Pattern Recognition 47 (2014) 1994–2010

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Kernel-based transition probability toward similarity measure for semi-supervised learning

Takumi Kobayashi *

National Institute of Advanced Industrial Science and Technology, Umezono 1-1-1, Tsukuba 305-8568, Japan

ARTICLE INFO

Article history:
Received 28 February 2013
Received in revised form
7 November 2013
Accepted 9 November 2013
Available online 19 November 2013

Keywords:
Semi-supervised learning
Similarity measure
Kernel-based method
Multiple kernel integrated similarity
Multiple kernel learning

ABSTRACT

For improving the classification performance on the cheap, it is necessary to exploit both labeled and unlabeled samples by applying semi-supervised learning methods, most of which are built upon the pair-wise similarities between the samples. While the similarities have so far been formulated in a heuristic manner such as by *k*-NN, we propose methods to construct similarities from the probabilistic viewpoint. The kernel-based formulation of a transition probability is first proposed via comparing kernel least squares to variational least squares in the probabilistic framework. The formulation results in a simple quadratic programming which flexibly introduces the constraint to improve practical robustness and is efficiently computed by SMO. The kernel-based transition probability is by nature favorably sparse even without applying *k*-NN and induces the similarity measure of the same characteristics. Besides, to cope with multiple types of kernel functions, the multiple transition probabilities obtained correspondingly from the kernels can be probabilistically integrated with prior probabilities represented by linear weights. We propose a computationally efficient method to optimize the weights in a discriminative manner. The optimized weights contribute to a composite similarity measure straightforwardly as well as to integrate the multiple kernels themselves as multiple kernel learning does, which consequently derives various types of multiple kernel based semi-supervised classification methods. In the experiments on semi-supervised classification tasks, the proposed methods demonstrate favorable performances, compared to the other methods, in terms of classification performances and computation time.

1. Introduction

The methods of pattern classification have been developed mainly to deal with labeled samples in the framework of supervised learning. In practice, however, it requires exhaustive human labor to assign with labels/annotations especially large-scaled samples. On the other hand, we can cheaply collect only samples without labeling, *i.e.*, *unlabeled* samples, and thus semi-supervised learning methods have attracted keen attention to incorporate such unlabeled samples for classification [1].

There are mainly two approaches for the semi-supervised classification. One is based on co-training [2]. The method of co-training starts with the initial classifier which is usually learnt by using a few labeled samples and then gradually adds unlabeled samples into the set of labeled samples in an iterative manner based on the classification results over those unlabeled samples. The co-training works well such as on car detection [3], video-concept detection [4] and multi-view learning [5]. It, however, is affected by both the initial classification and the way of turning the unlabeled samples into labeled ones through iterations, being amenable to local minima. The other line for semi-supervised learning is built on a graph structure consisting of samples (nodes) linked each other by weighted edges according to the pair-wise similarities [6]. The unlabeled samples are naturally incorporated into the graph and the optimization problems are formulated by utilizing the energy over the graph; for example, the label propagation methods [7–10] directly estimate the labels of the unlabeled samples by minimizing the graph energy with the information of the labeled samples, and the other semi-supervised methods can be developed by incorporating the graph energy as a regularization to the optimization problem defined in the supervised manner [11–14]. The global structure (manifold) of samples is exploited via the graph without resorting to iterative classification that the co-training is based on, nor the model-based marginal probability of samples [15]. And the optimization problem containing minimization of the graph energy is usually formulated in a convex form with the global optimum. In these graph-based semi-supervised methods, the similarity measure is fundamental for connecting labeled and unlabeled samples on the manifold of sample distribution. Thus, how to construct the similarities for improving the performance is an important issue.

The most commonly used similarity is measured by the Gaussian kernel $\exp(-(1/h)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2)$ on neighboring samples, called Gaussian kernel similarity (GKS). This is an ad hoc model

* Tel.: +81 29 861 5491; fax: +81 29 861 3313.
E-mail address: takumi.kobayashi@aist.go.jp

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.11.011

solely depending on the Euclidean distance between sample feature vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The bandwidth parameter h and the number of neighbors have to be determined in advance, requiring exhaustive manual tuning. In recent years, more sophisticated methods have been proposed for the similarities by considering the linear relationship among sample vectors [8,9]. The models employed in those methods, however, are derived somewhat heuristically. There are some other works [10,12] to construct similarities by improving the GKS, and we briefly review them in the next section.

In this paper, we propose methods to construct the pair-wise similarity measure from the probabilistic viewpoint for boosting performance of the graph-based semi-supervised methods. In the probabilistic framework, by comparing the kernel least squares [16] with the variational least squares [17] that gives Bayesian optimal solution, we first propose the kernel-based formulation to approximate the transition probabilities between samples. The kernel-based transition probability (KTP), which can also be interpreted from algebraic and geometric viewpoints, is essential for inducing subsequent methods regarding similarity measure. By using a single kernel function, the KTP is formulated as the simple quadratic programming that is solved in quite a low computation time via sequential minimal optimization [18] and that enables us to introduce some constraints so as to provide favorably sparse probabilities circumventing some practical issues. Those probabilities are turned into the (symmetric) pair-wise similarity measure based on the probabilistic metric. The proposed similarity is inherently sparse due to the favorable sparseness of KTP without resorting to k-NN.

In real-world classification problems, multiple types of kernel function are often available for improving performance such as by extracting multiple features. We also propose a novel method to integrate the multiple kernel functions through the KTP. Once transforming the respective kernel functions to the forms of KTP correspondingly, those KTPs are probabilistically integrated with prior probabilities represented by linear weights. We efficiently optimize the integration weights in a discriminative manner, as in multiple kernel learning (MKL) [19], and thus the multiple transition probabilities are effectively combined into a new composite one. It is straightforward to derive a novel similarity from the composite KTP, and besides we further exploit the optimized weights for integrating the kernel functions themselves to render novel multiple-kernel methods. We present various types of multiple-kernel semi-supervised methods and compare them thoroughly in the experiments, showing that the proposed method of multiple kernel integration favorably works even in comparison to the MKL methods.

Our contributions are summarized as follows:

1. We formulate (constrained) kernel-based transition probabilities by comparing the kernel least squares to the variational one in the probabilistic framework.
2. Inherently sparse similarity measure is constructed from the kernel-based transition probabilities without requiring ad hoc k-NN.
3. We propose a novel method for integrating multiple kernels into the novel similarity via the probabilistic formulation.
4. Based on the multiple kernel integration, various types of multiple kernel semi-supervised classification methods are presented and thoroughly evaluated in the experiments.

The rest of this paper is organized as follows: in the next section, we briefly review the related works in terms of designing similarity measure and integrating multiple kernels in the semi-supervised framework. Section 3 details the kernel-based transition probability, and Section 4 shows the formulation of the proposed similarity. Consequently, we propose the method for integrating multiple kernels and derive multiple kernel semi-supervised methods in Section 5. The experimental results on semi-supervised classification tasks are shown in Section 6, and finally Section 7 contains our concluding remarks.

This paper is extended from the ECCV2012 conference paper [20], including the substantial improvements mainly in that multiple kernel semi-supervised methods are proposed and thoroughly compared in the experiments from various aspects. The minor improvements are related to the kernel-based transition probability, such as the algebraic interpretation and the extension to the constrained version.

Throughout this paper, we use the following notations; the bold lower cases for representing vectors, *e.g.*, $\boldsymbol{x}$, bold upper cases for matrices, *e.g.*, $\boldsymbol{K}$, and normal cases for scalar elements in the vector or matrix, *e.g.*, x_i and K_{ij}, with the index subscript. As to functions, the ones that produce a scalar output are represented by k, p and q, while their bold notations, *e.g.*, $\mathbf{p}$, denote the functions outputting a vector.

2. Related works

We first mention the graph Laplacian [6] on which the graph-based semi-supervised methods [7–14] are built. Let S_{ij} denote the similarity between the i, j-th samples and $\mathsf{f}(\boldsymbol{x})$ be the projection function, such as classifier, of the sample $\boldsymbol{x}$. The projections of samples are expected to be close according to the similarity measure, which corresponds to minimize the following energy:

$$\frac{1}{2}\sum_{ij}^{n} S_{ij}|\mathsf{f}(\boldsymbol{x}_i)-\mathsf{f}(\boldsymbol{x}_j)|^2 = [\mathsf{f}(\boldsymbol{x}_1),\ldots,\mathsf{f}(\boldsymbol{x}_n)](\boldsymbol{D}-\boldsymbol{S})[\mathsf{f}(\boldsymbol{x}_1),\ldots,\mathsf{f}(\boldsymbol{x}_n)]^{\top}, \tag{1}$$

where $\boldsymbol{S}\in\mathbb{R}^{n\times n}$ is the similarity matrix and $\boldsymbol{D}\in\mathbb{R}^{n\times n}$ is the diagonal matrix of $\{D\}_{ii}=\sum_j^n S_{ij}$. The similarity measure works through the graph Laplacian $\boldsymbol{L}\triangleq\boldsymbol{D}-\boldsymbol{S}$ for the graph-based semi-supervised methods that utilize (1) as a regularization.

There are some works to formulate the similarity itself other than GKS. The linear neighborhood propagation (LNP) [9] has presented a similar formulation to ours in a *linear* input space. The method somewhat heuristically assumes that a sample vector is approximated by using its neighbors in a linear form, while in the proposed method we derive the kernel-based transition probabilities via considering kernel least squares in the probabilistic framework, which also induces the method for probabilistically integrating multiple kernels. In addition, we provide (1) the computationally efficient method to compute them by using SMO, (2) algebraic/geometrical characteristics to endow the sparsity inherently without ad hoc k-NN, and (3) the constrained kernel-based transition probability. The kernel LNP has also been proposed in [21] but it differs from our method in that it lacks a probabilistic constraint. Cheng et al. [8] applied the compressed sensing approach to construct sparse similarities by assuming the (strict) linear dependency $\boldsymbol{x}=\sum_i\alpha_i\boldsymbol{x}_i$ as in [9]. Such linear dependency (equality), however, is a too strong constraint to hold, especially in a high dimensional feature space. Kobayashi and Otsu [22] proposed the cone-based formulation of similarity measure for one-class classification. In that method, the local cone composed of neighbor samples is applied to each sample, but to achieve meaningful similarity, the locality is appropriately defined in advance such as by k-NN. Zhang and Yeung [12] has proposed the path-based similarity which is measured by searching the optimum path in min–max criterion on the initial graph. However, a problem remains on how to construct the initial graph (similarity), and the authors employ GKS. The parameter settings in GKS still affect the performances of the resulting similarity.

Liu and Chang [10] proposed an interesting method to produce a discriminative similarity. In that method, the similarity is sequentially updated by using given label information, although the method also starts from the GKS-based initial graph. Note that the proposed similarity (Section 4) other than multiple kernel integration is computed in *unsupervised* manner; it is subsequently fed into semi-supervised methods for classification. From the viewpoint of optimizing the similarity with a small amount of given label information, the method [10] is slightly close to the proposed method that integrates multiple kernels (Section 5). It, however, should be noted that the method to construct similarity measure from the multiple kernels has been rarely addressed so far in the framework of semi-supervised learning, except for [23,24].

The proposed method for integrating similarities derived from multiple kernels is closely related to [23]. The method in [23] combines multiple types of similarity measure in the framework of label propagation [7], which results in convex optimization problem. However, the LP-based optimization is dominated by the graph energy minimization, being less discriminative compared to the multiple kernel learning (MKL) [19,25]. Wang et al. [24] proposed the method of semi-supervised MKL, though it requires prior knowledge regarding class categories and is specific to the exponential family parametric model. Our proposed method contributes to integrate not only the similarities but also the kernel functions, and thereby several kinds of multiple kernel semi-supervised methods are proposed in general forms without relying on any specific knowledge.

In the Gaussian process (GP) [16], the kernel least squares also emerge in a probabilistic manner. But, the GP assumes the parametric (Gaussian) model for the whole samples and it cannot give explicit connection to the pair-wise transition probability that is our main concern for inducing similarity measure. In this paper, by comparing the kernel-based and the variational approaches based on the identical least-square criterion in the probabilistic framework, we propose the kernel-based transition probability (Section 3). The proposed transition probability benefits to construction of the similarity (Section 4) as well as multiple kernel integration (Section 5).

3. Kernel-based transition probability

Let us fist consider the regression problem from the feature $\boldsymbol{x}$ to the labels $\boldsymbol{y}$, which is formulated based on the least-square criterion. As shown in Fig. 1, we begin with the least squares in the probabilistic framework, followed by the variational optimization [17] and kernel-based approach [16], and then finally compare those two solutions to induce the kernel-based formulation of the transition probabilities between samples, called the kernel-based transition probability (KTP). Note that the identical least square criterion is employed in both methods so as to make the comparison reasonable. The proposed KTP is fundamental not only for the similarity measure (Section 4) but also for the multiple kernel integration (Section 5).

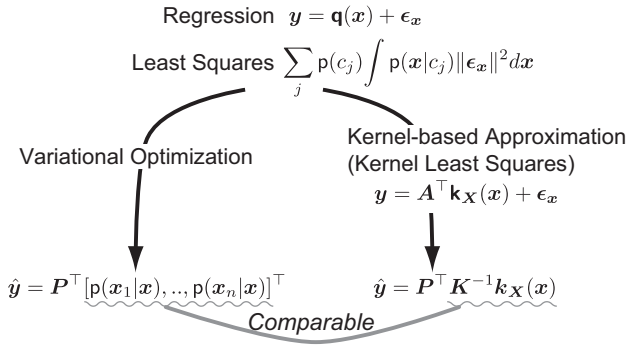


Fig. 1. Flow chart for inducing the kernel-based transition probability (KTP). Starting with the identical least squares, both the variational and the kernel-based approaches are compared.

3.1. Least squares in probabilistic framework

Let $\boldsymbol{x} \in \mathbb{R}^D$ be the input vector and $\boldsymbol{y} \in \mathbb{R}^C$ be a (multiple) objective variable(s) associated with $\boldsymbol{x}$. The regression model is generally formulated as $\boldsymbol{y} = \mathbf{q}(\boldsymbol{x}) + \boldsymbol{\epsilon}$ using a non-linear function $\mathbf{q}$ with the residual errors $\boldsymbol{\epsilon}$. Here, we suppose a C-class problem. Let c_j $(j = 1, \ldots, C)$ denote the j-th class and $\boldsymbol{e}_j$ be the C-dimensional binary vector representing the j-th class, in which only the j-th element is 1 and the others are 0. Suppose those class-representative vectors $\boldsymbol{e}_j$ are targets, $\boldsymbol{y} \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_C\}$. Thus, the regression function $\mathbf{q}$ is optimized based on the following least squares:

$$E[\|\boldsymbol{\epsilon}\|^2] = J(\mathbf{q}) \triangleq \sum_j^C \mathrm{p}(c_j) \int \mathrm{p}(\boldsymbol{x}|c_j)\|\boldsymbol{e}_j - \mathbf{q}(\boldsymbol{x})\|^2 \, d\boldsymbol{x} \to \min_{\mathbf{q}}. \tag{2}$$

By applying the variational method, we obtain the optimum form of $\mathbf{q}$ [17] by

$$\delta J = J(\mathbf{q} + \delta\mathbf{q}) - J(\mathbf{q}) = 2 \int \delta\mathbf{q}(\boldsymbol{x})^\top \left[\sum_j^C \mathrm{p}(c_j)\mathrm{p}(\boldsymbol{x}|c_j)\{\boldsymbol{e}_j - \mathbf{q}(\boldsymbol{x})\} \right] d\boldsymbol{x}, \tag{3}$$

$$\Rightarrow \mathbf{p}(\boldsymbol{x}, \boldsymbol{c}) - \mathrm{p}(\boldsymbol{x})\mathbf{q}(\boldsymbol{x}) = \mathbf{0}, \quad \therefore \ \mathbf{q}(\boldsymbol{x}) = [\mathrm{p}(c_1|\boldsymbol{x}), \ldots, \mathrm{p}(c_m|\boldsymbol{x})]^\top = \mathbf{p}(\boldsymbol{c}|\boldsymbol{x}), \tag{4}$$

where we use $\mathrm{p}(\boldsymbol{x}, c_j) = \mathrm{p}(c_j)\mathrm{p}(\boldsymbol{x}|c_j)$, $\mathrm{p}(\boldsymbol{x}) = \sum_j^C \mathrm{p}(\boldsymbol{x}, c_j)$, and $\mathbf{p}(\boldsymbol{x}, \boldsymbol{c}) = [\mathrm{p}(\boldsymbol{x}, c_1), \ldots, \mathrm{p}(\boldsymbol{x}, c_C)]^\top \in \mathbb{R}^C$. The regression function results in the posterior probabilities for the classes and it is further decomposed by using the finite sample set $\{\boldsymbol{x}_i\}_{i=1,\ldots,n}$ as follows:

$$\boldsymbol{y} \approx \mathbf{p}(\boldsymbol{c}|\boldsymbol{x}) = \int \mathbf{p}(\boldsymbol{c}|\tilde{\boldsymbol{x}})\mathrm{p}(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \, d\tilde{\boldsymbol{x}} \tag{5}$$

$$\approx \sum_i^n \mathbf{p}(\boldsymbol{c}|\boldsymbol{x}_i)\mathrm{p}(\boldsymbol{x}_i|\boldsymbol{x}) = \boldsymbol{P}^\top [\mathrm{p}(\boldsymbol{x}_1|\boldsymbol{x}), \ldots, \mathrm{p}(\boldsymbol{x}_n|\boldsymbol{x})]^\top, \tag{6}$$

where $\boldsymbol{P} \in \mathbb{R}^{n \times C}$ is a posterior probability matrix of $P_{ij} = \mathrm{p}(c_j|\boldsymbol{x}_i)$, $\boldsymbol{P} = [\mathbf{p}(\boldsymbol{c}|\boldsymbol{x}_1), \ldots, \mathbf{p}(\boldsymbol{c}|\boldsymbol{x}_n)]^\top$. Though it is well-known that the optimal classifier is the posterior probability from the Bayesian viewpoint, it should be noted here that the posterior probability is induced from the above least squares criterion via the variational approach.

