieLens dataset developed at the University of Minnesota. This dataset contains 1,000,209 ratings of approximately 3,900 movies made by 6,040 customers. We use 100,000 randomly selected ratings [14] containing 943 customers and 1682 movies. This set of ratings is divided into five subsets to perform five-fold cross-validation. The divided dataset can be retrieved from `http://www.grouplens.org/data/`. Moreover, in order to demonstrate the scalability of the proposed approach, we use the original full dataset, which is also randomly divided into five subsets to perform the cross-validation.

In these experiments, all of the rating values are converted into binary values, indicating whether a customer has rated a movie. This conversion has been used in several papers, including [14], [3]. Let M and N be the number of movies and customers, respectively. Then the dataset is represented as an $N \times N$ binary matrix A, where the $i-j$ element $A_{ij} = 1$ if customer i has rated movie j.

In order to generate the graph-based kernels, we need construct a k-nearest-neighbor graph $G(V, E)$ where the set of nodes V corresponds to that of the movies. For each node $j \in V$, let $\boldsymbol{A}_j$ denote the jth column vector of the matrix A. Based on the cosine similarities

$$\frac{\boldsymbol{A}_i^T \boldsymbol{A}_j}{\|\boldsymbol{A}_i\| \|\boldsymbol{A}_j\|}$$

between movie i and movie j, when movie i is among the k nearest neighbors of movie j, or when movie j is among those of movie i, we place an edge $(i, j) \in E$ and assign a unit weight $b_{ij} = 1$. We report the results obtained by the commute time kernel matrix (**CT**), the regularized Laplacian kernel matrix (**RL**), the normalized variant of the regularize Laplacian kernel matrix (**NL**) and the diffusion kernel matrix (**DF**).

For each kernel matrix, we consider the 1-SVM with the parameter $\nu = 1.0$ for generating the preference scores, which can be achieved by solving a system of linear equations as described in Sect. IV-C. More precisely, for each active customer a, let $\boldsymbol{y}_a \in \mathbf{R}^M$ be an M-dimensional binary vector representing the rated movies. Then, the preference score of each movie i is given as the ith element of the vectors which are given as (30) through (32).

The cross-validation is conducted using the training and test set splits described above. We first calculate the score of the movies using the training set. Note that, for each active customer, the movies contained in the corresponding test set are not contained in the training set. Then, if the score is ideally correct, these movies have to be ranked higher than any other movies not rated in the training set. The performance of the proposed method is evaluated in the manner described in [3] using the degree of agreement, which is the proportion of pairs ranked in the correct order with respect to the total number of pairs. Therefore, a degree of agreement of 0.5 will be generated by the random ranking, whereas a degree of agreement of 1.0 is the correct ranking.

The average degrees of agreement of the five-fold cross-validation are given in Figs. 1 through 6, as well as in Table I. Figs. 2 through 4 show the results for the 100,000 selected ratings, while Figs. 5 and 6 corresponds to the full MovieLens dataset with more than one million

