Efficient Similarity Derived from Kernel-Based Transition Probability

Takumi Kobayashi and Nobuyuki Otsu

National Institute of Advanced Industrial Science and Technology,
1-1-1, Umezono, Tsukuba, Japan
{takumi.kobayashi,otsu.n}@aist.go.jp

Abstract. Semi-supervised learning effectively integrates labeled and unlabeled samples for classification, and most of the methods are founded on the pair-wise similarities between the samples. In this paper, we propose methods to construct similarities from the probabilistic viewpoint, whilst the similarities have so far been formulated in a heuristic manner such as by k-NN. We first propose the kernel-based formulation of transition probabilities via considering kernel least squares in the probabilistic framework. The similarities are consequently derived from the kernel-based transition probabilities which are efficiently computed, and the similarities are inherently sparse without applying k-NN. In the case of multiple types of kernel functions, the multiple transition probabilities are also obtained correspondingly. From the probabilistic viewpoint, they can be integrated with prior probabilities, *i.e.*, linear weights, and we propose a computationally efficient method to optimize the weights in a discriminative manner, as in multiple kernel learning. The novel similarity is thereby constructed by the composite transition probability and it benefits the semi-supervised learning methods as well. In the various experiments on semi-supervised learning problems, 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 recognition have been developed mainly to deal with labeled samples in the framework of supervised learning. In practice, however, the process of labeling samples requires exhaustive labor especially for large-scaled samples. On the other hand, we can easily obtain unlabeled samples, and thus semi-supervised learning methods have attracted keen attentions to incorporate such unlabeled samples for classification [1–9].

Most of the semi-supervised learning methods are based on a graph structure. In the graph, samples (nodes) are linked each other by weighted edges according to the similarities between the samples [10]. The unlabeled samples are incorporated in the graph and the optimization problems are formulated based on the energy over the graph; for example, the label propagation methods [1–4] directly estimate the labels of the unlabeled samples by minimizing the graph energy with

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VI, LNCS 7577, pp. 371–385, 2012.

the information of the labeled samples, and the other semi-supervised methods can be developed by incorporating the graph energy to the optimization problem defined in the supervised manner [5–8]. Thus, the similarities are fundamental for the semi-supervised learning methods, and it is an important issue how to construct the similarities for improving the performance.

The most commonly used similarity is based on the Gaussian kernel $\exp(-\|\boldsymbol{x}-\boldsymbol{y}\|^2/h)$ on neighboring samples, called Gaussian kernel similarity (GKS). This is an ad-hoc model solely depending on the Euclidean distance between sample feature vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. The parameter value of the bandwidth h and the number of neighbors have to be determined in advance, which requires exhaustive labors. In recent years, more sophisticated methods have been proposed for the similarities by considering the linear relationship among sample vectors [2, 3]. The models employed in those methods, however, are derived somewhat heuristically. There are some other works [4, 6] 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 similarities between samples from the probabilistic viewpoint. In the probabilistic framework, by comparing the kernel least squares to the variational least squares [11] that gives Bayesian optimal solution, we first propose the kernel-based formulation to approximate the transition probabilities between samples. The inherently sparse similarities are then derived from the kernel-based transition probabilities which are actually computed by using kernel functions. We also present the method to compute the similarity in a low computation time. In the case that multiple types of kernel functions are defined, we correspondingly obtain multiple transition probabilities which are probabilistically integrated with prior probabilities, *i.e.*, linear weights. We propose a method to efficiently optimize the weights in a discriminative manner, as in multiple kernel learning [12], and thus the multiple transition probabilities are effectively combined into a new composite one, resulting in a novel similarity. The similarity derived from multiple kernels benefits the semi-supervised learning methods as well.

Our contributions are 1) to propose a method for producing kernel-based transition probability by comparing the kernel least squares to the variational one in the probabilistic framework, and thereby 2) to construct the inherently sparse similarity without requiring k-NN, and 3) to propose a method for integrating multiple kernels into the similarity via the probabilistic formulation.

1.1 Related Works

There are some works to formulate the similarity itself other than GKS.

The linear neighborhood propagation (LNP) [3] 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 induce the method for integrating multiple kernels. In addition, we provide the computationally efficient method to compute them. The kernel LNP [3]

has also been proposed in [13] but it differs from our method in that it lacks a probabilistic constraint. Cheng *et al*. [2] 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 [3]. Such linear dependency (equality), however, is a too strong constraint to hold, especially in a high dimensional feature space. Zhang and Yeung [6] 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 [4] recently proposed an interesting method to produce a discriminative similarity. In that method, the similarity is sequentially updated by using the information of the labeled samples in the semi-supervised framework, although the method also starts from the GKS-based initial graph. From the viewpoint of optimizing the similarity, it is slightly close to the proposed method that integrates multiple kernels (Sec.3). 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.

2 Proposed Similarity

In the probabilistic framework, we first compare the kernel least squares to the variational one [11] that produces Bayesian optimal solution, and then propose the kernel-based formulation of the transition probabilities between samples. Finally, we derive the similarities from the kernel-based transition probabilities.

