Artif Intell Rev (2015) 43:179–192
DOI 10.1007/s10462-012-9369-4

An overview of kernel alignment and its applications

Tinghua Wang · Dongyan Zhao · Shengfeng Tian

Published online: 8 November 2012

Abstract The success of kernel methods is very much dependent on the choice of kernel. Kernel design and learning a kernel from the data require evaluation measures to assess the quality of the kernel. In recent years, the notion of kernel alignment, which measures the degree of agreement between a kernel and a learning task, is widely used for kernel selection due to its effectiveness and low computational complexity. In this paper, we present an overview of the research progress of kernel alignment and its applications. We introduce the basic idea of kernel alignment and its theoretical properties, as well as the extensions and improvements for specific learning problems. The typical applications, including kernel parameter tuning, multiple kernel learning, spectral kernel learning and feature selection and extraction, are reviewed in the context of classification framework. The relationship between kernel alignment and other evaluation measures is also explored. Finally, concluding remarks and future directions are presented.

Keywords Kernel alignment · Kernel evaluation measure · Learning kernels · Kernel method · Model selection

T. Wang · D. Zhao
Institute of Computer Science and Technology, Peking University,
Beijing 100871, China

D. Zhao
e-mail: zhaodongyan@pku.edu.cn

T. Wang (✉)
School of Mathematics and Computer Science, Gannan Normal University,
Ganzhou 341000, China
e-mail: wthpku@163.com

S. Tian
School of Computer and Information Technology, Beijing Jiaotong University,
Beijing 100044, China
e-mail: sftian@bjtu.edu.cn

1 Introduction

Kernel methods such as support vector machines (SVM) and kernel Fisher discriminant analysis (KFDA) (Amayri and Bouguila 2010; Shawe-Taylor and Cristianini 2004; Vapnik 1998) have delivered extremely high performance in a wide variety of learning tasks. Basically, kernel methods work by mapping the data from the input space into a high-dimensional (possibly infinite) feature space, which is usually chosen to be a reproducing kernel Hilbert space (RKHS), and then building linear algorithms in the feature space to implement nonlinear counterparts in the input space. The mapping, rather than being given in an explicit form, is determined implicitly by specifying a kernel function, which computes the inner product between each pair of data points in the feature space. Since the geometrical structure of the mapped data in the feature space is totally determined by the kernel function, choosing the appropriate kernel function, thereby, the appropriate feature space has a crucial effect on the performance of the kernel methods.

In the literature, kernel selection and optimization is often considered as a problem of model selection. Typically, a parameterized family of kernel functions is considered and the model selection reduces to a real-valued parameter optimization problem. In recent years, various evaluation measures of kernel function for model selection have been proposed. Commonly used measure is cross validation or leave-one-out error estimate. Because of the high computational complexity, this measure is only suitable for the adjustment of very few parameters. To overcome this difficulty, in the context of SVM, different approximations or upper bounds of the leave-one-out error in terms of analytical expressions have been presented (Chapelle et al. 2002; Chung et al. 2003; Duan et al. 2003; Wang et al. 2008; Liu et al. 2011). Of these various bounds, the radius-margin bound is commonly used in practice. Usually the most sophisticated algorithms for model selection based on radius-margin bound are gradient-based methods. However, these approaches, at each iteration, require the training of the learning machine and the solution of an additional quadratic program to compute the radius of the smallest ball enclosing the training data in the feature space. Other techniques like structural risk functional (Gönen and Alpayın 2011; Lanckriet et al. 2004; Ong and Williamson 2005) and negative log-posterior (Girolami and Rogers 2005) can also be considered as expected evaluation measures. These measures do not give a specific value, but only assert certain criteria in form of regularities in certain spaces, for example, RKHS or hyper-RKHS. Furthermore, these measures also require the whole learning process for evaluation.