On the other hand, from the practical viewpoint, the regression function $\mathbf{q}$ is often approximated by $\mathbf{q} \approx \boldsymbol{A}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x})$ using a kernel function k and $\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}) = [\mathrm{k}(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, \mathrm{k}(\boldsymbol{x}_n, \boldsymbol{x})]^\top \in \mathbb{R}^n$ with the coefficients $\boldsymbol{A} \in \boldsymbol{R}^{n \times C}$. In this case, the target to be optimized is the matrix $\boldsymbol{A}$ in the kernel least squares:

$$J(\boldsymbol{A}) = \sum_j^C \mathrm{p}(c_j) \sum_i^n \mathrm{p}(\boldsymbol{x}_i|c_j)\|\boldsymbol{e}_j - \boldsymbol{A}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)\|^2 \tag{7}$$

$$= \mathrm{trace}\,(\boldsymbol{A}^\top \boldsymbol{K}\boldsymbol{\Lambda}\boldsymbol{K}\boldsymbol{A} - 2\boldsymbol{A}^\top \boldsymbol{K}\boldsymbol{\Theta} + 1) \to \min_{\boldsymbol{A}} \tag{8}$$

$$\therefore \ \boldsymbol{A} = \boldsymbol{K}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Theta}, \tag{9}$$

where $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is a (nonsingular) kernel Gram matrix of $K_{ij} = \mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\boldsymbol{\Lambda} = \mathrm{diag}([\mathrm{p}(\boldsymbol{x}_1), \ldots, \mathrm{p}(\boldsymbol{x}_n)]) \in \mathbb{R}^{n \times n}$, $\boldsymbol{\Theta} = [\mathbf{p}(\boldsymbol{x}_i, \boldsymbol{c}), \ldots, \mathbf{p}(\boldsymbol{x}_n, \boldsymbol{c})]^\top \in \mathbb{R}^{n \times C}$. By using $\mathrm{p}(c_j|\boldsymbol{x}_i) = \mathrm{p}(\boldsymbol{x}_i, c_j)/\mathrm{p}(\boldsymbol{x}_i)$, we finally obtain the following regression form:

$$\boldsymbol{A} = \boldsymbol{K}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Theta} = \boldsymbol{K}^{-1}\boldsymbol{P}, \quad \boldsymbol{y} \approx \boldsymbol{P}^\top \boldsymbol{K}^{-1}\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}). \tag{10}$$

Details of these derivations are shown in Appendix A.

Both of the abovementioned results (6) and (10) are comparable since they are derived from the identical least-square criterion, and thus we can find the kernel-based approximation of the transition probabilities:

$$[\mathrm{p}(\boldsymbol{x}_1|\boldsymbol{x}),\ldots,\mathrm{p}(\boldsymbol{x}_n|\boldsymbol{x})]^\top \approx \boldsymbol{K}^{-1}\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}) \triangleq \boldsymbol{\alpha} \in \mathbb{R}^n. \tag{11}$$

The right-hand side $\boldsymbol{\alpha}$, however, might take any values, while the transition probabilities in the left-hand side are subject to the probabilistic constraints of non-negativity $0 \le \mathrm{p}(\boldsymbol{x}_i|\boldsymbol{x})(\le 1)$ and unit sum $\sum_i \mathrm{p}(\boldsymbol{x}_i|\boldsymbol{x}) = 1$. In what follows, we impose these probabilistic constraints on $\boldsymbol{\alpha}$ in order to approximate the transition probability more accurately from the probabilistic perspective.

3.2. Definition of kernel-based transition probability

The vector $\boldsymbol{\alpha} \triangleq \boldsymbol{K}^{-1}\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x})$ can also be regarded as the solution of the following regression in reproducing kernel Hilbert space (RKHS):

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\Phi}_{\mathbf{X}}\boldsymbol{\alpha} - \boldsymbol{\phi}_{\boldsymbol{x}}\|^2 = (\boldsymbol{\Phi}_{\boldsymbol{X}}^\top \boldsymbol{\Phi}_{\mathbf{X}})^{-1}\boldsymbol{\Phi}_{\boldsymbol{X}}^\top \boldsymbol{\phi}_{\boldsymbol{x}} = \boldsymbol{K}^{-1}\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}), \tag{12}$$

where $\boldsymbol{x}$ is represented by $\boldsymbol{\phi}_{\boldsymbol{x}}$ in RKHS which is endowed with the inner product $\boldsymbol{\phi}_{\boldsymbol{x}_i}^\top \boldsymbol{\phi}_{\boldsymbol{x}_j} = \mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{\Phi}_{\mathbf{X}} = [\boldsymbol{\phi}_{\boldsymbol{x}_1},\ldots,\boldsymbol{\phi}_{\boldsymbol{x}_n}]$. We impose the probabilistic constraints on (12) to propose the kernel-based formulation for approximating the transition probabilities more accurately:

$$\min_{0\le\boldsymbol{\alpha}\le 1, \sum_i \alpha_i = 1} \|\boldsymbol{\Phi}_{\mathbf{X}}\boldsymbol{\alpha} - \boldsymbol{\phi}_{\boldsymbol{x}}\|^2 \Leftrightarrow \min_{0\le\boldsymbol{\alpha}\le 1, \sum_i \alpha_i = 1} \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}). \tag{13}$$

This may be viewed as a kernel-based extension of LNP [9], but is different from the kernel LNP [21] which lacks the probabilistic constraint $0 \le \boldsymbol{\alpha} \le 1$. We call the optimizer $\boldsymbol{\alpha}$ in (13) as the *kernel-based transition probability* (KTP). In this study, we use the kernel function that is normalized in unit norm, *i.e.*, $\mathrm{k}(\boldsymbol{x},\boldsymbol{x}) = 1,\ \forall \boldsymbol{x}$, to render the desirable property which is described below. Note that the kernel function is often (or inherently) normalized, *e.g.*, in Gaussian kernel $\mathrm{k}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp(-(1/h)\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$.

Eq. (13) produces favorably sparse KTP values $\boldsymbol{\alpha}$ even though neither the regularization nor the constraint regarding the sparseness of $\boldsymbol{\alpha}$ is introduced in that formulation. The "favorable sparseness" means that $\boldsymbol{\alpha}$ consists of *sparse* non-zero components assigned to the *neighbor* samples; for instance, Fig. 3 illustrates the favorable sparseness of the KTP compared to the Gaussian kernel. To discuss such a property of the KTP, we give two interpretations from the algebraic and the geometric perspectives, which have not been mentioned so far.

Algebraic interpretation: The formulation (13) is analogous to the dual (quadratic-)problem of SVM [26]. To make it clear, we newly introduce the following primal problem that has the dual in the form of (13), as

$$\begin{aligned} \min_{\boldsymbol{v},\xi,\{\xi_i\}_i}\ & \frac{1}{2}\|\boldsymbol{v}\|^2 + \xi + C\sum_i^n \xi_i \\ \text{s.t.}\ & \boldsymbol{v}^\top\boldsymbol{\phi}_{\boldsymbol{x}} + b \ge 1 - \xi,\ \ \forall i,\quad \boldsymbol{v}^\top\boldsymbol{\phi}_{\boldsymbol{x}_i} + b \le -1 + \xi_i,\ \ \xi_i \ge 0, \end{aligned} \tag{14}$$

where ξ, ξ_i are the slack variables and C is the (cost) parameter which is set as $C=1$ to get (13). This formulation is similar to SVM in that the sample $\boldsymbol{\phi}_{\boldsymbol{x}}$ assigned with positive class $y=+1$ is linearly separated from the other samples $\boldsymbol{\phi}_{\boldsymbol{x}_i}$ of negative class $y_i=-1$, though the slack variable ξ for $\boldsymbol{\phi}_{\boldsymbol{x}}$ is slightly differently defined from SVM[1]; see Appendix B for the primal–dual relationship between (14) and (13). The samples of $\alpha_i > 0$ are the support vectors composing the hyperplane to separate $\boldsymbol{\phi}_{\boldsymbol{x}}$ from the others $\boldsymbol{\phi}_{\boldsymbol{x}_i}$, which thus form a *sparse* set. Besides, in the case that the samples are distributed on a convex surface, such as a unit hypersphere, those support samples are located *near* to the target sample $\boldsymbol{\phi}_{\boldsymbol{x}}$. We only apply the normalized kernel function to achieve such favorable property for the KTP without resorting to k-NN.

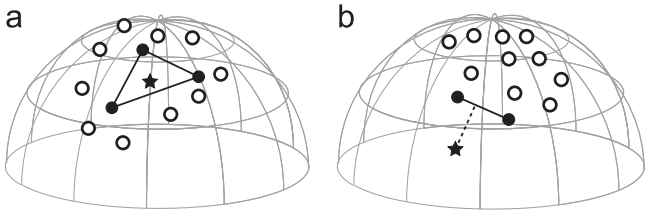


Fig. 2. Geometrical interpretation of KTP. Circle points denote samples and the star point is an input sample in RKHS. Only black dots have non-zero weights α_i in (13), and black solid lines show the contour of the convex hull spanned by those black points. (a) Inside and (b) Outside.

Geometric interpretation: Suppose that all the samples $\boldsymbol{\phi}_{\boldsymbol{x}}$ and $\boldsymbol{\phi}_{\boldsymbol{x}_i}$ lie on the unit hypersphere in RKHS via the normalized kernel function, $\mathrm{k}(\boldsymbol{x},\boldsymbol{x}) = \boldsymbol{\phi}_{\boldsymbol{x}}^\top\boldsymbol{\phi}_{\boldsymbol{x}} = 1,\ \forall \boldsymbol{x}$. The optimization (13) is also regarded as the projection from $\boldsymbol{\phi}_{\boldsymbol{x}}$ to the convex hull[2] that are spanned by the sample vectors $\boldsymbol{\Phi}_{\mathbf{X}}$. When $\boldsymbol{\phi}_{\boldsymbol{x}}$ is contained in the convex cone by $\boldsymbol{\Phi}_{\mathbf{X}}$, the small hull is selected to minimize the distance from $\boldsymbol{\phi}_{\boldsymbol{x}}$ to the hull (Fig. 2a). On the other hand, when $\boldsymbol{\phi}_{\boldsymbol{x}}$ lies outside the convex cone, the hull by the basis sample vectors closer to $\boldsymbol{\phi}_{\boldsymbol{x}}$ is selected (Fig. 2b). Thus, KTP has only a few non-zero components associated with the samples that span such a convex hull nearby the target $\boldsymbol{\phi}_{\boldsymbol{x}}$.

3.3. Constrained KTP

The KTP is inherently sparse as mentioned above, but such sparsity might occasionally harm classification performance for the case that samples are close enough to each other, *i.e.*, nearly duplicated. This is illustrated in Fig. 4 where the samples indicated by 'a' and 'f' are closely located. The KTP values from the samples 'a' and 'f' are shown in the left column of Fig. 4, indicating that the nearly duplicated sample dominates the transition probabilities though there are some other close samples. In such a case, those duplicated samples 'a' and 'f' are connected too strongly and are unfavorably isolated from the others, not reflecting the intrinsic neighborhood (manifold) structure. To alleviate it, we introduce into the KTP the upper bound controlled by the parameter $\nu(\ge 1)$ as

$$\min_{0\le\boldsymbol{\alpha}\le 1/\nu, \sum_i \alpha_i = 1} \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}). \tag{15}$$

In terms of $\sum_i \alpha_i = 1$, the number of non-zero KTPs is larger than ν, *i.e.*, $\|\boldsymbol{\alpha}\|_0 \ge \nu$ (see Appendix C). Thus, the parameter ν implies the lower bound for the number of non-zeros. The middle column of Fig. 4 shows the case of $\nu = 2$ where the samples other than the duplicate one are also assigned with favorably high KTP values as is the case with that the duplicated sample is excluded (the right column in Fig. 4). In this paper, we use $\nu = 2$, the effectiveness of which is validated in the experiment.

Finally, we address the practical issue for computing (15). It is a convex quadratic programming (QP) which is usually solved by using standard QP solvers, such as MOSEK optimization toolbox.[3] However, it requires significant computational cost, making the

[1] One might think that the original SVM is applied to produce KTP $\boldsymbol{\alpha}$ instead of (13). We empirically checked that it results in almost the same results as the proposed method without any performance difference.

[2] Such convex hull is the intersection between the hypersphere and the hyperplane produced by (14).

[3] The MOSEK optimization software http://www.mosek.com/.

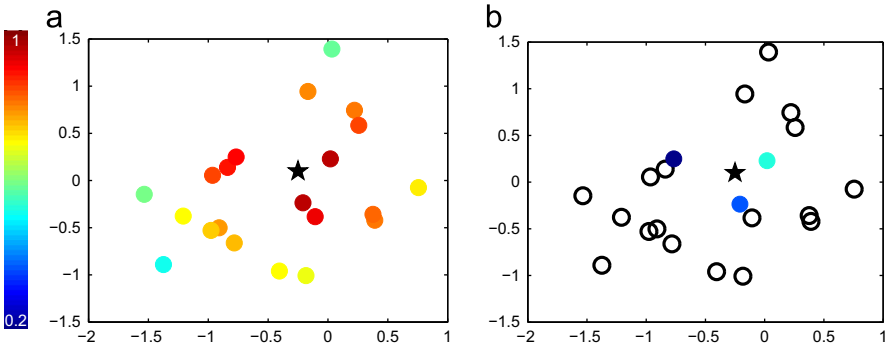


Fig. 3. KTP when using Gaussian kernel. The kernel values and KTP are shown in (a) and (b), respectively. The reference (input) point is denoted by the star. Pseudo colors indicate the values at the neighbor points, and the uncolored points in (b) have zero KTP. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

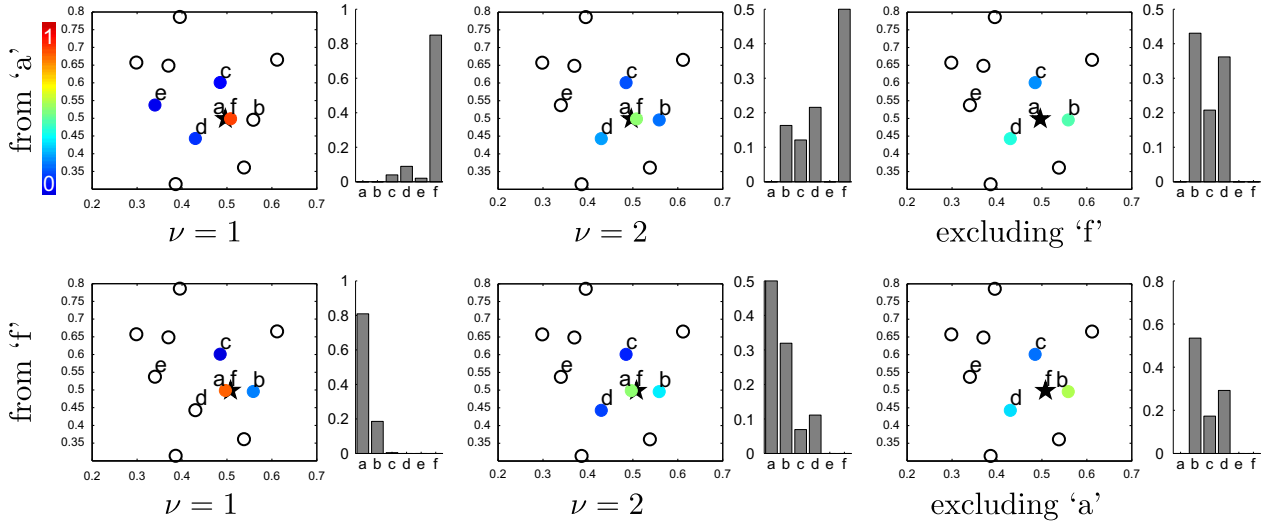


Fig. 4. Toy example for constrained KTP. 'a'–'f' indicate the sample indices, and the samples of 'a' and 'f' are nearly duplicated. Top row shows KTP from the sample 'a', and bottom row is from 'f', where the KTP values are shown in pseudo-colored scatter plot and in bar plot. Left column shows the case of KTP in (13), *i.e.*, $\alpha \le 1$, the middle column is for constrained KTP in (15) with $\nu = 2$, *i.e.*, $\alpha \le \frac{1}{2}$, and the right column is the case when the duplicated sample is excluded.

method inapplicable to large-scaled samples. As mentioned in Section 3.2, Eq. (15) is similar to the dual of SVM, especially almost the same as the dual problem in SVDD [27] except for the linear term of $\boldsymbol{\alpha}$, which is formulated by

$$\text{SVDD}: \min_{0 \le \boldsymbol{\alpha} \le 1/\nu, \sum_i \alpha_i = 1} \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top [\mathrm{k}(\boldsymbol{x}_1, \boldsymbol{x}_1), \ldots, \mathrm{k}(\boldsymbol{x}_n, \boldsymbol{x}_n)]^\top. \quad (16)$$

Various approaches have been developed to efficiently solve the dual problem [28], and in this study, we apply the sequential minimal optimization (SMO) approach [18], implemented in LIBSVM [29], due to which the proposed method is computationally efficient and thus applicable to large-scaled samples.