2.1 Kernel Least Squares in Probabilistic Framework

Let $\boldsymbol{x} \in \mathbb{R}^D$ be the input vector and $\boldsymbol{y} \in \mathbb{R}^m$ be a (multiple) objective variable(s) associated with $\boldsymbol{x}$. In the kernel least squares, the objective $\boldsymbol{y}$ is modeled by the following regression form using a kernel function k:

$$\boldsymbol{y} = \boldsymbol{A}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}) + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}) = [\mathsf{k}(\boldsymbol{x}_1, \boldsymbol{x}), \cdots, \mathsf{k}(\boldsymbol{x}_n, \boldsymbol{x})]^\top \in \mathbb{R}^n$, n is the number of samples, $\boldsymbol{A} \in \boldsymbol{R}^{n \times m}$ is a coefficient matrix, and $\boldsymbol{\epsilon}$ indicate residual errors. Here, we suppose a m-class problem. Let c_j $(j = 1, .., m)$ denote the j-th class and $\boldsymbol{e}_j$ be the m-dimensional binary vector representing the j-th class, in which only the j-th element is 1 and the others are 0. Regarding those class-representative vectors $\boldsymbol{e}_j$ as the targets, the optimum coefficients $\boldsymbol{A}$ are obtained by the least-squares method in the following probabilistic framework:

$$E[||\boldsymbol{\epsilon}||^2] = \sum_j p(c_j) \sum_i p(\boldsymbol{x}_i|c_j) \|\boldsymbol{e}_j - \boldsymbol{A}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}_i)\|^2 \tag{2}$$

$$= \mathrm{trace}\left(\boldsymbol{A}^\top \boldsymbol{K} \boldsymbol{\Lambda} \boldsymbol{K} \boldsymbol{A} - 2\boldsymbol{A}^\top \boldsymbol{K} \boldsymbol{\Theta} + 1\right) \rightarrow \min_{\boldsymbol{A}} \tag{3}$$

$$\therefore \boldsymbol{A} = \boldsymbol{K}^{-1} \boldsymbol{\Lambda}^{-1} \boldsymbol{\Theta}, \tag{4}$$

where $\boldsymbol{K}$ is a (nonsingular) kernel Gram matrix of $K_{ij} = \mathsf{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\boldsymbol{\Lambda} = \mathrm{diag}([p(\boldsymbol{x}_1), ..., p(\boldsymbol{x}_n)]) \in \mathbb{R}^{n \times n}$, $\boldsymbol{p}(\boldsymbol{x}_i, \boldsymbol{c}) = [p(\boldsymbol{x}_i, c_1), .., p(\boldsymbol{x}_i, c_m)]^\top \in \mathbb{R}^m$, $\boldsymbol{\Theta} = [\boldsymbol{p}(\boldsymbol{x}_i, \boldsymbol{c}), .., \boldsymbol{p}(\boldsymbol{x}_n, \boldsymbol{c})]^\top \in \mathbb{R}^{n \times m}$. By using $p(c_j|\boldsymbol{x}_i) = p(\boldsymbol{x}_i, c_j)/p(\boldsymbol{x}_i)$, we obtain the following representation,

$$\boldsymbol{A} = \boldsymbol{K}^{-1}\boldsymbol{\Lambda}^{-1}\boldsymbol{\Theta} = \boldsymbol{K}^{-1}\boldsymbol{P}, \tag{5}$$

where $\boldsymbol{P} \in \mathbb{R}^{n \times m}$ is a posterior probability matrix of $P_{ij} = p(c_j|\boldsymbol{x}_i)$. Thus, the objective values are estimated by

$$\hat{\boldsymbol{e}} = \boldsymbol{P}^\top \boldsymbol{K}^{-1}\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}). \tag{6}$$

On the other hand, we also consider a general model by using a (non-linear) function $\mathbf{q}$ as $\boldsymbol{e} = \mathbf{q}(\boldsymbol{x}) + \boldsymbol{\epsilon}$. Otsu [11] showed that the optimum function $\mathbf{q}$ is obtained by applying the variational method in the least squares:

$$L \triangleq \sum_j p(c_j) \int p(\boldsymbol{x}|c_j) \|\boldsymbol{e}_j - \mathbf{q}(\boldsymbol{x})\|^2 d\boldsymbol{x} \to \min_{\boldsymbol{q}} \tag{7}$$

$$\delta L = 2 \int \delta\mathbf{q}(\boldsymbol{x})^\top \left[\sum_j p(c_j) p(\boldsymbol{x}|c_j) \left\{ \boldsymbol{e}_j - \mathbf{q}(\boldsymbol{x}) \right\} \right] d\boldsymbol{x}, \tag{8}$$

$$\Rightarrow \boldsymbol{p}(\boldsymbol{x}, \boldsymbol{c}) - p(\boldsymbol{x})\mathbf{q}(\boldsymbol{x}) = \mathbf{0}, \quad \therefore \mathbf{q}(\boldsymbol{x}) = [p(c_1|\boldsymbol{x}), .., p(c_m|\boldsymbol{x})]^\top = \boldsymbol{p}(\boldsymbol{c}|\boldsymbol{x}). \tag{9}$$

Thus, the class-representative $\boldsymbol{e}$ is optimally approximated by the posterior probability for the classes and it is further decomposed as follows:

$$\hat{\boldsymbol{e}} = \boldsymbol{p}(\boldsymbol{c}|\boldsymbol{x}) = \int \boldsymbol{p}(\boldsymbol{c}|\tilde{\boldsymbol{x}}) p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\tilde{\boldsymbol{x}} \approx \sum_i \boldsymbol{p}(\boldsymbol{c}|\boldsymbol{x}_i) p(\boldsymbol{x}_i|\boldsymbol{x}) = \boldsymbol{P}^\top [p(\boldsymbol{x}_1|\boldsymbol{x}), .., p(\boldsymbol{x}_n|\boldsymbol{x})]^\top. \tag{10}$$

Comparing (6) to (10), we can find that $\boldsymbol{\alpha} \triangleq \boldsymbol{K}^{-1}\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x})$ is the kernel-based approximation of the transition probabilities $[p(\boldsymbol{x}_1|\boldsymbol{x}), .., p(\boldsymbol{x}_n|\boldsymbol{x})]^\top$.