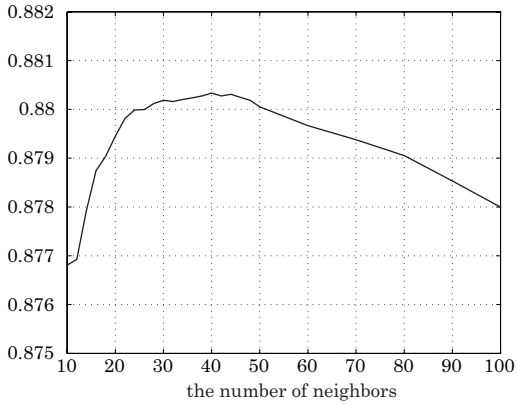

Figure 1. Results of the selected dataset by **CT**

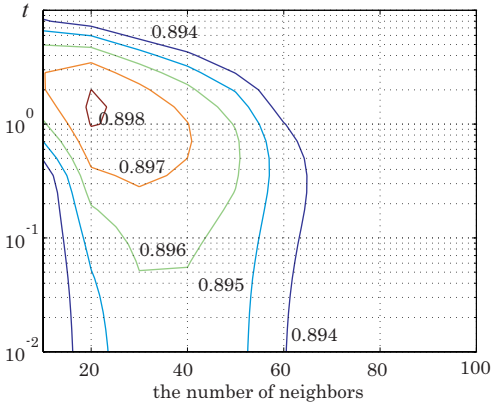

Figure 3. Results of the selected dataset by **NL**

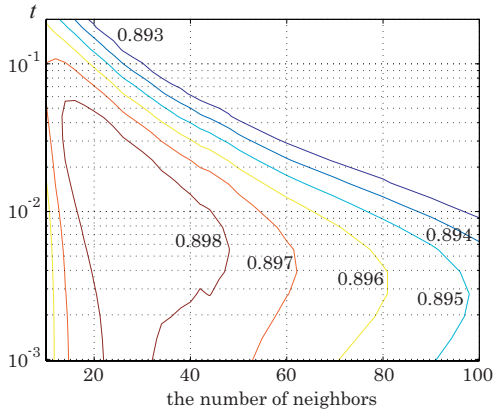

Figure 2. Results of the selected dataset by **RL**

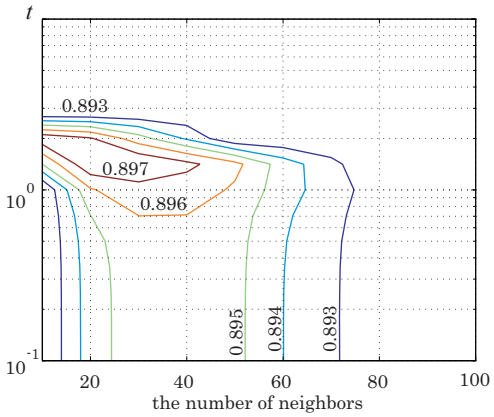

Figure 4. Results of the selected dataset by **DF**

ratings. These results are obtained by the four kernels, **CT**,**RL**,**NL** and **DF**, which are constructed by changing the number of neighbors ranging from $k = 4$ to $k = 100$, as well as the parameter t in (8), (11) and (12), which ranges from $t = 2^{-10}$ to $t = 2^{10}$. Note that the contour lines that are less than 0.893 are omitted from Figs. 2 through 4, and those that are less than 0.911 are omitted from Figs. 5 and 6.

From Figs. 1 through 4, we see that the performance of the three kernels, **RL**, **NL** and **DF**, are almost the same and slightly better than that of **CT**. We can see that the best performance is achieved when the number of neighbors (k) is around 30 for all the kernels. On the other hand, the best results are obtained when the parameter t is around 10^{-2} for **RL**, and $t = 1$ for **NL** and **DF**, both of which are defined by the normalized Laplacian, $\tilde{L} \equiv D^{-1/2}LD^{-1/2}$. It should be emphasized that the proposed method offers fairly high performance in a wide range of parameter settings. Furthermore, Figs. 5 and 6 indicate that almost the same parameter settings generate the highest performance for the case when we use the full dataset.

For comparison, we also perform the same five-fold cross-validation using a previously proposed scoring method by Fouss *et al.* [3], whose results are listed in Table I. We also summarize the highest performance obtained by the four kernels. We can see from this table that, for both selected and full datasets, the three kernels (**RL**, **NL** and **DF**) achieve almost the same performance which are better than **CT**, and significantly better than the method by Fouss *et al.* [3] when we use the full dataset.

VI. Conclusion

We have introduced new methods for recommendation tasks based on the 1-SVM. Using special structures of graph kernels, we show that the 1-SVM can be formulated as rather simple quadratic programming problems. In addition, the formulations can take advantage of the sparsity of the Laplacian matrix, which results in handling recommendation tasks with over one million ratings. Numerical experiments indicate that the quality of our recommendations is high.