4. KTP-based similarity

The similarity measure is derived from the kernel-based transition probabilities (KTPs) in (15). We first calculate the KTP $\boldsymbol{\alpha}$ from respective $\boldsymbol{x}_i$, $i = 1, \ldots, n$ to the others in a leave-one-out scheme; at the i-th sample, Eq. (15) is solved for $\boldsymbol{\Phi}_{\boldsymbol{X}} = [\ldots, \boldsymbol{\phi}_{\boldsymbol{x}_{i-1}}, \boldsymbol{\phi}_{\boldsymbol{x}_{i+1}}, \ldots]$ and $\boldsymbol{\phi}_{\boldsymbol{x}_i}$ to produce $\boldsymbol{\alpha}_i \in \mathbb{R}^n$ in which $\alpha_{ji} = \mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)$ and $\alpha_{ii} = 0$,[4] finally gathered into $\boldsymbol{P} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]^\top, P_{ij} = \mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)$. For speeding up the computation of KTP, we can apply a preprocessing of k-NN with somewhat larger k to reduce the size of (15) since only a small portion of neighbor samples has actually non-zero KTP as discussed in the previous section. The procedure for computing KTP is shown in Algorithm 1.

Then, we define the metric between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ based on the transition probability (*information*) by

$$D(\boldsymbol{x}_j \| \boldsymbol{x}_i) = -\log\, \mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i), \quad \overline{D}(\boldsymbol{x}_i, \boldsymbol{x}_j) = D(\boldsymbol{x}_j \| \boldsymbol{x}_i) + D(\boldsymbol{x}_i \| \boldsymbol{x}_j). \quad (17)$$

This is a symmetric metric as in symmetrized Kullback–Leibler divergence [30]. By using this metric, the similarity is formulated as

$$S_{ij} \triangleq \exp\left\{-\frac{1}{\lambda}\overline{D}(\boldsymbol{x}_i, \boldsymbol{x}_j)\right\} = \{\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)\mathrm{p}(\boldsymbol{x}_i|\boldsymbol{x}_j)\}^{1/\lambda} = (P_{ij}P_{ji})^{1/\lambda}, \quad (18)$$

where the bandwidth parameter is simply set as $\lambda = 1$ which is empirically validated in the experiment, and thereby the similarity matrix is

$$\boldsymbol{S} = (\boldsymbol{P} \circ \boldsymbol{P}^\top)^{1/\lambda} \in \mathbb{R}^{n \times n}, \quad (19)$$

where $\circ$ denotes the Hadamard product. This KTP-based similarity, called KTPS, ranges from 0 to 1 and the sparseness is further enhanced than the KTP since $S_{ij} > 0$ *iff* $P_{ij} > 0 \wedge P_{ji} > 0$.

Algorithm 1. Kernel-based transition probability.

Input: $\boldsymbol{K} \in \mathbb{R}^{n \times n}$: normalized Kernel Gram matrix, *i.e.*, $K_{ii} = 1, \ \forall i$,
ν: parameter in KTP, usually $\nu = 2$

[4] Since the self similarity does not affect the graph Laplacian [6], we simply set $\alpha_{ii} = 0$.

1: **for** $i=1$ to n **do**

2: $\begin{cases} (k\text{-NN search}) & \text{find the first } k \text{ sample indices that have larger } K_{ij}; \\ & \mathbf{J} = \arg\mathrm{k}-\max_{j\in\{1,\ldots,i-1,i+1,\ldots,n\}} K_{ij} \\ \text{or (full search)} & \mathbf{J} = \{1,\ldots,i-1,i+1,\ldots,n\} \end{cases}$

3: $\tilde{\boldsymbol{K}} = \{K_{jj'}\}_{j\in\mathbf{J}}^{j'\in\mathbf{J}} \in \mathbb{R}^{|\mathbf{J}|\times|\mathbf{J}|}$, $\tilde{\boldsymbol{k}} = \{K_{ji}\}_{j\in\mathbf{J}} \in \mathbb{R}^{|\mathbf{J}|}$

4: $\boldsymbol{\alpha} = \arg\min_{0\le\boldsymbol{\alpha}\le\frac{1}{\nu},\mathbf{1}^\top\boldsymbol{\alpha}=1} \frac{1}{2}\boldsymbol{\alpha}^\top\tilde{\boldsymbol{K}}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top\tilde{\boldsymbol{k}}$

5: $P_{ij} = \begin{cases} 0 & j\notin\mathbf{J} \\ \alpha_l & j\in\mathbf{J}, l \text{ is the order of } j \text{ in the set } \mathbf{J},\ i.e.,\ \mathbf{J}_l = j \end{cases}$

6: **end for**

Output: $\boldsymbol{P} = \{P_{ij}\}_{i=1,\ldots,n}^{j=1,\ldots,n} \in \mathbb{R}^{n\times n}$: transition probability matrix, $P_{ij} = \mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)$

5. Multiple kernel integration

We have proposed the kernel-based transition probability (KTP) and consequently KTPS by using a single type of kernel function. In practical classification tasks, multiple types of kernel functions are naturally available rather than only a single type, such as by extracting multiple types of features. In such a case, we obtain multiple transition probabilities (KTP) correspondingly, and as in multiple kernel learning (MKL) [19], it is desirable to integrate those multiple KTPs such that the resulting KTPS has high discriminative power. In this section, we propose the method for linearly integrating the KTPs derived from multiple kernels into the novel KTPS (Section 5.3) and then present the classification method using the multiple types of kernels as well as the composite KTPS (Section 5.3).

5.1. Multiple kernel KTP and KTPS

Suppose M types of kernel functions are given, $\mathrm{k}^{[l]}$, $l=1,\ldots,M$. Let $\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i,\mathrm{k}^{[l]}) = P_{ij}^{[l]}$ be the transition probability conditioned on the l-th type of kernel function $\mathrm{k}^{[l]}$. From the probabilistic viewpoint, those probabilities are integrated by

$$\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i) = \sum_{l=1}^{M} \mathrm{p}(\mathrm{k}^{[l]})\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i,\mathrm{k}^{[l]}) = \sum_{l=1}^{M} \omega_l P_{ij}^{[l]}, \tag{20}$$

where $\mathrm{p}(\mathrm{k}^{[l]})$ is the prior probability of the l-th type of kernel and it is regarded as the linear weight ω_l to be optimized subject to $\omega_l \ge 0$, $\sum_l \omega_l = 1$. We optimize it in a discriminative manner using (a small amount of) labeled samples since there is no any prior knowledge of $\omega_l \triangleq \mathrm{p}(\mathrm{k}^{[l]})$.

Let the set of labeled samples be denoted by $\mathbf{G}$. The labeled sample pairs in $\mathbf{G}\times\mathbf{G}$ are categorized into $\mathbf{P} = \{(i,j)|c_i = c_j, i,j\in\mathbf{G}\}$ and $\mathbf{N} = \{(i,j)|c_i \ne c_j, i,j\in\mathbf{G}\}$, where c_i indicates the class label of the i-th sample $\boldsymbol{x}_i$. For each labeled sample $i\in\mathbf{G}$, from the discriminative perspective, it is expected that the transition probability to the same class, $\sum_{j|(i,j)\in\mathbf{P}}\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)$, be maximized, while minimizing the probabilities to the different classes, $\sum_{j|(i,j)\in\mathbf{N}}\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)$. Thus, we define the following optimization problem with respect to $\boldsymbol{\omega} = \{\omega_l\}_{l=1}^{M} \in \mathbb{R}^M$:

$$\min_{\boldsymbol{\omega}|\boldsymbol{\omega}\ge\mathbf{0},\mathbf{1}^\top\boldsymbol{\omega}=1} \sum_{i\in\mathbf{G}} -\log\left\{\sum_{j|(i,j)\in\mathbf{P}}\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)\right\} - \log\left\{1-\sum_{j|(i,j)\in\mathbf{N}}\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)\right\}, \tag{21}$$

$$\Rightarrow \min_{\boldsymbol{\omega}|\boldsymbol{\omega}\ge\mathbf{0},\mathbf{1}^\top\boldsymbol{\omega}=1} \left[J(\boldsymbol{\omega}) \triangleq \sum_{i\in\mathbf{G}} -\log\{\boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{P}}\} - \log\{1-\boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{N}}\}\right], \tag{22}$$

where

$$\boldsymbol{p}_i^{\mathbf{P}} = \left[\sum_{j|(i,j)\in\mathbf{P}} P_{ij}^{[1]}, \ldots, \sum_{j|(i,j)\in\mathbf{P}} P_{ij}^{[M]}\right]^\top \in \mathbb{R}^M,$$

$$\boldsymbol{p}_i^{\mathbf{N}} = \left[\sum_{j|(i,j)\in\mathbf{N}} P_{ij}^{[1]}, \ldots, \sum_{j|(i,j)\in\mathbf{N}} P_{ij}^{[M]}\right]^\top \in \mathbb{R}^M,$$

and we use the probabilistic constraint, $\sum_{j|(i,j)\in\mathbf{N}}\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i) = \boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{N}} \le 1$. Note that the union set $\mathbf{P}\cup\mathbf{N}$ does not cover the whole sample since there exist unlabeled samples, resulting in $\sum_{j|(i,j)\in\mathbf{P}}\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i) + \sum_{j|(i,j)\in\mathbf{N}}\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i) \le 1$, $\forall i$, and thus the two terms in (22) imply different types of cost. The derivative and Hessian of J are given by

$$\nabla J = \sum_{i\in\mathbf{G}} -\frac{\boldsymbol{p}_i^{\mathbf{P}}}{\boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{P}}} + \frac{\boldsymbol{p}_i^{\mathbf{N}}}{1-\boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{N}}}, \quad \nabla^2 J = \sum_{i\in\mathbf{G}} \frac{\boldsymbol{p}_i^{\mathbf{P}}\boldsymbol{p}_i^{\mathbf{P}\top}}{(\boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{P}})^2} + \frac{\boldsymbol{p}_i^{\mathbf{N}}\boldsymbol{p}_i^{\mathbf{N}\top}}{(1-\boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{N}})^2} \succcurlyeq 0,$$

which shows that (22) is convex with the unique global optimum. We apply the reduced gradient descent method [25] to minimize J under the probabilistic constraint, $\boldsymbol{\omega}\ge\mathbf{0}$, $\mathbf{1}^\top\boldsymbol{\omega}=1$, see Algorithm 2. The transition probabilities are finally unified into $\overline{\boldsymbol{P}} = \sum_l \omega_l \boldsymbol{P}^{[l]}$ (multiple KTP: MKTP) via the optimized $\boldsymbol{\omega}$, and the novel KTPS is subsequently obtained by (18) as $\overline{\boldsymbol{S}} = \overline{\boldsymbol{P}}\circ\overline{\boldsymbol{P}}^\top$ (MKTPS). In practice, we use $\log(\cdot+\varepsilon)$, say $\varepsilon = 1e^{-4}$, instead of $\log(\cdot)$ in (22) to avoid numerical instability.

The above proposed method is advantageous in terms of computation cost as compared to the standard MKL such as [25]. The size of training samples in (22) is $O(|\mathbf{G}|M)$ which is independent of the number of classes. The class information is reduced into only the two categories $\mathbf{P},\mathbf{N}$ indicating the coincidence of class labels in pairwise samples. Then, for each labeled sample, such pairwise information is merged into $\boldsymbol{p}_i^{\mathbf{P}}, \boldsymbol{p}_i^{\mathbf{N}}$, which suppresses the combinatorial increase of training sample vectors. Therefore, the computation cost for (22) depends only on the number of kernel functions M and that of labeled samples $|\mathbf{G}|$ even on multi-class problems. In addition, the proposed method does not contain any parameters to be set by users, such as a cost parameter in SVM-based MKL [25].

Algorithm 2. Multiple kernel-based transition probability similarity.

Input: $\{\boldsymbol{P}^{[l]} \in \mathbb{R}^{n\times n}\}_{l=1,\ldots,M}$: KTP matrices computed by using respective kernel functions $\mathrm{k}^{[l]}$
$\mathbf{G}$: the labeled sample indices
$\mathbf{P} = \{(i,j)|c_i = c_j, i,j\in\mathbf{G}\}$, $\mathbf{N} = \{(i,j)|c_i \ne c_j, i,j\in\mathbf{G}\}$

1: **Initialize** $\boldsymbol{\omega} = \frac{1}{M}\mathbf{1} \in \mathbb{R}^M$

2: $\boldsymbol{p}_i^{\mathbf{P}} = [\sum_{j|(i,j)\in\mathbf{P}} P_{ij}^{[1]}, \ldots, \sum_{j|(i,j)\in\mathbf{P}} P_{ij}^{[M]}]^\top \in \mathbb{R}^M$
$\boldsymbol{p}_i^{\mathbf{N}} = [\sum_{j|(i,j)\in\mathbf{N}} P_{ij}^{[1]}, \ldots, \sum_{j|(i,j)\in\mathbf{N}} P_{ij}^{[M]}]^\top \in \mathbb{R}^M$

3: **repeat**

4: $\boldsymbol{g} = \nabla J = \sum_{i\in\mathbf{G}} -\frac{\boldsymbol{p}_i^{\mathbf{P}}}{\boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{P}}} + \frac{\boldsymbol{p}_i^{\mathbf{N}}}{1-\boldsymbol{\omega}^\top\boldsymbol{p}_i^{\mathbf{N}}}$

5: $l^* = \arg\max_l \omega_l$

6: $\tilde{\boldsymbol{g}} : \tilde{g}_l = \begin{cases} g_l - g_{l^*} & (\omega_l > 0 \vee g_l - g_{l^*} < 0) \wedge l \ne l^* \\ -\sum_{l|\omega_l>0 \vee g_l - g_{l^*}<0} g_l - g_{l^*} & l = l^* \\ 0 & otherwise \end{cases}$

7: $\eta^* = \mathrm{argmin}_{\eta|\boldsymbol{\omega}-\eta\tilde{\boldsymbol{g}}\ge 0} J(\boldsymbol{\omega}-\eta\tilde{\boldsymbol{g}})$

8: $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \eta^*\tilde{\boldsymbol{g}}$

9: **until** convergence

Output: $\boldsymbol{\omega} \in \mathbb{R}^M$, $\overline{\boldsymbol{P}} = \sum_l \omega_l^M \boldsymbol{P}^{[l]}$

5.2. Comparison to unsupervised learning

In the previous section, we derive the method for integrating the multiple kernels in terms of KTP based on the discriminative,

Table 1
Classification approaches using similarity measure. Here, the similarity matrix is obtained via (22) $\boldsymbol{S}=(\sum_l w_l \boldsymbol{P}^{[l]})\circ(\sum_l w_l \boldsymbol{P}^{[l]})^\top$. The last row indicates the supervised method for comparison. The columns of 'Similarity' and 'Kernel' indicate the similarity matrix and the kernel Gram matrix input into the methods.