The kernel least squares, however, produces unconstrained $\boldsymbol{\alpha}$ which might take any values, while the transition probabilities are subject to the probability constraints of non-negativity $p(\boldsymbol{x}_i|\boldsymbol{x}) \geq 0$ and unit sum $\sum_i p(\boldsymbol{x}_i|\boldsymbol{x}) = 1$. We impose these constraints on $\boldsymbol{\alpha}$ in order to approximate the transition probability more accurately from the probabilistic perspective.

2.2 Kernel-Based Transition Probability

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

$$\boldsymbol{\alpha} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\Phi}_{\boldsymbol{X}}\boldsymbol{\alpha} - \boldsymbol{\phi}_{\boldsymbol{x}}\|^2 = \left(\boldsymbol{\Phi}_{\boldsymbol{X}}^\top \boldsymbol{\Phi}_{\boldsymbol{X}}\right)^{-1} \boldsymbol{\Phi}_{\boldsymbol{X}}^\top \boldsymbol{\phi}_{\boldsymbol{x}} = \boldsymbol{K}^{-1}\mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}), \tag{11}$$

where $\boldsymbol{x}$ is represented by $\boldsymbol{\phi}_{\boldsymbol{x}}$ in RKHS ($\mathsf{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}_{\boldsymbol{x}_i}^\top \boldsymbol{\phi}_{\boldsymbol{x}_j}$) and $\boldsymbol{\Phi}_{\boldsymbol{X}} = [\boldsymbol{\phi}_{\boldsymbol{x}_1}, .., \boldsymbol{\phi}_{\boldsymbol{x}_n}]$. By imposing the probability constraints on (11), we propose the

kernel-based formulation to approximate the transition probabilities more accurately:

$$\min_{\boldsymbol{\alpha}\geq 0,\, \sum_i \alpha_i = 1} \boldsymbol{\alpha}^\top \boldsymbol{K}\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^\top \mathbf{k}_{\boldsymbol{X}}(\boldsymbol{x}) + \mathsf{k}(\boldsymbol{x},\boldsymbol{x}). \tag{12}$$

This can be viewed as a kernel-based extension of LNP [3], but is different from the kernel LNP [13] which lacks the probability constraint $\boldsymbol{\alpha} \geq 0$.

(12) is a convex quadratic programming (QP) which is usually solved by using standard QP solvers, such as MOSEK optimization toolbox [14]. However, it requires significant computational cost and is inapplicable to large-scaled samples. We find that (12) is almost the same formulation as the dual problem in SVM [15] except for the linear term of $\boldsymbol{\alpha}$. Various approaches have been developed to efficiently solve the SVM dual problem, and in this study, we apply the SMO-based QP solver in LIBSVM [16] to solve (12), which enables the proposed method computationally efficient and thus applicable to large-scaled samples. We call the resultant $\boldsymbol{\alpha}$ by (12) as the *kernel-based transition probability* (KTP).

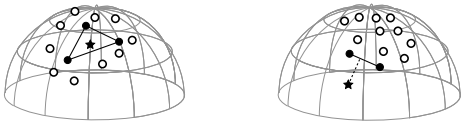
The KTP values can also be interpreted from the geometrical viewpoint. We suppose that the vector $\boldsymbol{\phi}_{\boldsymbol{x}}$ lies on the unit hyper sphere in RKHS; the kernel function is often (or inherently) normalized, *i.e.*, $\mathsf{k}(\boldsymbol{x},\boldsymbol{x}) = \boldsymbol{\phi}_{\boldsymbol{x}}^\top \boldsymbol{\phi}_{\boldsymbol{x}} = 1, \forall \boldsymbol{x}$. The optimization (12) is also regarded as the projection from $\boldsymbol{\phi}_{\boldsymbol{x}}$ to the convex hull that are spanned by the sample vectors $\boldsymbol{\Phi}_{\boldsymbol{X}}$. When $\boldsymbol{\phi}_{\boldsymbol{x}}$ is contained in the convex cone by $\boldsymbol{\Phi}_{\boldsymbol{X}}$, the closer hull is selected to minimize the distance from $\boldsymbol{\phi}_{\boldsymbol{x}}$ to the hull (Fig. 1a). 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. 1b). Thus, KTP has only a few non-zero elements associated with the samples near by the input in RKHS that span such a convex hull. In other words, the KTP results in sparse favorably even without ad-hoc k-NN nor sparsity constraints, as shown in Fig. 2. Therefore, We employ the *normalized* kernel function to obtain the KTP in (12).