TABLE I.
COMPARISON OF THE BEST DEGREE OF AGREEMENTS

Dataset	Fouss *et al.*[3]	**CT**	**RL**	**NL**	**DF**
Selected	0.878	0.880	**0.899**	0.898	0.898
Full	0.889	0.902	**0.920**	0.918	0.918

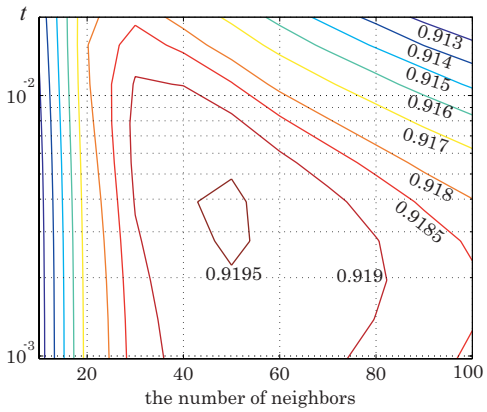

Figure 5. Results of the full dataset by **RL**

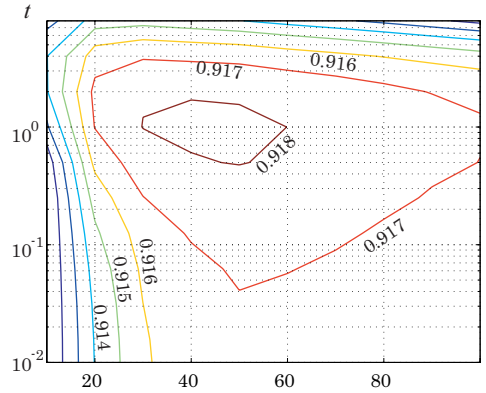

Figure 6. Results of the full dataset by **NL**

ACKNOWLEDGMENTS

This study was supported in part by Grants-in-Aid for Scientific Research (16201032 and 16510106) from JSPS.

REFERENCES

[1] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*. Chapel Hill, North Carolina: ACM, 1994, pp. 175–186.

[2] U. Shardanand and P. Maes, "Social information filtering: Algorithms for automating "word of mouth"," in *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, vol. 1, 1995, pp. 210–217.

[3] F. Fouss, A. Pirotte, and M. Saerens, "A novel way of computing dissimilarities between nodes of a graph, with application to collaborative filtering," in *15th European Conference on Machine Learning (ECML 2004); Proceedings of the Workshop on Statistical Approaches for Web Mining (SAWM)*, 2004, pp. 26–37.

[4] M. Szummer and T. Jaakkola, "Partially labeled classification with Markov random walks," in *Advances in Neural Information Processing Systems*, vol. 14, 2002, pp. 945–952.

[5] A. Smola and I. Kondor, "Kernels and regularization on graphs," in *Proceedings of the Annual Conference on Computational Learning Theory*, ser. Lecture Notes in Computer Science, B. Schölkopf and M. Warmuth, Eds. Springer, 2003.

[6] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Machine Learning*, vol. 56, pp. 209–239, 2004.

[7] T. Ito, M. Shimbo, T. Kudo, and Y. Matsumoto, "Application of kernels to link analysis," in *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. New York, NY, USA: ACM Press, 2005, pp. 586–592.

[8] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, 2001.

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

[10] F. R. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

[11] X. Zhu, J. Lafferty, and Z. Ghahramani, "Semi-supervised learning: From Gaussian fields to Gaussian processes," Carnegie Mellon University, Technical Report CMU-CS-03-175, 2003.

[12] O. L. Mangasarian and M. V. Solodov, "Nonlinear complementarity as unconstrained and constrained minimization," *Math. Programming*, vol. 62, no. 2, Ser. B, pp. 277–297, 1993.

[13] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems*, vol. 16, pp. 321–328, 2004.

[14] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce," in *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*. New York, NY, USA: ACM Press, 2000, pp. 158–167.

Yasutoshi Yajima was born in Nagano, Japan. He received his Ph.D., MS, and BS degrees in industrial engineering and management from Tokyo Institute of Technology in 1993, 1990 and 1988, respectively.

He is currently an Associate Professor of Industrial Engineering and Management at Tokyo Institute of Technology in 1993. His current research interests include numerical optimization, operations research and data mining.

Prof. Yajima is a member of the Operations Research Society of Japan.

Tien-Fang Kuo is currently a Ph.D. candidate at Tokyo Institute of Technology. He received his MS in industrial engineering and management from Tokyo Institute of Technology in 2003, and BS degrees in mathematics from National Taiwan University, Taipei, Taiwan, R.O.C., in 1997. His current research interests include data mining and machine learning.