	Method	*Similarity*		*Kernel*		Learning
		$\{\boldsymbol{S}^{[l]}\}_l$	$\overline{\boldsymbol{S}}$	$\{\boldsymbol{K}^{[l]}\}_l$	$\overline{\boldsymbol{K}}$	
i	[23]	✓				*Semi-supervised*
ii	LP		✓			*Semi-supervised*
iii	Lap-SVM		✓		✓	*Semi-supervised*
iv	Lap-MKL		✓	✓		*Semi-supervised*
v	SVM				✓	*Supervised*
vi	MKL [25]			✓		*Supervised*

i.e., *supervised*, learning. It might seem to be slightly inconsistent with the KTP learning which is performed in the *unsupervised* framework (15). In what follows, we argue that the discriminative learning is required to the integration of multiple KTPs.

It is conceivable to incorporate the multiple kernel integration into the KTP learning (15) such as by

$$\min_{\boldsymbol{\omega}_{\boldsymbol{x}}\in\mathbb{R}^M,\{\boldsymbol{\alpha}^{[l]}\}_{l=1,\dots,M}} \sum_l^M \|\omega_{\boldsymbol{x}l}\boldsymbol{\Phi}_{\boldsymbol{X}}^{[l]}\boldsymbol{\alpha}^{[l]}-\omega_{\boldsymbol{x}l}\boldsymbol{\phi}_{\boldsymbol{x}}^{[l]}\|^2,$$
$$\text{s.t.}\quad \boldsymbol{\omega}_{\boldsymbol{x}}\geq 0,\ \mathbf{1}^\top\boldsymbol{\omega}_{\boldsymbol{x}}=1,\quad \forall l,\quad \frac{1}{\nu}\geq\boldsymbol{\alpha}^{[l]}\geq 0,\ \mathbf{1}^\top\boldsymbol{\alpha}^{[l]}=1, \tag{23}$$

and the KTP is retrieved by $\boldsymbol{\alpha}=\sum_l\omega_{\boldsymbol{x}l}\boldsymbol{\alpha}^{[l]}$. Note that the weight $\boldsymbol{\omega}_{\boldsymbol{x}}$ is defined at each sample. Recalling that (15) implies separability between the sample $\boldsymbol{x}$ and the others, the above formulation favors the kernel that embeds the samples onto a "smooth" distribution; that is, the high weight $\omega_{\boldsymbol{x}l}$ is assigned with such "smooth" kernel. Actually, the optimal weights are obtained as

$$\omega_{\boldsymbol{x}l}=\frac{\|\boldsymbol{\Phi}_{\boldsymbol{X}}^{[l]}\boldsymbol{\alpha}^{[l]}-\boldsymbol{\phi}_{\boldsymbol{x}}^{[l]}\|^{-2}}{\sum_{l'}\|\boldsymbol{\Phi}_{\boldsymbol{X}}^{[l']}\boldsymbol{\alpha}^{[l']}-\boldsymbol{\phi}_{\boldsymbol{x}}^{[l']}\|^{-2}}. \tag{24}$$

For instance, in the case of multiple Gaussian kernels $\mathrm{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp(-(1/h)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2)$ with different bandwidths h, the unsupervised integration (23) produces the weights in which the Gaussian of the larger bandwidth is highly weighted, no matter how the label distribution is, degrading discriminative power. Thus, we insist on the discriminative learning (22) for integrating the multiple kernels.

5.3. Multiple kernel based classification

The proposed method to integrate multiple kernels via KTP (Section 5.1) produces the integration weight $\boldsymbol{\omega}$ which is subsequently utilized to induce the composite similarity measure (MKTPS $\overline{\boldsymbol{S}}$). The weights are optimized by maximizing the discriminativity of the KTPs, each of which is derived from each kernel function. The KTP characterizes the kernel function, exploiting the inherent manifold structure of sample distribution in the kernel space. Thus, the optimum weight must somehow reflect the contributions of the respective kernel functions for discrimination, which leads to combine the multiple kernels via the weights; $\overline{\boldsymbol{K}}=\sum_l^M\omega_l\boldsymbol{K}^{[l]}$, as the MKL [19] does. It should be noted that the weight is not optimized directly in terms of specific classification but from the viewpoint of discriminative transition probabilities. Due to the generality, we can expect that the weight also conveys favorable discriminative power to the kernel functions used for the classifier, as well as the similarity measure.

As a result, given multiple types of kernel functions, we obtain four kinds of feature pool available for classification: $\{\boldsymbol{S}^{[l]}=\boldsymbol{P}^{[l]}\circ\boldsymbol{P}^{[l]\top}\}_l$, $\overline{\boldsymbol{S}}$, $\{\boldsymbol{K}^{[l]}\}_l$, $\overline{\boldsymbol{K}}$. By exploiting these, we propose multiple kernel based classification methods in semi-supervised/supervised learning, which are summarized in Table 1.

i. *Only multiple similarities* $\{\boldsymbol{S}^{[l]}\}_l$ (*semi-supervised*): Tsuda et al. [23] proposed the method for combining the multiple similarity measures in the framework of the label propagation [7]. The method is related to our proposed method (Section 5.1) in that the multiple types of similarities are combined, though taking a different way. The proposed method works on the KTP by $\overline{\boldsymbol{P}}=\sum_l\omega_l\boldsymbol{P}^{[l]}$ to produce $\overline{\boldsymbol{S}}=(\sum_l\omega_l\boldsymbol{P}^{[l]})\circ(\sum_l\omega_l\boldsymbol{P}^{[l]})^\top$, while the method [23] directly combines the similarity measure as $\boldsymbol{S}=\sum_l\omega_l(\boldsymbol{P}^{[l]}\circ\boldsymbol{P}^{[l]\top})$. In the proposed method, we can further exploit the relationships between different types of KTP via $\omega_l\omega_{l'}\boldsymbol{P}^{[l]}\circ\boldsymbol{P}^{[l']\top}$, $l\neq l'$.

ii. *Only MKTPS* $\overline{\boldsymbol{S}}$ (*semi-supervised*): The MKTPS is simply applied to label propagation for (transductive) classification.

iii. *MKTPS* $\overline{\boldsymbol{S}}$ *and composite kernel* $\overline{\boldsymbol{K}}$ (*semi-supervised*): In the following, we provide the kernel-based classification methods which are applicable to transductive/inductive classification. Both the composite similarity $\overline{\boldsymbol{S}}$ (MKTPS) and the composite kernel $\overline{\boldsymbol{K}}$ mentioned above are fed into the kernel-based semi-supervised methods, such as Laplacian SVM (Lap-SVM) [13] and semi-supervised discriminant analysis [11]. Those methods are based on a *single* kernel function and a *single* similarity measure for training the classifier.

iv. *MKTPS* $\overline{\boldsymbol{S}}$ *and multiple kernels* $\{\boldsymbol{K}^{[l]}\}_l$ (*semi-supervised*): Here, we substitute the composite kernel $\overline{\boldsymbol{K}}$ in the above method with multiple kernels $\{\boldsymbol{K}^{[l]}\}_l$, which raises semi-supervised MKL using MKTPS. The semi-supervised MKL has been mentioned in [24], though it requires prior knowledge regarding class categories and is specific to the exponential family parametric model. In this paper, we extend the method of simpleMKL [25] to formulate the semi-supervised method based on the graph Laplacian in a general form; this method is a counterpart of the Lap-SVM [13], as is the case that simpleMKL is regarded as multiple-kernel version of SVM in the supervised learning.

We newly introduce the kernel weight $\boldsymbol{d}=\{d_l\}_{l=1,\dots,M}$ to integrate multiple kernel functions into the classifier, and thereby define the following formulation (see Appendix D):

$$\min_{\boldsymbol{d},\{\boldsymbol{v}^{[l]}\}_l,\{\xi_i\}_i} \frac{1}{2}\sum_l^M\frac{1}{d_l}\|\boldsymbol{v}^{[l]}\|^2+\frac{\theta}{4}\sum_l^M\frac{1}{d_l}\sum_{ij}^{n+N}\overline{S}_{ij}\|\boldsymbol{v}^{[l]\top}(\boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]}-\boldsymbol{\phi}_{\boldsymbol{x}_j}^{[l]})\|^2+C\sum_i^n\xi_i$$
$$\text{s.t.}\quad \forall i\in\{1,\dots,n\},\ y_i\left(\sum_l^M\boldsymbol{v}^{[l]\top}\boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]}+b\right)$$
$$\geq 1-\xi_i,\ \xi_i\geq 0,\ \forall l,\ d_l\geq 0,\sum_l^M d_l=1, \tag{25}$$

where the n samples of $i=1,\dots,n$ are labeled while the rest N samples of $i=n+1,\dots,n+N$ are unlabeled, $\overline{S}_{ij}$ indicates the MKTPS between the i, j-th samples and θ is a regularization parameter. The second term in the objective cost is introduced for dealing with the unlabeled samples via the graph Laplacian (1) in a manner similar to Lap-SVM. Note that the formulation (25) is convex by virtue of the convexity in the regularization (1). This proposed method contains two types of integration weights for similarity and kernels which are separately optimized; after obtaining the integrated similarity (Section 5.1), we apply the Laplacian simpleMKL, which is referred to as Lap-MKL.

v. *Only composite kernel* $\overline{\boldsymbol{K}}$ (*supervised*): In the last two methods, we mention the classification in the supervised learning, for comparison. The composite kernel $\overline{\boldsymbol{K}}$ is simply applied to any kinds of kernel-based classification methods, such as SVM.

vi. *Only multiple kernels* $\{\boldsymbol{K}^{[l]}\}_l$ (*supervised*): Multiple types of kernels are usually treated in the supervised framework, such as simpleMKL [25].

In summary, while the last two methods are supervised, the first four methods are semi-supervised methods; especially, the

a

Method	Acc. (%)
proposed $\boldsymbol{S}=\boldsymbol{P}\circ\boldsymbol{P}^{\top}$	95.3 ±0.6
mean $\boldsymbol{S}=\frac{1}{2}(\boldsymbol{P}+\boldsymbol{P}^{\top})$	89.1 ±1.6

b

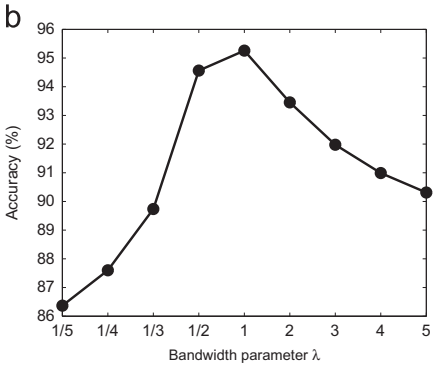


Fig. 5. Comparison with respect to (18) for constructing KTPS. The performance is reported at 2% labeled samples. (a) Comparison to the arithmetic mean and (b) parameter λ.

methods of [23] and LP are intended for transductive classification, which could have difficulty in predicting unseen samples. In terms of the weight $\boldsymbol{\omega}$ learnt by the proposed method (Section 5.1), we highly contribute to the method of iii. Lap-SVM that utilizes the weight both for similarity and kernel, as well as iv. Lap-MKL and v. SVM that use it for similarity and kernel, respectively. Thus, it should be noted that those methods (Table 1 iii–v) are novel in this work by integrating the proposed method (Section 5.1) into the classification frameworks that have been proposed in [13,26] and the proposed framework (Table 1 iv).

6. Experimental results

We conducted experiments in the framework of semi-supervised learning using similarities to validate the proposed methods (Sections 4 and 5). The experiments are categorized into two parts: the first part in Sections 6.1 and 6.2 deals with the similarity derived from a single kernel, while the second part in Sections 6.3 and 6.4 focuses on the integration of multiple kernels.

6.1. Similarity learning from a single kernel

We apply the similarity to label propagation [7] for estimating the labels based on a few labeled samples. This transductive classification using LP enables us to simply evaluate the performance of the similarity measure itself.

USPS dataset [31] is used for this experiment, containing 7291 hand-written digits (0–9) images (16×16 pixels) to form a 10-class problem. The image vector whose dimensionality $D=256$ is simply employed as the image feature. We used the Gaussian kernel $\exp(-(1/h)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2)$ of which bandwidth parameter h is determined as the mean of the pairwise distances denoted by γ. The labeled samples are randomly drawn from the whole dataset and the classification accuracy is measured on the remaining unlabeled samples; the ratio of the labeled samples ranges from 1% to 10% per category. The trial is repeated 10 times and the average performance is reported.

We investigate the settings in the proposed KTPS described in Section 4, regarding both the formulation (18) itself and λ in (18). Fig. 5 shows the performance results at 2% labeled samples. The formulation (18) to compute similarity (KTPS) from the transition probabilities (KTP) is a sort of geometric mean, for which the *arithmetic mean* is conceivable as an alternative; $\boldsymbol{S}=(\boldsymbol{P}+\boldsymbol{P}^{\top})/2$. The comparison of those two approaches is shown in Fig. 5a, demonstrating that the proposed method (18) significantly outperforms the arithmetic mean. As described in Section 4, (18) is derived based on the probabilistic metric of the transition probability KTP, and besides it renders sparser similarities compared to the arithmetic mean. Then, various parameter values of λ controlling the bandwidth in (18) are tested as shown in Fig. 5b. The best performance is achieved at the case of $\lambda=1$ simply retaining the form of the transition probabilities $\mathrm{p}(\boldsymbol{x}_j|\boldsymbol{x}_i)$ in KTPS. Thus, we set $\lambda=1$ in KTPS.

Table 2
Similarities.

KTPS	Proposed similarity (Section 4)
LNS	Linear neighborhood similarity [9]
KCS	Kernel cone-based similarity [22]
LSIS	(Linear) sparsity induced similarity [8]
KSIS	Kernel extension of sparsity induced similarity [8]
KS	(Gaussian) kernel-based similarity with k-NN
KS-tuned	(Gaussian) kernel-based similarity with tuned k

Next, the proposed KTPS is compared to the other types of similarity listed in Table 2; linear neighborhood similarity (LNS) [9], kernel cone-based similarity (KCS) [22], sparsity induced similarity (SIS) [8], and (Gaussian) kernel-based similarity (KS). For LNS, we utilized the coefficients obtained in linear neighborhood propagation [9] for similarities as in [8]. In [8], SIS is proposed in linear (original) input space (LSIS), and in this paper we also develop it to the kernel-based similarity (KSIS) via kernel tricks. For KS, we directly utilize the (Gaussian) kernel values as similarities, $S_{ij}=\mathrm{k}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp(-(1/h)\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2)$, corresponding to Gaussian kernel similarity (GKS) in this experiment. For computational efficiency, all the methods compute the similarities on k nearest neighbors with somewhat larger k. In KS, however, since the number of neighbors k has to be carefully tuned for better performance, we additionally apply improved KS with tuned k so as to produce favorable performances, which is denoted by KS-tuned. The kernel-based methods, KTPS, KCS, KSIS and KS, use the identical Gaussian kernel for fair comparison. We implemented these methods on 3.33 GHz PC by using MATLAB with MOSEK toolbox for LNS, FNNLS [32] for KCS and L_1-magic toolbox[5] for LSIS/KSIS.[6] The number of neighbors k is set by $k=1.5\times256=384$, as in [8], and for KS-tuned, $k=100$.

The performance results are shown in Fig. 6. The proposed KTPS significantly outperforms the others; in particular, the performance is over 90% even when only 1% samples are labeled. To account for the high performance in KTPS, we analyze the characteristics of the similarity in detail, focusing on the precision in (non-zero) similarities. The non-zero similarity $S_{ij}(>0)$ is

[5] L1-Magic http://www.acm.caltech.edu/l1magic/.