In the Gaussian process (GP) [17], the kernel least squares also emerges in a probabilistic manner. But, the GP assumes the parametric (Gaussian) model for the whole samples and it can not give explicit connection to the pair-wise transition probability that is our main concern for inducing similarity measure; the form (6) is not actually obtained in the GP. In this paper, by comparing the kernel-based and the variational approaches based on the identical criterion, *i.e.*, least squares, in the probabilistic framework, we propose the kernel-based transition probability (KTP) as described above. The proposed KTP benefits to construction of the similarity in Sec.2.3 as well as multiple kernel integration in Sec.3.

2.3 KTP-Based Similarity

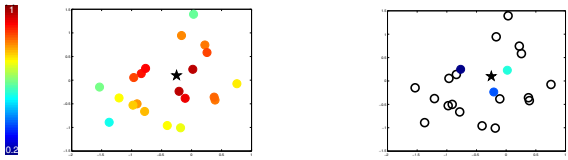
We derive the similarities from the kernel-based transition probabilities (KTP). We calculate the KTP $\boldsymbol{\alpha}$ from respective $\boldsymbol{x}_i$ to the others in a leave-one-out scheme; at the i-th sample, (12) is solved for $\boldsymbol{\Phi}_{\boldsymbol{X}} = [.., \boldsymbol{\phi}_{\boldsymbol{x}_{i-1}}, \boldsymbol{\phi}_{\boldsymbol{x}_{i+1}}, ..]$ and $\boldsymbol{\phi}_{\boldsymbol{x}_i}$ to produce the $\boldsymbol{\alpha}_i$ in which $\alpha_{ij} = p(\boldsymbol{x}_j|\boldsymbol{x}_i)$ and $\alpha_{ii} = 0$.[1]

[1] Since the self similarity does not affect the graph Laplacian [10], we simply set $\alpha_{ii} = 0$.



(a) Inside (b) Outside

Fig. 1. 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 (12), and black solid lines show the contour of the convex hull spanned by those black points.



(a) Gaussian kernel values (b) KTP

Fig. 2. 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. (This figure is best viewed in color.)

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

$$D(\boldsymbol{x}_j||\boldsymbol{x}_i) = -\log p(\boldsymbol{x}_j|\boldsymbol{x}_i), \qquad \bar{D}(\boldsymbol{x}_i, \boldsymbol{x}_j) = D(\boldsymbol{x}_j||\boldsymbol{x}_i) + D(\boldsymbol{x}_i||\boldsymbol{x}_j). \tag{13}$$

This is a symmetric metric as in symmetrized Kullback-Leibler divergence [18]. Thereby, the similarity is simply formulated by using this metric as

$$s_{ij} \triangleq \exp\{-\bar{D}(\boldsymbol{x}_i, \boldsymbol{x}_j)\} = p(\boldsymbol{x}_j|\boldsymbol{x}_i)p(\boldsymbol{x}_i|\boldsymbol{x}_j) = \alpha_{ij}\alpha_{ji}. \tag{14}$$

This KTP-based similarity, called KTPS, ranges from 0 to 1 and the sparsity is further enhanced than the KTP since $s_{ij} > 0$ *iff* $\alpha_{ij} > 0 \bigwedge \alpha_{ji} > 0$.

3 Multiple Kernel Integration for KTPS

As described above, by using a (single) kernel function, we derive the kernel-based transition probability and consequently KTPS. In the case that multiple types of kernel function are given, we obtain multiple transition probabilities correspondingly. As in multiple kernel learning (MKL) [12], it is desirable to integrate those multiple transition probabilities such that the resulting KTPS has high discriminative power.

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

$$p(\boldsymbol{x}_j|\boldsymbol{x}_i) = \sum_{l=1}^{M} p(\mathsf{k}^{[l]}) p(\boldsymbol{x}_j|\boldsymbol{x}_i, \mathsf{k}^{[l]}) = \sum_{l=1}^{M} w_l \alpha_{ij}^{[l]}, \tag{15}$$

where $p(\mathsf{k}^{[l]})$ is the prior probability of the l-th type of kernel and in this study, it is regarded as the weight parameter w_l to be optimized subject to $w_l \geq 0, \sum_l w_l = 1$, as follows.

We have no prior knowledge about $w_l \triangleq p(\mathsf{k}^{[l]})$ and thus optimize it in a discriminative manner using labeled samples. Let $\mathcal{G}$ be the set of labeled samples. The labeled sample pairs in $\mathcal{G} \times \mathcal{G}$ are categorized into $\mathcal{P} = \{(i,j)|c_i = c_j, i,j \in \mathcal{G}\}$ and $\mathcal{N} = \{(i,j)|c_i \neq c_j, i,j \in \mathcal{G}\}$. For each labeled sample $i \in \mathcal{G}$, from the discriminative perspective, it is expected that the transition probability to the same class, $\sum_{j|(i,j)\in\mathcal{P}} p(\boldsymbol{x}_j|\boldsymbol{x}_i)$, is high, while that to the different class, $\sum_{j|(i,j)\in\mathcal{N}} p(\boldsymbol{x}_j|\boldsymbol{x}_i)$, is low. Thus, we define the following optimization problem for w_m:

$$\min_{\boldsymbol{w}|\boldsymbol{w}\geq\mathbf{0},\,\mathbf{1}^\top\boldsymbol{w}=1} \sum_{i\in\mathcal{G}} -\log\left\{\sum_{j|(i,j)\in\mathcal{P}} p(\boldsymbol{x}_j|\boldsymbol{x}_i)\right\} - \log\left\{1 - \sum_{j|(i,j)\in\mathcal{N}} p(\boldsymbol{x}_j|\boldsymbol{x}_i)\right\}, \tag{16}$$

$$\Rightarrow \min_{\boldsymbol{w}|\boldsymbol{w}\geq\mathbf{0},\,\mathbf{1}^\top\boldsymbol{w}=1} \Big[J(\boldsymbol{w}) \triangleq \sum_{i\in\mathcal{G}} -\log\{\boldsymbol{w}^\top\boldsymbol{\alpha}_{i\mathcal{P}}\} - \log\{1 - \boldsymbol{w}^\top\boldsymbol{\alpha}_{i\mathcal{N}}\}\Big], \tag{17}$$

$$\text{where } \boldsymbol{\alpha}_{i\mathcal{P}} = \big[\textstyle\sum_{j|(i,j)\in\mathcal{P}} \alpha_{ij}^{[1]}, \cdots, \sum_{j|(i,j)\in\mathcal{P}} \alpha_{ij}^{[M]}\big]^\top \in \mathbb{R}^M,$$

$$\boldsymbol{\alpha}_{i\mathcal{N}} = [\textstyle\sum_{j|(i,j)\in\mathcal{N}} \alpha_{ij}^{[1]}, \cdots, \sum_{j|(i,j)\in\mathcal{N}} \alpha_{ij}^{[M]}]^\top \in \mathbb{R}^M,$$

and we use the probabilistic constraint, $\sum_{j|(i,j)\in\mathcal{N}} p(\boldsymbol{x}_j|\boldsymbol{x}_i) \leq 1$. The derivative and Hessian of J are given by

$$\nabla J = \sum_{i\in\mathcal{G}} -\frac{\boldsymbol{\alpha}_{i\mathcal{P}}}{\boldsymbol{w}^\top\boldsymbol{\alpha}_{i\mathcal{P}}} + \frac{\boldsymbol{\alpha}_{i\mathcal{N}}}{1 - \boldsymbol{w}^\top\boldsymbol{\alpha}_{i\mathcal{N}}}, \quad \nabla^2 J = \sum_{i\in\mathcal{G}} \frac{\boldsymbol{\alpha}_{i\mathcal{P}}\boldsymbol{\alpha}_{i\mathcal{P}}^\top}{(\boldsymbol{w}^\top\boldsymbol{\alpha}_{i\mathcal{P}})^2} + \frac{\boldsymbol{\alpha}_{i\mathcal{N}}\boldsymbol{\alpha}_{i\mathcal{N}}^\top}{(1 - \boldsymbol{w}^\top\boldsymbol{\alpha}_{i\mathcal{N}})^2} \succcurlyeq 0,$$

which shows (17) is convex with the unique global optimum. We apply the reduced gradient descent method [19] to minimize J under the probabilistic constraint, $\boldsymbol{w} \geq 0$, $\mathbf{1}^\top\boldsymbol{w} = 1$. By using the optimized $\boldsymbol{w}$, the transition probability is obtained by (15) and the KTPS is finally obtained by (14) as multiple KTPS (MKTPS). In practice, we use $\log(\cdot + \epsilon)$, say $\epsilon = 1e^{-4}$, instead for $\log(\cdot)$ in (17) to avoid numerical instability.

The above proposed method is advantageous in terms of computation cost as compared to the standard MKL such as [19]. The size of training samples in (17) is $O(|\mathcal{G}|M)$ independent of the number of classes. By considering pairwise attributes, the class information is reduced into only the two categories $\mathcal{P}, \mathcal{N}$ which indicate the identity of pairwise samples. Then, for each labeled sample, such pairwise information is grouped into $\boldsymbol{\alpha}_{i\mathcal{P}}, \boldsymbol{\alpha}_{i\mathcal{N}}$ which suppresses the combinatorial increase of training sample vectors. Therefore, the computation cost for (17) depends only on the number of kernel functions M and that of labeled samples $|\mathcal{G}|$ even in multi-class problems. In addition, the proposed method does not contain any parameters to be set by users, such as cost parameter in SVM-based MKL [19].

4 Experimental Result

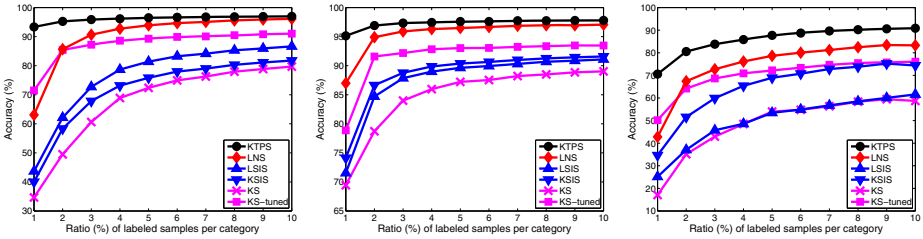
We conducted various experiments in the framework of semi-supervised learning using similarities. For comparison, we employed the other types of similarity; linear neighborhood similarity (LNS) [3], sparsity induced similarity (SIS) [2], and (Gaussian) kernel-based similarity (KS). For LNS, we utilized the coefficients obtained in linear neighborhood propagation [3] for similarities as in [2]. In [2], 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 kernel values as similarities, $s_{ij} = \mathsf{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$; especially, it corresponds to Gaussian kernel similarity (GKS) when the Gaussian kernel is used. 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, KSIS and KS, use the identical kernel for fair comparison. We implemented these methods on 3.33GHz PC by using MATLAB with MOSEK toolbox [14] for LNS and with L_1-magic toolbox [20] for LSIS/KSIS.

4.1 Label Propagation

First, we apply the similarity to label propagation [1] for estimating the labels by using a few labeled samples. We randomly drew labeled samples from the whole datasets and measured classification accuracy 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 on various datasets; USPS dataset [21], INRIA person dataset [22] and ETH80 object dataset [23].

USPS dataset [21]. We used 7,291 hand-written digits ($0 \sim 9$) images (16×16 pixels), resulting in a 10-class problem. The image vector whose dimensionality $D = 256$ is simply employed as the image feature. We employed the Gaussian kernel $\exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/h)$ of which bandwidth parameter h is determined as the mean of the pairwise distances denoted by γ. The number of neighbors k is set by $k = 1.5D = 384$, as in [2], and for KS-tuned, $k = 100$. The performance results are shown in Fig. 3a. The proposed KTPS significantly outperforms the others; in particular, the performance is over 90% even when only 1% samples are labeled.

Then, we show the robustness of the methods to the parameter settings; 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 $k \in \{0.5D, D, 1.5D, 2D, 4D\}$ by using 2% labeled samples. For comparison, the method of RMGT [4] is also applied, 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 1. 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



(a) USPS (b) INRIA (c) ETH-80

Fig. 3. Classification accuracy by label propagation

Table 1. Average accuracy (%) and its standard deviation for virous parameter values on USPS with 2% labeled samples

KTPS	LNS	LSIS	KSIS	KS	RMGT
95.4	86.1	62.8	60.7	50.4	91.9
±0.5	±2.6	±14.7	±19.3	±19.8	±6.0

Table 2. Computation time (msec) per sample for constructing similarities on USPS

KTPS SMO	KTPS MOSEK	LNS	LSIS	KSIS	RMGT
2.9	113.7	124.4	177.2	185.5	107.4

proposed KTPS is quite robust to such parameter settings, which is important to free us from exhaustively tuning the parameters, as discussed in Sec.2.2.

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 2, omitting the result of KS which requires only kernel computation and k-NN, and the result of KTPS using standard QP solver (`mskqpopt` in MOSEK) is shown as a reference. For RMGT, we report the averaged computation time on the above experimental setting (Table 1). The computation time of KTPS is significantly short compared to the others, demonstrating that the SMO approach (Sec.2.2) is quite effective in practice.

INRIA Person Dataset [22]. We used 3,548 person and 25,770 person-free images (64×128 pixels). The GLAC feature vectors [24] whose dimensionality is 6,480 (for 4×5 spatial bins) were extracted from the images with the same parameter settings as in [24]. Note that the person-free images contain various types of objects and their feature distribution is not so structured as in the 'person' category. The Gaussian kernel is employed in the same manner as in USPS dataset. The number of neighbors is set to $k = 500$, and for KS-tuned, $k = 100$. The performance results are shown in Fig. 3b, and we can see that the proposed KTPS is again superior to the others, especially for the small amount of labeled samples.

ETH-80 Object Dataset [23]. There are eight object categories, each of which contains 10 different objects captured with 41 different poses. 3,280 images

(256×256 pixels) are available in total. We took the bag-of-features approach [25] using SIFT local descriptors [26] extracted at 10 pixel-spaced grid points in the image. Then, we applied hierarchical k-means clustering with five layers and five branches to construct totally 3,905 hierarchical visual words (clusters). The image is represented by the 3,905-dimensional histogram features. We employed the pyramid match kernel [25] for the features and normalized the kernel in unit L2 norm. For constructing similarities, the number of neighbors is set to $k=500$, and for KS-tuned, $k=100$. The performance results are shown in Fig. 3c, demonstrating that the KTPS significantly outperforms the others for such type of kernel other than Gaussian kernel.

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.

4.2 Semi-supervised Discriminant Analysis

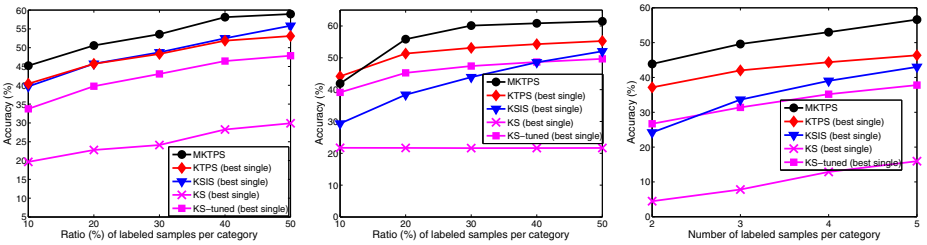
Next, we applied the similarity to semi-supervised discriminant analysis (SDA) [5]. The method of SDA is recently developed by extending (supervised) Fisher discriminant analysis so as to incorporate the unlabeled samples via the graph Laplacian [10] based on the similarity. As in DA, the SDA provides the projection vectors into the discriminant space, and in these experiments, the samples are classified by 1-NN method in the discriminant space. Note that the newly input samples which are out of the training samples can be easily classified by projecting them into that space unlike label propagation. We used ORL face dataset[2] and UMIST face dataset [27]. Each dataset is first split into a training set and a test set, and then the training set is further split into an unlabeled and labeled set which contains one image per category. We run on 50 random splits and report the averaged performances. The performances are evaluated in two ways; the classification accuracy on the training unlabeled set (transduction accuracy) and on the test set (induction accuracy).