[6] For SIS [8], we apply PCA to k-NN samples so as to satisfy the linear dependency defined in equality constraint, since such dependency rarely holds in our experiments. The violation of the linear dependency in the intrinsic sample distribution seem to degenerate the performance of SIS.

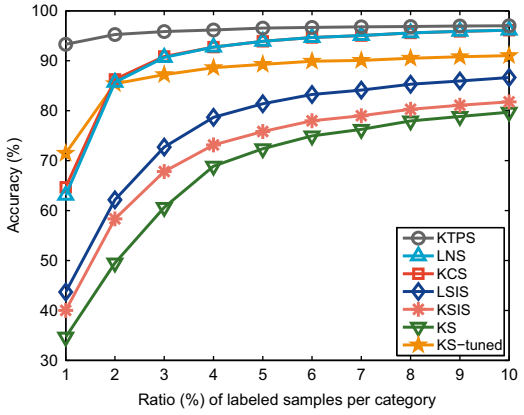


Fig. 6. Classification accuracy on USPS.

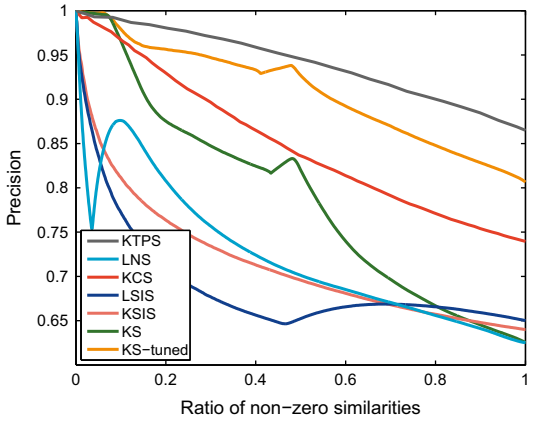


Fig. 7. Precision on similarities.

assigned with the binary label y_{ij} where $y_{ij}=+1$ for $c_i=c_j$ and $y_{ij}=-1$ for $c_i\neq c_j$. Based on the labels with certain threshold $\tau\geq 0$, the precision in the similarity is computed by

$$prec(\tau)=\frac{|\{(i,j)|i>j,S_{ij}>\tau,y_{ij}=+1\}|}{|\{(i,j)|i>j,S_{ij}>\tau\}|}, \tag{26}$$

together with the ratio of the samples

$$ratio(\tau)=\frac{|\{(i,j)|i>j,S_{ij}>\tau\}|}{|\{(i,j)|i>j,S_{ij}>0\}|}. \tag{27}$$

The couples of $\{prec(\tau),ratio(\tau)\}$ are shown in Fig. 7 for various τ. This result shows that KTPS is quite *clean* predominantly containing *correct* similarities, *e.g.*, $prec(0)>0.85$, with a small amount of wrong ones, compared to the other types of similarity. Due to the *clean* similarity, the label information is precisely propagated via LP, producing the superior performance.

There remains the practical issue regarding the robustness to the parameter settings for the bandwidth h in the Gaussian kernel and the number of neighbors k. We evaluated the performances for $h\in\{0.1\gamma,0.5\gamma,\gamma,5\gamma,10\gamma\}$ and $k\in\{0.5D,D,1.5D,2D,4D\}$ on 2% labeled samples. For comparison, the method of RMGT [10] is also applied as a reference, though it constructs similarities in a discriminative manner using labeled samples in contrast to the other methods which produce similarities in an unsupervised manner. The results are shown in Table 3. The performances of KTPS are stably high and robust, whereas those of the other similarities significantly fluctuate at lower performance accuracies. This result shows that the proposed KTPS is robust to such parameter settings, which is important to free us from exhaustively

Table 3
Average accuracy (%) and its standard deviation for virous parameter values on USPS with 2% labeled samples.

KTPS	LNS	KCS	LSIS	KSIS	KS	RMGT [10]
95.4 ± 0.5	86.1 ± 2.6	87.0 ± 2.4	62.8 ± 14.7	60.7 ± 19.3	50.4 ± 19.8	91.9 ± 6.0

Table 4
Computation time (ms) per sample for constructing similarities on USPS.

KTPS SMO	KTPS MOSEK	LNS	KCS	LSIS	KSIS	RMGT [10]
2.0	113.7	124.4	3.2	177.2	185.5	107.4

tuning the parameters, as discussed in Section 3.2; especially, it is enough to set somewhat larger k without tuning unlike GKS.

We also measured the average computation time required only for calculating the similarity per sample except for kernel computation and k-NN search which are common across the methods. The results are shown in Table 4, omitting the result of KS which requires only kernel computation and k-NN, while appending the result of KTPS using standard QP solver (`mskqpopt` in MOSEK) as a reference. For RMGT, we report the averaged computation time on the above experimental setting (Table 3). The computation time of KTPS is significantly short compared to the others, demonstrating that the SMO approach (Section 3.2) is effective in practice.

As shown in the above experimental results, we can say that the proposed KTPS works in the label propagation quite effectively in terms of the classification performance as well as the computational cost, showing also the robustness to the parameter settings.

6.2. *Similarity-based semi-supervised classification using single kernel*

Next, we applied the similarity to semi-supervised classification methods; semi-supervised discriminant analysis (SDA) [11] and Laplacian support vector machine (Lap-SVM) [13] which produce the classifiers directly working on the feature vector to perform both transductive and inductive classifications. The methods of SDA and Lap-SVM are developed by extending (supervised) Fisher discriminant analysis and SVM so as to incorporate the unlabeled samples via the graph Laplacian regularization [6], respectively. The SDA provides the projection vectors into the discriminant space, and in this experiment, the samples are classified by the 1-NN method in the discriminant space. On the other hand, the Lap-SVM optimizes the classifier in the framework of maximum margin [33], which is formulated as a linear classifier in this experiment. We employ one-vs-rest approach to cope with multi-classes.

We used ORL[7] and UMIST face dataset [34]:

ORL face dataset is composed of ten face images for each of the 40 human subjects. While the size of original images is 92×112, we resized the images to 32×32 for efficiency. The image vector ($\in\mathbb{R}^{1024}$) is simply employed as the feature vector, and the Gaussian kernel is applied in the same manner as in USPS dataset. We drew seven training samples per category for learning the classification methods and the remaining samples are used for test set. All samples are used as neighbors ($k=279$), while $k=5$ for KS-tuned.

[7] http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip.

Table 5
Classification accuracy (%) by SDA.

Method	(a) ORL face dataset		(b) UMIST face dataset	
	Transductive	Inductive	Transductive	Inductive
DA	67.6 ± 3.0	67.5 ± 4.1	44.1 ± 3.6	46.3 ± 3.6
Wang et al. [35]	72.1 ± 1.9	71.3 ± 2.2	63.1 ± 1.9	62.6 ± 1.8
KS-tuned + SDA	74.5 ± 3.1	70.6 ± 3.9	53.1 ± 4.6	54.1 ± 4.1
KS + SDA	50.5 ± 2.8	54.9 ± 5.0	34.0 ± 3.7	36.7 ± 4.3
KSIS + SDA	63.0 ± 3.1	64.5 ± 4.9	41.3 ± 3.4	43.3 ± 3.9
LSIS + SDA	60.2 ± 3.5	62.0 ± 4.8	39.3 ± 4.0	41.6 ± 4.2
KCS + SDA	76.2 ± 3.0	65.0 ± 4.1	50.7 ± 3.1	48.6 ± 3.2
LNS + SDA	76.1 ± 3.2	65.8 ± 4.3	51.9 ± 3.3	50.2 ± 3.3
KTPS + SDA	**83.7 ± 2.9**	**77.9 ± 3.9**	**71.6 ± 4.3**	**71.9 ± 4.5**

Table 6
Classification accuracy (%) by Lap-SVM.

Method	(a) ORL face dataset		(b) UMIST face dataset	
	Transductive	Inductive	Transductive	Inductive
SVM	72.1 ± 2.6	72.6 ± 4.3	46.2 ± 3.3	49.2 ± 3.3
KS-tuned + Lap-SVM	75.9 ± 2.6	74.8 ± 3.6	54.9 ± 4.3	56.6 ± 3.8
KS + Lap-SVM	72.3 ± 2.6	72.5 ± 4.2	46.8 ± 2.8	49.8 ± 3.5
KSIS + Lap-SVM	72.3 ± 2.7	72.5 ± 4.3	46.5 ± 2.9	49.7 ± 3.5
LSIS + Lap-SVM	72.2 ± 2.7	72.5 ± 4.3	46.5 ± 2.9	49.7 ± 3.5
KCS + Lap-SVM	79.5 ± 2.1	73.4 ± 4.1	54.3 ± 3.2	52.9 ± 3.0
LNS + Lap-SVM	79.9 ± 2.2	73.1 ± 4.3	55.6 ± 3.2	53.7 ± 3.1
KTPS + Lap-SVM	**86.3 ± 2.4**	**82.1 ± 3.3**	**73.1 ± 3.9**	**74.1 ± 3.9**

UMIST face dataset [34] consists of 575 images from 20 persons. While the original pre-cropped images are of size 92×112, we resized the images to 32×32 as in ORL dataset. The image vector ($\in \mathbb{R}^{1024}$) is simply employed as the feature vector, and the Gaussian kernel is applied. We drew 15 training samples per category and the remaining samples are used for test set. All samples are used as neighbors ($k = 299$), while $k = 5$ for KS-tuned.

In each dataset, only one sample per category is labeled in the training set, while the others in the training set are regarded as *unlabeled* samples in the semi-supervised methods. We run on 50 random splits and report the averaged performance which is evaluated in two ways; the classification accuracy on the training unlabeled set (transductive accuracy) and on the test set (inductive accuracy) [35].

The performance results are shown in Tables 5 and 6 in which the performance by Wang et al. [35] measured in the same protocol is also presented as a reference. Note that the methods of DA and SVM are applied in the supervised setting using only one labeled sample per category. Both of SDA and Lap-SVM using the proposed KTPS outperform the other methods including [35] and even supervised methods (DA and SVM) in terms of transductive and inductive accuracies. These results demonstrate that the KTPS is well-suited to those semi-supervised classification methods. The KTPS can favorably boost the performances of the semi-supervised methods via the graph Laplacian regularization; especially, the KTPS+Lap-SVM is superior to KTPS+DA, exhibiting better generalization performance of the (linear) classifier by Lap-SVM compared to that of the 1-NN in SDA space.

We then show in Figs. 8 and 9 the effectiveness of the upper-bound constraints by $1/\nu$ in (15). In these experimental settings of weak labeling (only one sample per category is labeled), the issue of too sparse similarity discussed in Section 3.3 would be crucial. The performance is improved by imposing the upper bound with $\nu = 2$, while the larger ν degrades it by producing the tighter bound and thus unfavorably deteriorating the sparseness in the similarities. We thus employ $\nu = 2$ in this paper which stably produces favorable performance in all the experiments.

6.3. Similarity learning from multiple kernels

In this section, utilizing multiple types of kernel functions, we applied the method of multiple-kernel integration for MKTPS described in Section 5.1. In order to simply evaluate the composite similarity measure as in Section 6.1, the transductive classification by LP with MKTPS (Table 1 ii) is performed on the following two datasets:

Bird dataset [36] contains six bird classes with 100 images per class. All samples are used as neighbors ($k = 599$), while $k = 10$ for KS-tuned.

Butterfly dataset [37] has 619 images of seven butterfly classes. All samples are used as neighbors ($k = 618$), while $k = 10$ for KS-tuned.

For these datasets, we employed three types of precomputed pairwise distances provided in the website[8] of the authors [38]; for details of the distances refer to [38]. The multiple (three) types of kernels are accordingly constructed by applying Gaussian kernel to those precomputed distances in the same manner as in USPS dataset. We drew labeled samples ranging from 10% to 50%, and the remaining unlabeled samples are classified by LP [7]. The classification accuracies averaged over 10 trials are reported.

First of all, the proposed MKTPS is compared to the best single similarity that produces the highest performance among the three types of kernels. Note that the similarities of KTPS, KCS, KSIS and KS are constructed for each type of kernel, while MKTPS is obtained by integrating those multiple kernels. Fig. 10 shows the performance results. The multiple kernels are favorably combined in MKTPS, improving the performance compared to the other best single similarities, even to the single KTPS. The performance gain increases along the number of labeled samples since the discriminative learning (Section 5.1) becomes more effective for larger training samples.

Then, the proposed multiple kernel integration (22) is compared to the other alternative formulations/methods as follows.

a. *Cost function*: In (22), we measure the errors based on log likelihood from the probabilistic viewpoint. For minimizing the erroneous transition probabilities as described in Section 5.1, the other formulations are also induced from other types of error function:

$$L_2 \text{ error}: \min_{\boldsymbol{\omega} | \boldsymbol{\omega} \geq \mathbf{0}, \mathbf{1}^\top \boldsymbol{\omega} = 1} \sum_{i \in \mathbf{G}} \{1 - \boldsymbol{\omega}^\top \boldsymbol{p}_i^{\mathbf{P}}\}^2 + \{\boldsymbol{\omega}^\top \boldsymbol{p}_i^{\mathbf{N}}\}^2$$

$$\Leftrightarrow \min_{\boldsymbol{\omega} | \boldsymbol{\omega} \geq \mathbf{0}, \mathbf{1}^\top \boldsymbol{\omega} = 1} \boldsymbol{\omega}^\top \left\{ \sum_{i \in \mathbf{G}} \boldsymbol{p}_i^{\mathbf{P}} \boldsymbol{p}_i^{\mathbf{P}\top} + \boldsymbol{p}_i^{\mathbf{N}} \boldsymbol{p}_i^{\mathbf{N}\top} \right\} \boldsymbol{\omega} - \boldsymbol{\omega}^\top \sum_{i \in \mathbf{G}} \boldsymbol{p}_i^{\mathbf{P}}.$$

$$L_1 \text{ error}: \min_{\boldsymbol{\omega} | \boldsymbol{\omega} \geq \mathbf{0}, \mathbf{1}^\top \boldsymbol{\omega} = 1} \sum_{i \in \mathbf{G}} \{1 - \boldsymbol{\omega}^\top \boldsymbol{p}_i^{\mathbf{P}}\} + \{\boldsymbol{\omega}^\top \boldsymbol{p}_i^{\mathbf{N}}\}$$

$$\Leftrightarrow \min_{\boldsymbol{\omega} | \boldsymbol{\omega} \geq \mathbf{0}, \mathbf{1}^\top \boldsymbol{\omega} = 1} \boldsymbol{\omega}^\top \sum_{i \in \mathbf{G}} \{\boldsymbol{p}_i^{\mathbf{N}} - \boldsymbol{p}_i^{\mathbf{P}}\}.$$

Note that $0 \leq \boldsymbol{\omega}^\top \boldsymbol{p}_i^{\mathbf{P},\mathbf{N}} \leq 1$ from the probabilistic constraints regarding $\boldsymbol{\omega}$ and $\boldsymbol{p}$. The method of L_2 error results in quadratic programming, while L_1 error leads to linear programming which produces the most sparse weight, $\omega_{l \neq l^*} = 0, \omega_{l^*} = 1$ where $l^* = \arg\min_l \sum_{i \in \mathbf{G}} \{p_i^{\mathbf{N}\,[l]} - p_i^{\mathbf{P}\,[l]}\}$.

b. *Unsupervised integration*: The unsupervised integration method in Section 5.2 is applied to contrast the proposed method (22) defined in supervised (discriminative) learning. This method (23) produces the weight $\boldsymbol{\omega}_{\boldsymbol{x}}$ at each sample, while the method (22) outputs the single weight $\boldsymbol{\omega}$ across the whole sample.

[8] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/msorec/.

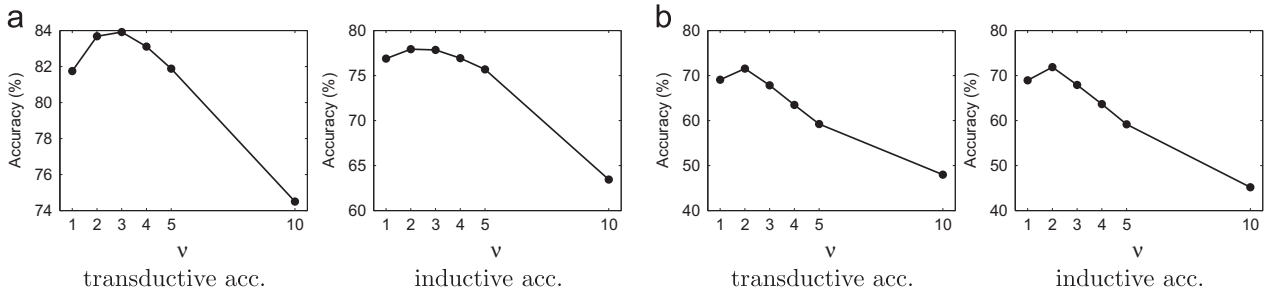


Fig. 8. Classification accuracy on various ν for (constrained) KTPS+SDA. (a) ORL and (b) UMIST.

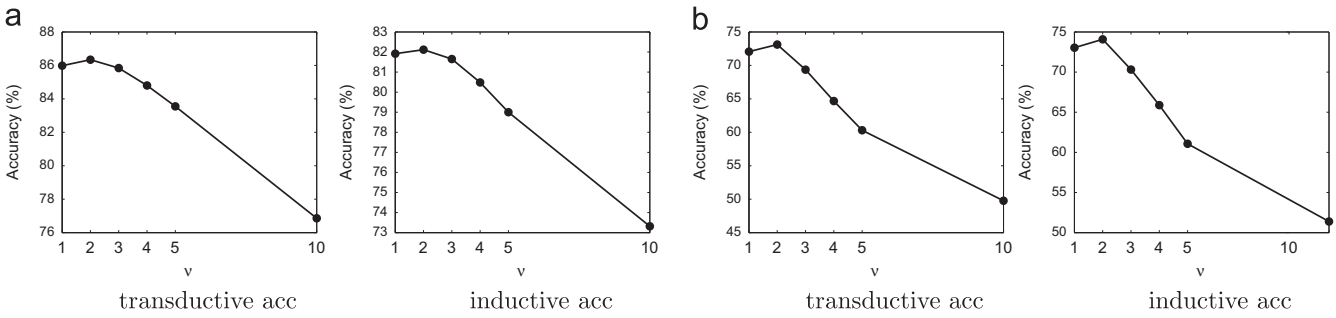


Fig. 9. Classification accuracy on various ν for (constrained) KTPS+Lap-SVM. (a) ORL and (b) UMIST.

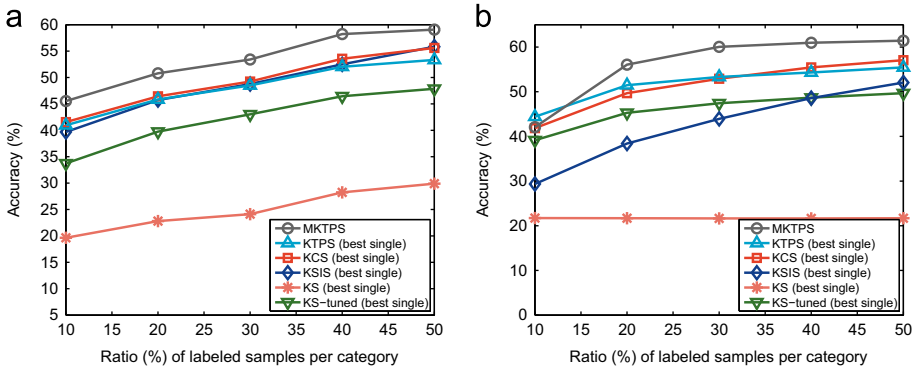


Fig. 10. Classification accuracy using similarities derived from multiple kernels. (a) Bird and (b) Butterfly.

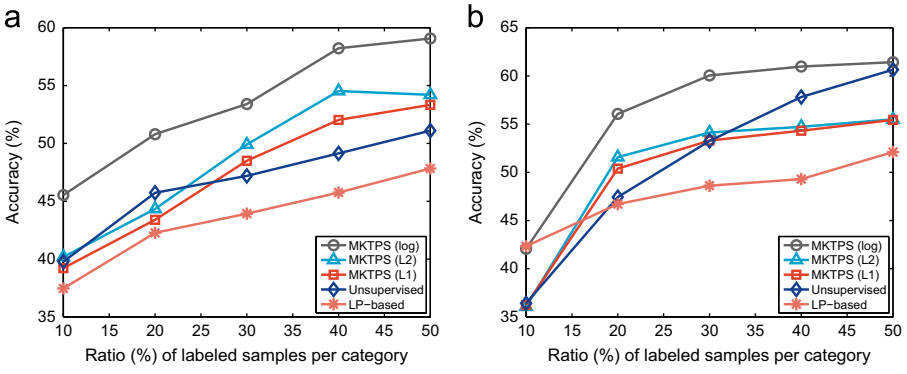


Fig. 11. Performance comparison on various formulations in MKTPS. (a) Bird and (b) Butterfly.

c. *LP-based integration* [23]: Tsuda et al. [23] extended the LP-based cost function to cope with the multiple types of similarity measure (Table 1 i). From the viewpoint of multiple kernel/similarity integration, the method resembles the proposed method (22), but the criterion in cost function to be minimized is completely different from ours.

The performance comparison is shown in Fig. 11. Fig. 12 depicts the examples of the kernel weight; the performance by individual KTPS is also shown in the leftmost column for comparison. As to a cost function, MKTPS-L_1 produces too sparse weights which have only one non-zero component, and the weights by MKTPS-L_2 are also rather sparse. The proposed MKTPS-log favorably takes into account all of the kernel functions with non-zero weights for constructing discriminative similarities, exhibiting superior performances. We can see that the weights are assigned by MKTPS-log in accordance with the performance of the individual KTPS by

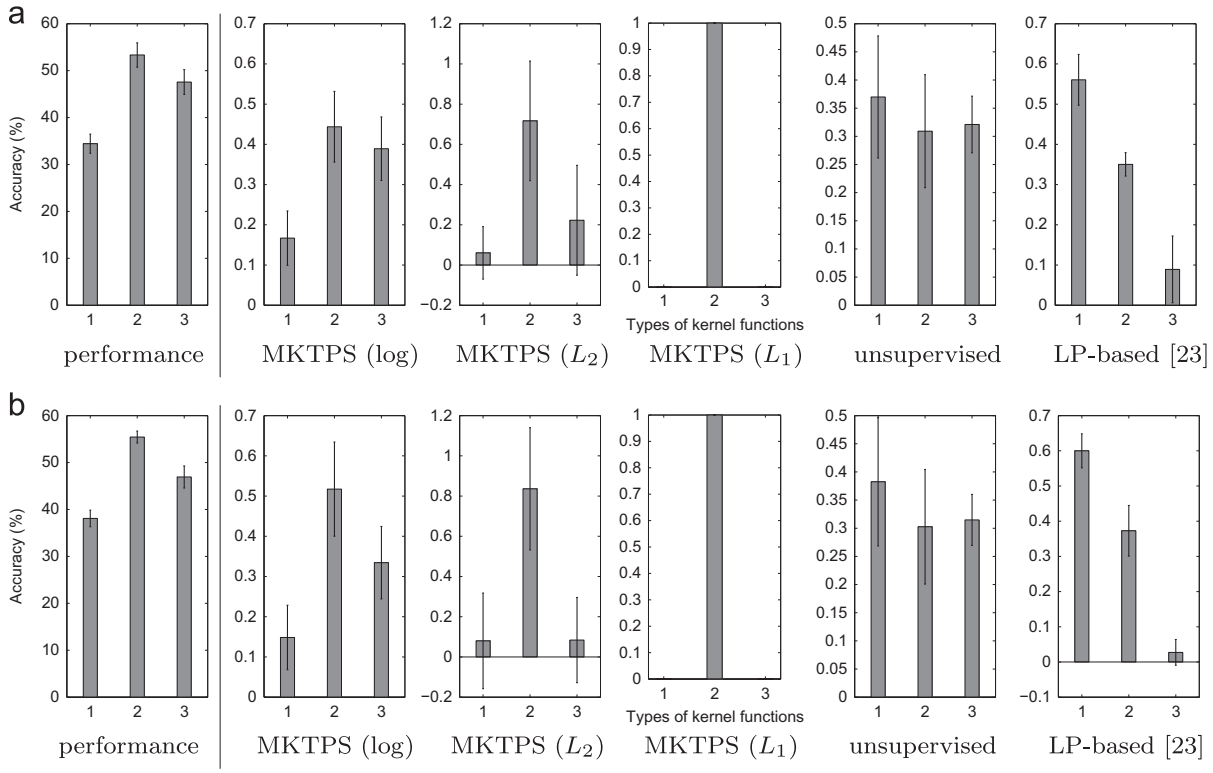


Fig. 12. Weights over multiple kernels/similarities on various methods. The left-most column shows the performance of the individual KTPS. (a) Bird and (b) Butterfly.

comparing the left two columns in Fig. 12. The unsupervised integration results in dense weights which do not reflect the discriminativity of the individual similarity (kernel). As described in Section 5.2, the unsupervised method determines the weights based on the smoothness of the kernel functions in disregard of their discriminative power. Though the LP-based method [23] produces rather sparse weights, it also suffers from the smoothness of the similarity via the graph Laplacian regularization, hardly taking into account the discriminativity. These experimental results exhibit the effectiveness of the proposed MKTPS in terms of the cost function and the discriminative learning.

6.4. Similarity-based semi-supervised classification using multiple kernels

At the last, we evaluate the semi-supervised/supervised classification methods using multiple types of kernel functions. Based on the results in Section 6.2 that Lap-SVM is superior to SDA, the maximum-margin classifiers listed as Table 1 iii–vi are applied. Note that in this experiment these methods optimize 'kernel'-based classifiers [28] unlike in Section 6.2. Those methods are again summarized and categorized in Table 7; we employ the method of simpleMKL [25] for vi. Supervised MKL, and the others are novel methods in this paper since the iii and v methods especially utilize the kernel weights produced via the proposed MKTP (Section 5.1) for integrating the multiple kernels and the vi method is newly proposed in Section 5.3. Those methods are tested on object classification using Caltech101 [39] and Caltech256 [40] datasets.

Caltech101 dataset [39] contains images in 102 object categories including 'background' category. We used ten types of precomputed kernels provided in the website[9] of the authors [46]; for details of the kernels refer to [46]. The number of neighbors is set to $k=500$. We randomly draw three types of (disjoint) set, the labeled, unlabeled training sets and the test set; the labeled set contains 2–15 samples per category, while the unlabeled and the test sets are composed of 15 samples per category. The semi-supervised methods of iii and iv use the labeled and unlabeled training sets for learning the classifier, while the supervised methods of v and vi are trained on the labeled set. The trial is repeated three times and the average classification accuracies are reported. The classification accuracies are measured over both the unlabeled set (*transductive*) and the test set (*inductive*).

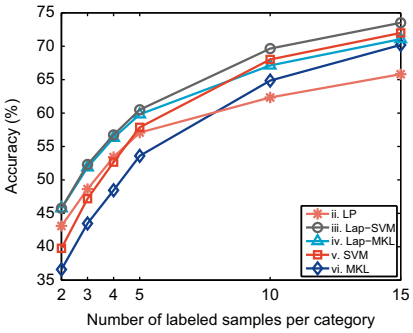
Table 7
Categorization of methods.

Learning	*Kernel weight*	
	MKTP	MKL
Semi-supervised	iii. Lap-SVM	iv. Lap-MKL
Supervised	v. SVM	vi. MKL [25]

Caltech256 dataset [40] is a more challenging dataset than Caltech101 since it consists of 256 object categories with large intra-class variations regarding such as object locations, sizes and poses in the images. We employed 39 types of kernels used in [42]; for details of the kernels refer to [42]. As in Caltech101, three types of set are randomly picked up; the labeled, unlabeled and test sets consist of 2–30, 30 and 30 samples per category, respectively. The averaged *transductive* and *inductive* classification accuracies are measured on three-time trials.

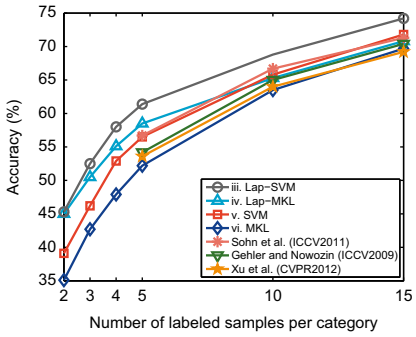
Figs. 13 and 14 show the performance results on Caltech101 dataset, and Figs. 15 and 16 are for Caltech256 dataset; for comparison, in the transductive classifications, the results by LP (Table 1 ii) are shown and the inductive results include the performance of the prior works. These kernel-based classifiers exhibit superior performance compared to LP, showing that the

[9] http://www.robots.ox.ac.uk/~vgg/software/MKL/ker-details.html.



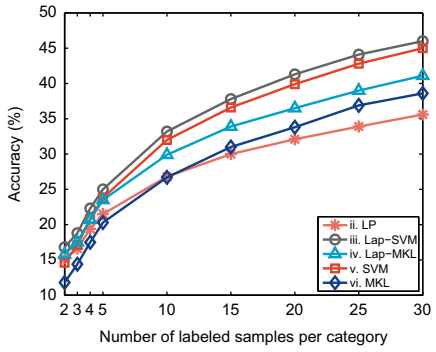
	2	3	4	5	10	15
ii. LP	43.1 ±1.7	48.6 ±0.0	53.4 ±1.6	57.1 ±0.5	62.3 ±0.7	65.8 ±1.3
iii. Lap-SVM	45.8 ±1.7	52.3 ±0.2	56.7 ±1.3	60.5 ±0.3	69.7 ±0.5	73.5 ±0.7
iv. Lap-MKL	45.9 ±1.1	51.9 ±0.8	56.6 ±1.3	59.9 ±0.4	68.3 ±0.5	71.9 ±0.3
v. SVM	39.8 ±1.4	47.2 ±0.2	52.7 ±1.4	57.9 ±1.1	68.0 ±1.0	72.0 ±0.8
vi. MKL	36.6 ±1.0	43.5 ±0.1	48.5 ±1.8	53.6 ±1.1	64.9 ±0.9	70.2 ±1.0

Fig. 13. Transductive classification accuracies on Caltech101 dataset.



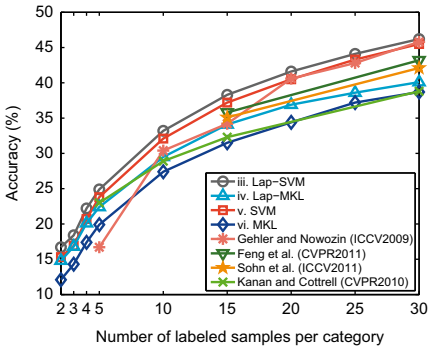
	2	3	4	5	10	15
iii. Lap-SVM	45.3 ±2.6	52.5 ±0.3	58.0 ±0.6	61.4 ±0.6	68.8 ±0.2	74.2 ±0.8
iv. Lap-MKL	45.0 ±1.8	50.5 ±0.5	55.1 ±0.6	58.5 ±0.9	65.3 ±0.8	70.8 ±0.1
v. SVM	39.1 ±2.4	46.2 ±1.6	52.9 ±0.8	56.5 ±0.7	65.8 ±0.5	71.8 ±0.5
vi. MKL	35.1 ±1.4	42.7 ±0.8	47.9 ±1.1	52.2 ±0.1	63.5 ±0.2	69.7 ±1.2
[41]				56.7	66.7	71.3
[42]				54.2	65.0	70.4
[43]				53.6	64.0	69.2

Fig. 14. Inductive classification accuracies on Caltech101 dataset [41–43].



	2	3	4	5	10	15	20	25	30
ii. LP	15.2 ±0.2	16.6 ±0.5	19.4 ±0.3	21.5 ±0.6	26.8 ±0.2	30.0 ±0.0	32.1 ±0.4	33.9 ±0.5	35.6 ±0.2
iii. Lap-SVM	16.8 ±0.2	18.8 ±0.6	22.3 ±0.0	25.0 ±0.6	33.2 ±0.3	37.8 ±0.2	41.3 ±0.2	44.1 ±0.4	46.0 ±0.3
iv. Lap-MKL	15.8 ±0.0	17.5 ±0.6	20.7 ±0.2	23.5 ±0.3	29.9 ±0.3	33.9 ±0.2	36.5 ±0.2	39.0 ±0.5	41.1 ±0.1
v. SVM	14.6 ±0.3	17.1 ±0.9	20.8 ±0.4	23.8 ±0.7	32.0 ±0.5	36.6 ±0.1	39.9 ±0.2	42.8 ±0.2	45.0 ±0.3
vi. MKL	11.8 ±0.2	14.4 ±0.5	17.5 ±0.5	20.3 ±0.6	26.7 ±0.7	31.0 ±0.1	33.8 ±0.3	36.9 ±0.5	38.6 ±0.4

Fig. 15. Transductive classification accuracies on Caltech256 dataset.