ORL face dataset.[2] There are 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 discriminant space by SDA and the remaining samples are used for test set. All samples are used as neighbors ($k = 279$), while $k = 5$ for KS-tuned. The performance results are shown in Table 3a in which the performance by [9] measured in the same protocol is also presented as a reference. SDA using the proposed KTPS outperforms the other methods including [9] in terms of both transductive and inductive accuracies, while LSIS, KSIS and KS degrade the performances compared to DA which does not utilize any similarities.

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

Table 3. Classification accuracy (%) by SDA

	(a) ORL face dataset		(b) UMIST face dataset	
method	Transductive	Inductive	Transductive	Inductive
DA	67.6±3.0	67.5±4.1	44.1±3.6	46.3±3.6
Wang *et al.* [9]	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
LNS + SDA	76.1±3.2	65.8±4.3	51.9±3.3	50.2±3.3
KTPS + SDA	**81.8±3.0**	**76.9±3.8**	**69.1±4.8**	**69.0±5.3**



(a) Bird (b) Butterfly (c) Caltech101

Fig. 4. Classification accuracy by label propagation with MKTPS

UMIST face dataset [27]. The dataset consists of 575 images from 20 persons. While the original pre-cropped images are of size 112×92, 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 in the same manner as in USPS dataset. 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. The results are shown in Table 3b, demonstrating that the proposed KTPS is again superior to the others.

These experimental results demonstrate that the proposed KTPS is successfully incorporated into SDA. The KTPS can favorably boost the performances of the semi-supervised methods.

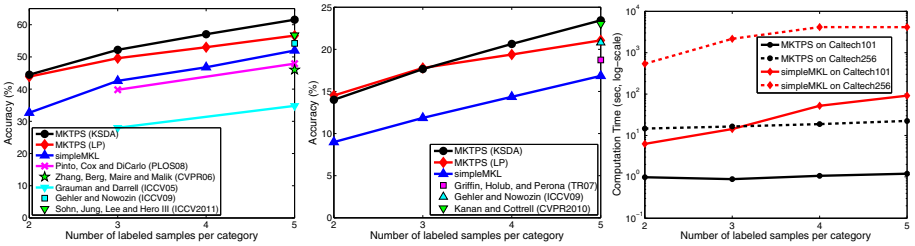
4.3 Multiple Kernel Integration

At the last, we applied the method to integrate multiple kernels for MKTPS (Sec.3).

We used Bird dataset [28], Butterfly dataset [29] and Caltech101 dataset [30].

Bird dataset [28]. The dataset 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 [29]. The dataset has 619 images of seven butterfly classes. All samples are used as neighbors ($k = 618$), while $k = 10$ for KS-tuned.



(a) Caltech101 (b) Caltech256 (c) Computation time

Fig. 5. Comparative performance results by MKTPS on Caltech datasets

For these datasets, we used three types of precomputed pairwise distances provided in the website[3] of the authors [31]; for details of the distances, refer to [31]. The multiple (three) types of kernels are constructed by applying Gaussian kernel to those precomputed distances in the same manner as in USPS dataset. The similarities of KTPS, KSIS and KS are constructed for each type of kernel, while MKTPS is obtained by integrating those multiple kernels. We drew labeled samples ranging from 10% to 50%, and repeated the trial 10 times. By applying label propagation, the remaining unlabeled samples are classified and the average classification accuracies are reported in Fig. 4a and Fig. 4b; we compare MKTPS to the best single similarity that produces the highest performance among the three types of kernels. The multiple kernels are favorably combined in MKTPS, improving the performance compared to the other best single similarities, even to the single KTPS.

Caltech101 dataset [30]. The dataset contains images in 102 object categories including 'background' category. We used ten types of precomputed kernels provided in the website[4] of the authors [32]; for details of the kernels, refer to [32]. The number of neighbors is set to $k = 500$, and for KS-tuned, $k = 10$. We selected 30 images per category (3,060 images in total) and drew labeled samples ranging from 2 to 5 samples per category in those images. The remaining unlabeled samples are classified by label propagation. The trial is repeated three times and the average classification accuracies are shown in Fig. 4c. Even on such a few labeled samples, the multiple kernels are effectively integrated to improve the performance by MKTPS; the performance gain increases along the number of labeled samples since the discriminative learning (Sec.3) is more effective for larger training samples.