	2	3	4	5	10	15	20	25	30
iii. Lap-SVM	16.7 ±0.2	18.4 ±0.5	22.2 ±0.2	24.9 ±0.7	33.2 ±0.3	38.3 ±0.6	41.6 ±0.5	44.1 ±0.6	46.2 ±0.6
iv. Lap-MKL	14.8 ±0.2	16.8 ±0.6	20.1 ±0.2	22.4 ±0.6	29.5 ±0.1	34.1 ±0.4	36.9 ±0.5	38.6 ±0.4	40.1 ±0.1
v. SVM	15.3 ±0.2	16.9 ±0.9	20.7 ±0.3	23.8 ±0.9	32.1 ±0.2	37.2 ±0.4	40.5 ±0.7	43.3 ±0.4	45.5 ±0.6
vi. MKL	12.1 ±0.2	14.3 ±0.6	17.4 ±0.5	19.9 ±0.8	27.4 ±0.7	31.5 ±0.5	34.4 ±0.7	37.2 ±0.4	38.7 ±0.4
[42]				16.7	30.4	34.2	40.6	42.8	45.8
[44]						35.8			43.2
[41]						35.1			42.1
[45]				23.0	28.9	32.3			38.8

Fig. 16. Inductive classification accuracies on Caltech256 dataset [41,42,44,45].

maximum margin framework provides highly discriminative classifiers. Then, we give detailed analysis of these methods according to Table 7.

The effectiveness of the semi-supervised learning over the supervised one is confirmed by comparing iii. Lap-SVM to v. SVM, and iv. Lap-MKL to vi. MKL. The semi-supervised methods work quite well especially on fewer labeled samples (2–15), while the performance improvements are slightly attenuated on larger amount of labeled samples. The supervised methods are blessed with enough discriminative information of plenty of labeled samples, nevertheless the semi-supervised methods gain improvements even in such cases.

Surprisingly, the kernel weights learnt by MKTP (Section 5.1) are superior to those by MKL in both the semi-supervised and

supervised classifications, according to the comparison between iii. Lap-SVM to iv. Lap-MKL, and v. SVM to vi. MKL. The MKTP is formulated not particularly for optimizing the kernel weights but for enhancing the discriminativity of the transition probabilities (KTP). The (discriminative) characteristics of the kernel functions are extracted by the KTP, and thus the criterion to maximize the discriminative power of the KTP is so general as to be applicable for the kernel weights. Fig. 17 shows the relationships between the weights by MKTP and the individual classification performance of multiple kernel functions. Roughly speaking, those two quantities are positively correlated (correlation coefficient is 0.48), which indicates that the larger weights are assigned to the discriminative kernels. This is the evidence for the better performance of the classifiers using the composite kernel produced via MKTP.

As a result, iii. Lap-SVM, which utilizes both the composite kernel and similarity by MKTP for semi-supervised classifier, significantly outperforms the MKL method [25]. In the inductive classifications (Figs. 14 and 16) that enable us to fairly compare the semi-supervised with supervised methods, the performance of the Lap-SVM is favorably compared to those of the other prior works. It particularly outperforms the others in the case of fewer labeled samples (≤ 15 labeled samples) in virtue of the semi-supervised classification. The recent work [47] reports slightly better performance (47.4% at 30 labeled samples) by using powerful Fisher kernel features, while in this experiment our methods utilize multiple *weak* kernels. Finally, we show the efficiency of the proposed method in terms of the computation time in Fig. 18. The Lap-SVM using the proposed MKTP is greatly faster than the MKL methods; as the amount of labeled samples becomes large, the method gains greater efficiency, for example, over 100 times faster than [25] at 30 labeled samples. Note that the method [42] using the sample kernels as ours requires a similar computation time to the MKL [25]: the paper [42] reported the computation time of 3.4 h at 15 labeled samples.

We conclude that the proposed MKTP contributes to integrate the multiple kernel functions as well as the similarities. Apart from semi-supervised learning, we especially mention the applicability of the method to MKL even in the supervised setting. In such a case, MKTP is applied to the KTP only for producing the weights to integrate kernel functions without providing any similarity measure; the KTP is utilized just as 'steppingstone' for multiple kernel integration. As shown in the above-mentioned experimental results, v. supervised SVM that is built on the composite kernel by MKTP exhibits favorable performance, and besides the computation time for optimizing the kernel weights as well as learning the classifier is significantly short compared to the MKL methods.

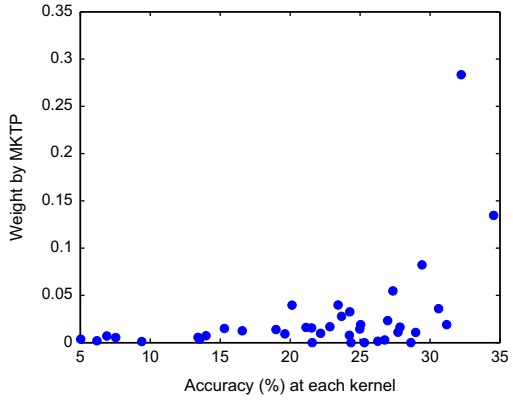


Fig. 17. Relationship between the weight by MKTP and individual performance of kernels. The 39 types of kernel are denoted by the dot points.

7. Conclusion

We have proposed methods to construct pair-wise similarity measure from the probabilistic viewpoint for improving performance of semi-supervised classification methods. The kernel-based transition probability (KTP) is first defined by using a single kernel function through the comparison between kernel-based and variational least squares in the probabilistic framework, which subsequently induces the similarities. It is simply formulated in a quadratic programming which flexibly introduces the constraint to improve practical robustness and is efficiently computed by applying SMO. From algebraic and geometrical viewpoints, the KTP is by nature favorably sparse even without ad hoc k-NN, and thereby the similarity measure derived from the KTP inherits such a characteristic. Besides, in order to cope with multiple types of kernel function which are practically available, we also proposed a method to effectively integrate them into a novel similarity via probabilistic formulation. The method discriminatively learns the linear weights for combining the multiple transition probabilities derived from multiple kernels, and the computation time required in that learning is quite low in disregard of number of classes. Those weights contribute to a composite similarity measure via combining KTPs straightforwardly as well as to integrate multiple kernel functions themselves as multiple kernel learning does. As a result, various types of multiple kernel based semi-supervised methods are proposed based on the method of multiple kernel integration. In the experiments on semi-supervised classifications using various datasets, the proposed similarities both of single and multiple kernels exhibit favorable performance compared to the other methods in transductive/inductive classifications. In addition, the proposed multiple kernel based semi-supervised methods

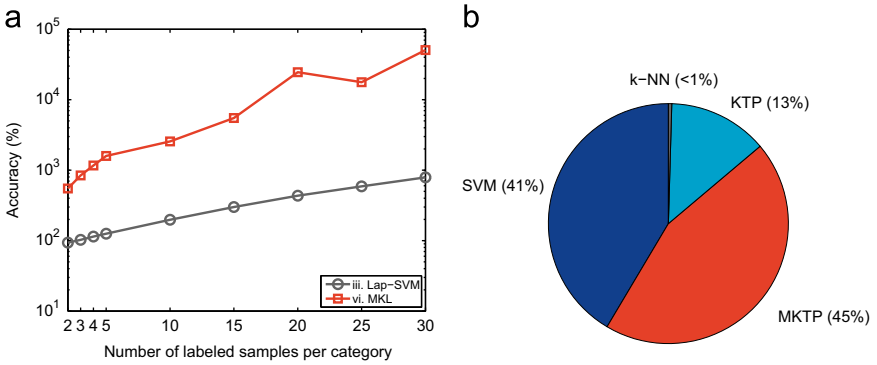


Fig. 18. Computation time in learning classifiers on Caltech256 dataset. Note that in iii. Lap-SVM, KTP is computed over the labeled and unlabeled samples; *e.g.*, at 15 labeled samples, those methods deal with $(15+30)\times 256 = 11\,520$ samples for KTP in total. (a) Comparison and (b) breakdown of Lap-SVM at 15 labeled samples.

are thoroughly compared and outperform the MKL methods in terms of classification accuracies and computation time.

Conflict of interest statement

None declared.

Appendix A. Details of derivation in probabilistic kernel least-squares

The kernel least-squares is defined in the probabilistic framework as the following minimization problem:

$$J(\boldsymbol{A}) = \sum_j \mathrm{p}(c_j) \sum_i \mathrm{p}(\boldsymbol{x}_i|c_j) \|\boldsymbol{e}_j - \boldsymbol{A}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)\|^2$$
$$= \mathrm{trace}\left\{ \boldsymbol{A}^\top \left(\sum_i \sum_j \mathrm{p}(c_j)\mathrm{p}(\boldsymbol{x}_i|c_j) \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)^\top \right) \boldsymbol{A} - 2\boldsymbol{A}^\top \left(\sum_j \mathrm{p}(c_j) \sum_i \mathrm{p}(\boldsymbol{x}_i|c_j) \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i) \boldsymbol{e}_j^\top \right) \right\} + 1.$$

Then, each term is rewritten by

$$(1)\ \sum_i \sum_j \mathrm{p}(c_j)\mathrm{p}(\boldsymbol{x}_i|c_j)\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)^\top = \sum_i \mathrm{p}(\boldsymbol{x}_i)\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)^\top$$
$$= \boldsymbol{K}\mathrm{diag}([\mathrm{p}(\boldsymbol{x}_1), \ldots, \mathrm{p}(\boldsymbol{x}_n)])\boldsymbol{K} = \boldsymbol{K}\boldsymbol{\Lambda}\boldsymbol{K},$$
$$(2)\ \sum_j \mathrm{p}(c_j) \sum_i \mathrm{p}(\boldsymbol{x}_i|c_j)\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)\boldsymbol{e}_j^\top = \sum_i \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i) \sum_j \mathrm{p}(\boldsymbol{x}_i|c_j)\mathrm{p}(c_j)\boldsymbol{e}_j^\top$$
$$= \sum_i \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)[\mathrm{p}(\boldsymbol{x}_i, c_1), \ldots, \mathrm{p}(\boldsymbol{x}_i, c_m)] = \sum_i \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)\mathbf{p}(\boldsymbol{x}_i, \boldsymbol{c})^\top = \boldsymbol{K}\boldsymbol{\Theta},$$

where $\boldsymbol{\Lambda} = \mathrm{diag}([\mathrm{p}(\boldsymbol{x}_1), \ldots, \mathrm{p}(\boldsymbol{x}_n)]) \in \mathbb{R}^{n\times n}$, $\mathbf{p}(\boldsymbol{x}_i, \boldsymbol{c}) = [\mathrm{p}(\boldsymbol{x}_i, c_1), \ldots, \mathrm{p}(\boldsymbol{x}_i, c_C)]^\top \in \mathbb{R}^C$, $\boldsymbol{\Theta} = [\mathbf{p}(\boldsymbol{x}_i, \boldsymbol{c}), \ldots, \mathbf{p}(\boldsymbol{x}_n, \boldsymbol{c})]^\top \in \mathbb{R}^{n\times C}$.

Thus, we obtain

$$J(\boldsymbol{A}) = \mathrm{trace}(\boldsymbol{A}^\top \boldsymbol{K}\boldsymbol{\Lambda}\boldsymbol{K}\boldsymbol{A} - 2\boldsymbol{A}^\top \boldsymbol{K}\boldsymbol{\Theta}) + 1,$$

and its minimizer is given by

$$\boldsymbol{A} = \boldsymbol{K}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Theta} = \boldsymbol{K}^{-1} \begin{bmatrix} \frac{\mathrm{p}(\boldsymbol{x}_1, c_1)}{\mathrm{p}(\boldsymbol{x}_1)} & \cdots & \frac{\mathrm{p}(\boldsymbol{x}_1, c_m)}{\mathrm{p}(\boldsymbol{x}_1)} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{p}(\boldsymbol{x}_n, c_1)}{\mathrm{p}(\boldsymbol{x}_n)} & \cdots & \frac{\mathrm{p}(\boldsymbol{x}_n, c_m)}{\mathrm{p}(\boldsymbol{x}_n)} \end{bmatrix}$$
$$= \boldsymbol{K}^{-1} \begin{bmatrix} \mathrm{p}(c_1|\boldsymbol{x}_1) & \cdots & \mathrm{p}(c_m|\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \mathrm{p}(c_1|\boldsymbol{x}_n) & \cdots & \mathrm{p}(c_m|\boldsymbol{x}_n) \end{bmatrix} = \boldsymbol{K}^{-1}\boldsymbol{P},$$

where we use $\mathrm{p}(c_j|\boldsymbol{x}_i) = \mathrm{p}(\boldsymbol{x}_i, c_j)/\mathrm{p}(\boldsymbol{x}_i)$ and $\boldsymbol{P} \in \mathbb{R}^{n\times C}$ is a posterior probability matrix of $P_{ij} = \mathrm{p}(c_j|\boldsymbol{x}_i)$.

Appendix B. The primal problem of KTP

We show that the problem (14) has the dual form (13). The Lagrangian of (14) is written as

$$L = \frac{1}{2}\|\boldsymbol{v}\|^2 + \xi - \alpha\{\boldsymbol{v}^\top \boldsymbol{\phi}_{\boldsymbol{x}} + b - 1 + \xi\} + C\sum_i \xi_i - \sum_i \alpha_i \left\{ -\boldsymbol{v}^\top \boldsymbol{\phi}_{\boldsymbol{x}_i} - b - 1 + \xi_i \right\} - \sum_i \beta_i \xi_i, \tag{B.1}$$

where $\alpha \geq 0,\ \alpha_i \geq 0,\ \beta_i \geq 0$. The derivatives are given by

$$\frac{\partial L}{\partial \xi} = 1 - \alpha = 0 \Rightarrow \alpha = 1, \tag{B.2}$$

$$\frac{\partial L}{\partial \boldsymbol{v}} = \boldsymbol{v} - \alpha\boldsymbol{\phi}_{\boldsymbol{x}} + \sum_i \alpha_i \boldsymbol{\phi}_{\boldsymbol{x}_i} = 0 \Rightarrow \boldsymbol{v} = \boldsymbol{\phi}_{\boldsymbol{x}} - \sum_i \alpha_i \boldsymbol{\phi}_{\boldsymbol{x}_i}, \tag{B.3}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \Rightarrow 0 \leq \alpha_i \leq C, \tag{B.4}$$

$$\frac{\partial L}{\partial b} = -\alpha + \sum_i \alpha_i = 0 \Rightarrow \sum_i \alpha_i = 1. \tag{B.5}$$

By using above relationships, we finally obtain the dual in the following form:

$$\max_{0 \leq \boldsymbol{\alpha} \leq C, \mathbf{1}^\top \boldsymbol{\alpha} = 1} -\frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{\Phi}_{\boldsymbol{X}}^\top \boldsymbol{\Phi}_{\boldsymbol{X}}\boldsymbol{\alpha} + \boldsymbol{\phi}_{\boldsymbol{x}}^\top \boldsymbol{\Phi}_{\boldsymbol{X}}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\phi}_{\boldsymbol{x}}^\top \boldsymbol{\phi}_{\boldsymbol{x}} + 1 + \sum_i \alpha_i \tag{B.6}$$

$$\Leftrightarrow \min_{0 \leq \boldsymbol{\alpha} \leq C, \mathbf{1}^\top \boldsymbol{\alpha} = 1} \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}). \tag{B.7}$$

This is exactly the same as (13).

Appendix C. Lower bound for the number of non-zeros in constrained KTP

The constrained KTP values in (15) are subject to

$$0 \leq \alpha_i \leq \frac{1}{\nu},\ \forall i, \quad \sum_i \alpha_i = 1.$$

Thus, we obtain the following lower bound for the number of non-zero α_i:

$$1 = \sum_i \alpha_i = \sum_{i|\alpha_i > 0} \alpha_i \leq \sum_{i|\alpha_i > 0} \frac{1}{\nu} = \frac{\|\boldsymbol{\alpha}\|_0}{\nu}, \quad \therefore\ \|\boldsymbol{\alpha}\|_0 \geq \nu,$$

where $\|\boldsymbol{\alpha}\|_0$ corresponds to the number of non-zero elements in $\boldsymbol{\alpha}$, that is the cardinality of $\{i|\alpha_i > 0\}$.