We then applied the MKTPS to kernel SDA (KSDA) [5] with 1-NN as in Sec.4.2. The weights $\boldsymbol{w}$ obtained in MKTPS are also utilized to construct the composite kernel fed into KSDA. We additionally measured the performance on Caltech256 [33], which is a more challenging dataset containing images of 256 object categories, in the same protocol as Caltech101 by employing 39 types of kernels used in [34]. Fig. 5a and Fig. 5b show the performance results compared to those of above label propagation (LP), simpleMKL [19] (supervised method) and the prior works which have reported performances on such a small amount

[3] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/msorec/
[4] http://www.robots.ox.ac.uk/~vgg/software/MKL/ker-details.html

of labeled (training) samples. As the number of labeled samples increases, the KSDA with MKTPS effectively improves the performances over the LP with MKTPS. While the LP simply estimates the labels based only on similarity measures, the KSDA construct the subspace in a discriminative manner and such discriminative learning becomes effective for increased number of labeled samples. The performances of the KSDA with MKTPS are competitive with the simpleMKL [19] and the other prior works. The computation time for learning MKTPS is shown in Fig. 5c, compared to simpleMKL [19]. It is significantly faster than the simpleMKL [19], especially on larger amount of labeled samples.

These experimental results show that the MKTPS derived from multiple kernels is effective for the semi-supervised methods of both LP and SDA in terms of both classification performance and computation time.

5 Conclusion

We have proposed methods to construct similarities between samples from the probabilistic viewpoint. In the proposed method, the similarities are derived from the kernel-based transition probabilities through considering kernel least squares in the probabilistic framework. From a geometrical viewpoint, the proposed similarities are favorably sparse even without k-NN. We also presented the method to efficiently compute the similarity by using SMO. In addition, for the case of multiple kernel functions, we 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. In the experiments on semi-supervised learning problems using various datasets, the proposed methods exhibited the favorable performances compared to the other methods in terms of both classification accuracy and computation time, by applying label propagation as well as semi-supervised discriminant analysis. The similarities are fundamental in most of the semi-supervised learning methods, and thus the proposed similarity would benefit to the other semi-supervised methods, such as Laplacian SVM [7].

References

1. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)
2. Cheng, H., Liu, Z., Yang, J.: Sparsity induced similarity measure for label propagation. In: ICCV, pp. 317–324 (2009)
3. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. IEEE Transactions on Knowledge and Data Engineering 20, 55–67 (2008)
4. Liu, W., Chang, S.F.: Robust multi-class transductive learning with graphs. In: CVPR, pp. 381–388 (2009)
5. Cai, D., He, X., Han, J.: Semi-supervised discriminant analysis. In: ICCV (2007)
6. Zhang, Y., Yeung, D.Y.: Semi-supervised discriminant analysis using robust path-based similarity. In: CVPR (2008)

7. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7, 2399–2434 (2006)
8. Kobayashi, T., Watanabe, K., Otsu, N.: Logistic label propagation. Pattern Recognition Letters 33, 580–588 (2012)
9. Wang, F., Wang, X., Li, T.: Beyond the graphs: Semi-parametric semi-supervised discriminant analysis. In: CVPR, pp. 2113–2120 (2009)
10. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2003)
11. Otsu, N.: Optimal linear and nonlinear solutions for least-square discriminant feature extraction. In: ICPR (1982)
12. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research 5, 27–72 (2004)
13. Tang, J., Hua, X.-S., Song, Y., Qi, G.-J., Wu, X.: Kernel-Based Linear Neighborhood Propagation for Semantic Video Annotation. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 793–800. Springer, Heidelberg (2007)
14. The mosek optimization software, `http://www.mosek.com/`
15. Vapnik, V.: Statistical Learning Theory. Wiley (1998)
16. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
17. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2007)
18. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics 22, 79–86 (1951)
19. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: Simplemkl. Journal of Machine Learning Research 9, 2491–2521 (2008)
20. L1-magic, `http://users.ece.gatech.edu/~justin/l1magic`
21. Hull, J.: A database for handwritten text recognition research. IEEE Transaction on Pattern Analysis and Machine Intelligence 16, 550–554 (1994)
22. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
23. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: CVPR, pp. 409–415 (2003)
24. Kobayashi, T., Otsu, N.: Image Feature Extraction Using Gradient Local Auto-Correlations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 346–358. Springer, Heidelberg (2008)
25. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV, pp. 1458–1465 (2005)
26. Lowe, D.: Distinctive image features from scale invariant features. International Journal of Compuater Vision 60, 91–110 (2004)
27. Graham, D.B., Allinson, N.M.: Characterizing virtual eigensignatures for general purpose face recognition. In: Face Recognition: From Theory to Applications. NATO ASI Series F, Computer and Systems Sciences, vol. 163, pp. 446–456 (1998)
28. Lazebnik, S., Schmid, C., Ponce, J.: A maximum entropy framework for part-based texture and object recognition. In: ICCV, pp. 832–838 (2005)
29. Lazebnik, S., Schmid, C., Ponce, J.: Semi-local affine parts for object recognition. In: BMVC, pp. 779–788 (2004)

30. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transaction on Pattern Analysis and Machine Intelligence 28, 594–611 (2006)
31. Mario Christoudias, C., Urtasun, R., Salzmann, M., Darrell, T.: Learning to Recognize Objects from Unseen Modalities. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 677–691. Springer, Heidelberg (2010)
32. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV, pp. 606–613 (2009)
33. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, Caltech (2007)
34. Gehler, P., Nowozin, S.: Supplementary material for the paper: On feature combination for multiclass object classification. In: ICCV (2009)