Appendix D. Semi-supervised simpleMKL

This appendix gives the details of the Laplacian simpleMKL (Lap-SMKL) in (25). Suppose reproducing kernel Hilbert space (RKHS) $\mathbf{H}^{[l]}$, $l = 1, \ldots, M$, each of which is endowed with an inner product via the kernel function $\mathrm{k}^{[l]}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]\top} \boldsymbol{\phi}_{\boldsymbol{x}_j}^{[l]}$, $\boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]} \in \mathbf{H}^{[l]}$. The simpleMKL [25] is formulated as

$$\min_{\boldsymbol{d}, \{\boldsymbol{v}^{[l]}\}_l, \{\xi_i\}_i} \frac{1}{2}\sum_l \frac{1}{d_l}\|\boldsymbol{v}^{[l]}\|^2 + C\sum_i \xi_i \tag{D.1}$$

$$\text{s.t.} \quad \forall i,\ y_i \left(\sum_l \boldsymbol{v}^{[l]\top} \boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]} + b \right) \geq 1 - \xi_i,\ \xi_i \geq 0,\ \forall l,\ d_l \geq 0, \sum_l d_l = 1, \tag{D.2}$$

where $\|\boldsymbol{v}\|^2 = \boldsymbol{v}^\top \boldsymbol{v}$, $\boldsymbol{d} \in \mathbb{R}^M$ is the weight for integrating the kernels and $\{\boldsymbol{v}^{[l]}\}_{l=1,\ldots,M}$ are the classifier vectors in respective RKHSs. This is proved to be a convex formulation and is optimized iteratively by applying the off-the-shelf SVM solver with fixing the weights $\boldsymbol{d}$.

We introduce the graph Laplacian (1) to (D.1) for incorporating unlabeled samples, which results in

$$\min_{\boldsymbol{d}, \{\boldsymbol{v}^{[l]}\}_{l=1,\ldots,M}} \frac{1}{2}\sum_l \frac{1}{d_l}\|\boldsymbol{v}^{[l]}\|^2 + \frac{\theta}{4}\sum_l \frac{1}{d_l} \sum_{ij}^{n+N} S_{ij}\|\boldsymbol{v}^{[l]\top}(\boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]} - \boldsymbol{\phi}_{\boldsymbol{x}_j}^{[l]})\|^2 + C\sum_i \xi_i$$
$$\text{s.t.} \quad \forall i \in \{1, \ldots, n\},\ y_i \left(\sum_l^M \boldsymbol{v}^{[l]\top} \boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]} + b \right) \geq 1 - \xi_i,\ \xi_i \geq 0,\ \forall l,\ d_l \geq 0, \sum_l^M d_l = 1, \tag{D.3}$$

where the n samples of $i = 1, \ldots, n$ are labeled while the rest N samples of $i = n+1, \ldots, n+N$ are unlabeled. The graph Laplacian works as a regularization to enforce that similar samples take similar classification outputs. The graph Laplacian is obviously convex, which does not deteriorate the convexity of (D.1), and consequently the formulation (25) is also convex. The Lagrangian

is given by

$$L=\frac{1}{2}\sum_{l}^{M}\frac{1}{d_l}\|\boldsymbol{v}^{[l]}\|^2+\frac{\theta}{4}\sum_{l}^{M}\frac{1}{d_l}\sum_{ij}^{n+N}S_{ij}\|\boldsymbol{v}^{[l]\top}(\boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]}-\boldsymbol{\phi}_{\boldsymbol{x}_j}^{[l]})\|^2+C\sum_{i}^{n}\xi_i \tag{D.4}$$

$$-\sum_{i}^{n}\alpha_i\left\{y_i\left(\sum_{l}\boldsymbol{v}^{[l]\top}\boldsymbol{\phi}_{\boldsymbol{x}_i}^{[l]}+b\right)-1+\xi_i\right\}-\sum_{i}^{n}\beta_i\xi_i+\gamma\left(\sum_{l}^{M}d_l-1\right)-\sum_{l}^{M}\eta_l d_l, \tag{D.5}$$

and its derivatives are also written as

$$\frac{\partial L}{\partial b}=\sum_{i}^{n}\alpha_i y_i=0, \tag{D.6}$$

$$\frac{\partial L}{\partial \xi_i}=C-\alpha_i-\beta_i=0, \tag{D.7}$$

$$\frac{\partial L}{\partial d_l}=-\frac{1}{2d_l^2}\boldsymbol{v}^{[l]\top}(\boldsymbol{I}+\theta\boldsymbol{\Phi}^{[l]}\boldsymbol{L}\boldsymbol{\Phi}^{[l]\top})\boldsymbol{v}^{[l]}+\gamma-\eta_l=0, \tag{D.8}$$

$$\frac{\partial L}{\partial \boldsymbol{v}^{[l]}}=\frac{1}{d_l}(\boldsymbol{I}+\theta\boldsymbol{\Phi}^{[l]}\boldsymbol{L}\boldsymbol{\Phi}^{[l]\top})\boldsymbol{v}^{[l]}-\boldsymbol{\Phi}^{[l]}\boldsymbol{J}(\boldsymbol{\alpha}\circ\boldsymbol{y})=0, \tag{D.9}$$

$$\therefore\ \boldsymbol{v}^{[l]}=d_l\boldsymbol{\Phi}^{[l]}(\boldsymbol{I}+\theta\boldsymbol{L}\boldsymbol{K}^{[l]})\boldsymbol{J}(\boldsymbol{\alpha}\circ\boldsymbol{y}), \tag{D.10}$$

where $\circ$ denotes the Hadamard product, $\boldsymbol{K}^{[l]}=\boldsymbol{\Phi}^{[l]\top}\boldsymbol{\Phi}^{[l]}\in\mathbb{R}^{(n+N)\times(n+N)}$, $\boldsymbol{J}=[\boldsymbol{I},\boldsymbol{0}]$ of which the first n columns form the identity matrix $\boldsymbol{I}\in\mathbb{R}^{n\times n}$ and the others are zeros, and we use the following matrix algebra to get (D.10):

$$(\boldsymbol{I}+\theta\boldsymbol{\Phi}^{[l]}\boldsymbol{L}\boldsymbol{\Phi}^{[l]\top})^{-1}\boldsymbol{\Phi}^{[l]}=\{\boldsymbol{I}-\theta\boldsymbol{\Phi}^{[l]}(\boldsymbol{L}^{-1}+\theta\boldsymbol{\Phi}^{[l]\top}\boldsymbol{\Phi}^{[l]})^{-1}\boldsymbol{\Phi}^{[l]\top}\}\boldsymbol{\Phi}^{[l]} \tag{D.11}$$

$$=\boldsymbol{\Phi}^{[l]}\{\boldsymbol{I}-(\boldsymbol{L}^{-1}+\theta\boldsymbol{K}^{[l]})^{-1}\theta\boldsymbol{K}^{[l]}\} \tag{D.12}$$

$$=\boldsymbol{\Phi}^{[l]}\{(\boldsymbol{L}^{-1}+\theta\boldsymbol{K}^{[l]})^{-1}(\boldsymbol{L}^{-1}+\theta\boldsymbol{K}^{[l]}-\theta\boldsymbol{K}^{[l]})\}=\boldsymbol{\Phi}^{[l]}(\boldsymbol{I}+\theta\boldsymbol{L}\boldsymbol{K}^{[l]})^{-1}. \tag{D.13}$$

By using (D.6)–(D.10), the dual problem is obtained as

$$\max_{\boldsymbol{\alpha},\gamma}\quad\sum_{i}\alpha_i-\gamma \tag{D.14}$$

$$\text{s.t.}\quad\forall l,\ \frac{1}{2}(\boldsymbol{\alpha}\circ\boldsymbol{y})^{\top}\boldsymbol{J}^{\top}\boldsymbol{K}^{[l]}(\boldsymbol{I}+\theta\boldsymbol{L}\boldsymbol{K}^{[l]})^{-1}\boldsymbol{J}(\boldsymbol{\alpha}\circ\boldsymbol{y})\le\gamma, \tag{D.15}$$

$$\forall i,\ 0\le\alpha_i\le C,\ \sum_{i}y_i\alpha_i=0. \tag{D.16}$$

The lap-MKL is actually optimized by iteratively applying SVM solver to the following subproblem under the fixed wight $\boldsymbol{d}$:

$$\max_{\boldsymbol{\alpha}}\quad-\frac{1}{2}(\boldsymbol{\alpha}\circ\boldsymbol{y})^{\top}\boldsymbol{J}^{\top}\left\{\sum_{l}^{M}d_l\boldsymbol{K}^{[l]}(\boldsymbol{I}+\theta\boldsymbol{L}\boldsymbol{K}^{[l]})^{-1}\right\}\boldsymbol{J}(\boldsymbol{\alpha}\circ\boldsymbol{y})+\sum_{i}^{n}\alpha_i \tag{D.17}$$

$$\text{s.t.}\quad\forall i,\ 0\le\alpha_i\le C,\ \sum_{i}y_i\alpha_i=0, \tag{D.18}$$

while the weight $\boldsymbol{d}$ is optimized via the projected gradient descent [25] using

$$\frac{\partial J}{\partial d_l}=-\frac{1}{2}(\boldsymbol{\alpha}\circ\boldsymbol{y})^{\top}\boldsymbol{J}^{\top}\boldsymbol{K}^{[l]}(\boldsymbol{I}+\theta\boldsymbol{L}\boldsymbol{K}^{[l]})^{-1}\boldsymbol{J}(\boldsymbol{\alpha}\circ\boldsymbol{y}), \tag{D.19}$$

$$\text{s.t.}\ \forall l,\ d_l\ge0,\sum_{l}d_l=1. \tag{D.20}$$

In the above formulations, the lap-MKL is analogous to the original simpleMKL in that the kernel Gram matrix $\boldsymbol{K}^{[l]}$ is replaced with the modified one $\boldsymbol{J}\boldsymbol{K}^{[l]}(\boldsymbol{I}+\theta\boldsymbol{L}\boldsymbol{K}^{[l]})^{-1}\boldsymbol{J}$ based on the graph Laplacian.

References

[1] O. Chapelle, B. Schölkopf, A. Zien, Semi-Supervised Learning, MIT Press, Cambridge, MA, USA, 2006.
[2] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Annual Conference on Computational Learning Theory, Madison, WI, USA, 1998, pp. 92–100.
[3] A. Levin, P. Viola, Y. Freund, Unsupervised improvement of visual detectors using co-training, in: International Conference on Computer Vision, Nice, France, 2003, pp. 626–633.
[4] R. Yan, M. Naphade, Semi-supervised cross feature learning for semantic concept detection in videos, in: IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 2005, pp. 657–663.
[5] C.M. Christoudias, Probabilistic Models for Multi-View Semi-Supervised Learning and Coding (Ph.D. thesis), Massachusetts Institute of Technology, September 2009.
[6] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[7] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, in: International Conference on Machine Learning, Washington, DC, USA, 2003, pp. 912–919.
[8] H. Cheng, Z. Liu, J. Yang, Sparsity induced similarity measure for label propagation, in: International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 317–324.
[9] F. Wang, C. Zhang, Label propagation through linear neighborhoods, IEEE Trans. Knowl. Data Eng. 20 (1) (2008) 55–67.
[10] W. Liu, S.-F. Chang, Robust multi-class transductive learning with graphs, in: IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 381–388.
[11] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1–7.
[12] Y. Zhang, D.-Y. Yeung, Semi-supervised discriminant analysis using robust path-based similarity, in: IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1–8.
[13] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[14] T. Kobayashi, K. Watanabe, N. Otsu, Logistic label propagation, Pattern Recognition Lett. 33 (5) (2012) 580–588.
[15] N. Sokolovska, Aspects of semi-supervised and active learning in conditional random fields, in: European Conference on Machine learning and Principles and Practice of Knowledge Discovery in Databases, Athens, Greece, 2011, pp. 273–288.
[16] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, Berlin, Germany, 2007.
[17] N. Otsu, Optimal linear and nonlinear solutions for least-square discriminant feature extraction, in: International Conference on Pattern Recognition, Munich, Germany, 1982, pp. 557–560.
[18] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, USA, 1999, pp. 185–208.
[19] G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, M.I. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5 (2004) 27–72.
[20] T. Kobayashi, N. Otsu, Efficient similarity derived from kernel-based transition probability, in: European Conference on Computer Vision, Florence, Italy, 2012, pp. 371–385.
[21] J. Tang, X.-S. Hua, Y. Song, G.-J. Qi, X. Wu, Kernel-based linear neighborhood propagation for semantic video annotation, in: Pacific–Asia Conference on Advances in Knowledge Discovery and Data, Nanjing, China, 2007, pp. 793–800.
[22] T. Kobayashi, N. Otsu, One-class label propagation using local cone based similarity, in: IEEE Conference on Automatic Face and Gesture Recognition, Santa Barbara, CA, USA, 2011, pp. 394–399.
[23] K. Tsuda, H. Shin, B. Schölkopf, Fast protein classification with multiple networks, Bioinformatics 21 (2) (2005) ii59–ii65.
[24] S. Wang, Q. Huang, S. Jiang, Q. Tian, S^{3}mkl: scalable semi-supervised multiple kernel learning for real-world image applications, IEEE Trans. Multimedia 14 (4) (2012) 1259–1274.
[25] A. Rakotomamonjy, F.R. Bach, S. Canu, Y. Grandvalet, Simplemkl, J. Mach. Learn. Res. 9 (2008) 2491–2521.
[26] V. Vapnik, Statistical Learning Theory, Wiley, New York, NY, USA, 1998.
[27] D.M. Tax, R.P. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.
[28] B. Schölkopf, A. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, USA, 2002.
[29] C.-C. Chang, C.-J. Lin, LIBSVM: A Library for Support Vector Machines, Software Available at ⟨http://www.csie.ntu.edu.tw/∼cjlin/libsvm⟩, 2001.
[30] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.
[31] J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554.
[32] R. Bro, S. Jong, A fast non-negativity-constrained least squares algorithm, J. Chemometrics 11 (5) (1997) 393–401.

[33] P.J. Bartlett, B. Schölkopf, D. Schuurmans, A.J. Smola, Advances in Large-Margin Classifiers, MIT Press, Cambridge, MA, USA, 2000.
[34] D.B. Graham, N.M. Allinson, Characterizing virtual eigensignatures for general purpose face recognition, face recognition: from theory to applications, NATO ASI Ser. F: Comput. Syst. Sci. 163 (1998) 446–456.
[35] F. Wang, X. Wang, T. Li, Beyond the graphs: semi-parametric semi-supervised discriminant analysis, in: IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 2113–2120.
[36] S. Lazebnik, C. Schmid, J. Ponce, A maximum entropy framework for part-based texture and object recognition, in: International Conference on Computer Vision, Beijing, China, 2005, pp. 832–838.
[37] S. Lazebnik, C. Schmid, J. Ponce, Semi-local affine parts for object recognition, in: British Machine Vision Conference, London, UK, 2004, pp. 779–788.
[38] C. Christoudias, R. Urtasun, M. Salzmann, T. Darrell, Learning to recognize objects from unseen modalities, in: European Conference on Computer Vision, Crete, Greece, 2010, pp. 677–691.
[39] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Trans. Pattern Anal. Mach. Intell. 28 (4) (2006) 594–611.
[40] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, Technical Report 7694, Caltech, 2007.
[41] K. Sohn, D.Y. Jung, H. Lee, A.O. Hero III, Efficient learning of sparse, distributed, convolutional feature representations for object recognition, in: International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2643–2650.
[42] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 221–228.
[43] Y. Xu, Y. Quan, Z. Zhang, H. Ji, C. Fermüller, M. Nishigaki, D. Dementhon, Contour-based recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 3402–3409.
[44] J. Feng, B. Ni, Q. Tian, S. Yan, Geometric l_p norm feature pooling for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011, pp. 2609–2616.
[45] C. Kanan, G. Cottrell, Robust classification of objects, faces, and flowers using natural image statistics, in: IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 2472–2479.
[46] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple kernels for object detection, in: International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 606–613.
[47] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image Classification with the Fisher Vector: Theory and Practice, Technical Report RR-8209, INRIA Research, 2013.

Takumi Kobayashi received Ms. Eng. from University of Tokyo in 2005 and Dr. Eng. from University of Tsukuba in 2009. He was a researcher at Toshiba Corporation in 2006 and then joined National Institute of Advanced Industrial Science and Technology (AIST), Japan, in 2007. His research interest includes pattern recognition.