In contrast, many efficient universal kernel evaluation measures have been derived recently, such as kernel alignment (Cortes et al. 2012; Cristianini et al. 2001), kernel polarization (Baram 2005; Wang et al. 2009), kernel class separability (Wang 2008; Xiong et al. 2005) and feature space-based kernel matrix evaluation measure (FSM) (Nguyen and Ho 2008). The significant property of evaluating these measures for model selection is that they make use of the information from the complete training data and can be computed efficiently. Furthermore, they are independent of the actual learning machine used. To the best of our knowledge, the most commonly used evaluation measure is kernel alignment due to its simplicity, efficiency and theoretical guarantee. There is significant amount of work in the literature for kernel alignment and its applications. In this paper, we hope to provide an extensive overview of the state of the art that helps researchers further address the problem of evaluating kernel quality for kernel learning and model selection.

The rest of this paper is organized as follows. The basic idea of kernel alignment and its theoretical properties are provided in Sect. 2. Section 3 concerns with the extensions and

improvements of kernel alignment for multiclass classification, unbalanced class distribution and regression. Section 4 reviews the typical applications, including kernel parameter tuning, multiple kernel learning (MKL), spectral kernel learning (SKL) and feature selection and extraction. The relationship between kernel alignment and other efficient kernel evaluation measures is discussed in Sect. 5, and the conclusion and future work are given in Sect. 6.

2 Kernel alignment

The notion of kernel alignment was first introduced by Cristianini et al. (2001). In a binary classification problem, we are given l pairs of training samples denoted by $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, where $\boldsymbol{x}_i \in X \subset R^n$ (the input space) and $y_i \in \{\pm 1\}$. Each $\boldsymbol{x}$ is then mapped to a $\phi(\boldsymbol{x})$ in the feature space F ($\phi : X \to F$), which is implicitly determined by the kernel function $k(\boldsymbol{x}, \boldsymbol{z}) = \phi(\boldsymbol{x})^{\mathrm{T}}\phi(\boldsymbol{z})$ for any $\boldsymbol{x}, \boldsymbol{z} \in X$. The kernel matrix (Gram matrix) $\boldsymbol{K}$ for kernel k is defined as $[\boldsymbol{K}]_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Given two kernels k_1 and k_2, the empirical alignment evaluates the similarity between the corresponding kernel matrices. Mathematically, it is defined as:

$$A(\boldsymbol{K}_1, \boldsymbol{K}_2) = \frac{\langle \boldsymbol{K}_1, \boldsymbol{K}_2 \rangle_{\mathrm{F}}}{\sqrt{\langle \boldsymbol{K}_1, \boldsymbol{K}_1 \rangle_{\mathrm{F}} \langle \boldsymbol{K}_2, \boldsymbol{K}_2 \rangle_{\mathrm{F}}}} \tag{1}$$

where $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ denote the kernel matrix (Gram matrix) for the kernel k_1 and k_2, respectively. $\langle \boldsymbol{K}_1, \boldsymbol{K}_2 \rangle_{\mathrm{F}}$ is the Frobenius inner product between two matrices, which is given by

$$\langle \boldsymbol{K}_1, \boldsymbol{K}_2 \rangle_{\mathrm{F}} = \sum_{i=1}^{l} \sum_{j=1}^{l} k_1(\boldsymbol{x}_i, \boldsymbol{x}_j) k_2(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{2}$$

If the kernel matrices $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ are considered as bidimensional vectors, the alignment can be seen as a similarity score based on the cosine of the angle. For arbitrary matrices, this score ranges between -1 and 1. However, since kernel alignment is only measured using positive semidefinite Gram matrices, the score is lower-bounded by 0 (Shawe-Taylor and Cristianini 2004).

For classification purposes we can define an ideal target matrix as $\boldsymbol{K}^* = \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}$, where $\boldsymbol{y} = (y_1, \ldots, y_l)^{\mathrm{T}}$ is the vector of labels for the training set D. Then the empirical alignment between the kernel matrix $\boldsymbol{K}$ and the target matrix $\boldsymbol{K}^*$ can be written as:

$$\begin{aligned} A(\boldsymbol{K}, \boldsymbol{K}^*) &= \frac{\langle \boldsymbol{K}, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}} \rangle_{\mathrm{F}}}{\sqrt{\langle \boldsymbol{K}, \boldsymbol{K} \rangle_{\mathrm{F}} \langle \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}} \rangle_{\mathrm{F}}}} \\ &= \frac{\langle \boldsymbol{K}, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}} \rangle_{\mathrm{F}}}{l\sqrt{\langle \boldsymbol{K}, \boldsymbol{K} \rangle_{\mathrm{F}}}} = \frac{\boldsymbol{y}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{y}}{l \, \|\boldsymbol{K}\|_{\mathrm{F}}} \end{aligned} \tag{3}$$

where $\|\boldsymbol{K}\|_{\mathrm{F}}$ is the Frobenius norm of the kernel matrix K.

It has been shown that kernel alignment possesses several convenient theoretical properties (Cristianini et al. 2001; Kandola et al. 2002a,b; Nguyen and Ho 2008):

(1) **Computational efficiency.** It uses only the training samples and can be efficiently computed in $O(l^2)$ time complexity prior to any computationally intensive training of the kernel machines. For example, compared to the radius-margin bound (Chapelle et al. 2002), which needs to solve two quadratic optimization problems, the evaluation of kernel alignment has much less computational load, since it does not involve any optimization.

(2) **Concentration.** Concentration means that the probability of an empirical estimate deviating from its mean can be bounded as an exponentially decaying function of that deviation. Kernel alignment is sharply concentrated around its expected value, and hence its empirical value is stable with respect to different splits of the data. This gives a high probability that an empirical estimate of kernel alignment is close to the true alignment value.
(3) **Generalization.** Generalization means the classification accuracy or error rate of a classifier on the test set. If the expected value of kernel alignment is high, then there exists a separation of the data with a low bound on the generalization error.

Furthermore, geometrically, kernel alignment maximization aims at finding a kernel k from a restricted family of "reasonable" kernel functions such that the normalized Gram matrix induced by k has the smallest distance to the normalized ideal target matrix $\boldsymbol{K}^*$. Formally, maximizing $A(\boldsymbol{K}, \boldsymbol{K}^*)$ is equivalent to minimizing the following distance (Igel et al. 2007):

$$d(\boldsymbol{K}, \boldsymbol{K}^*) = \left\| \frac{\boldsymbol{K}}{\|\boldsymbol{K}\|_{\mathrm{F}}} - \frac{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}}{\|\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\|_{\mathrm{F}}} \right\| = \sqrt{2 - 2A(\boldsymbol{K}, \boldsymbol{K}^*)} \tag{4}$$

All these observations together suggest that we can evaluate and optimize the kernel alignment on a training set, and expect to keep high alignment and hence achieve a better generalization performance on the test set.

3 Extensions and improvements

Kernel alignment was originally presented for binary classification problems. However, it can be extended to some other learning cases, such as multiclass classification (Guermeur et al. 2004; Vert 2002), unbalanced class distribution (Cortes et al. 2012; Kandola et al. 2002a; Kawanabe et al. 2009) and regression (Kandola et al. 2002a).

3.1 Multiclass classification

In kernel methods, multiclass SVM has recently attracted much attention due to the demands for multiclass classification in many practical applications and the success of SVM in binary classification. There are several approaches available to extend binary classification to multiclass case (Hsu and Lin 2002; Rifkin and Klautau 2004; Tsochantaridis et al. 2005). These approaches roughly fall into two categories. The first approach denoted as all-in-one or single machine is to directly consider all data in one optimization formulation. The second one involves considering a decomposition of the multiclass problem into several binary subproblems and then combining their solutions. The well-known one-versus-rest (1-v-r) and one-versus-one (1-v-1) strategies are the two most common ways for building a multiclass decision function based on pairwise decision functions. Although both approaches present usually no significant difference in classification accuracy when the parameters of SVM are properly tuned, the decomposition one is often recommended for practical use because of lower computational overhead and conceptual simplicity (Hsu and Lin 2002; Rifkin and Klautau 2004).

Similar to binary classification, we need to build a new kernel function that takes into account the information of all classes simultaneously, i.e. multiclass case. Vert (2002) proposed to build the ideal target matrix as follows:

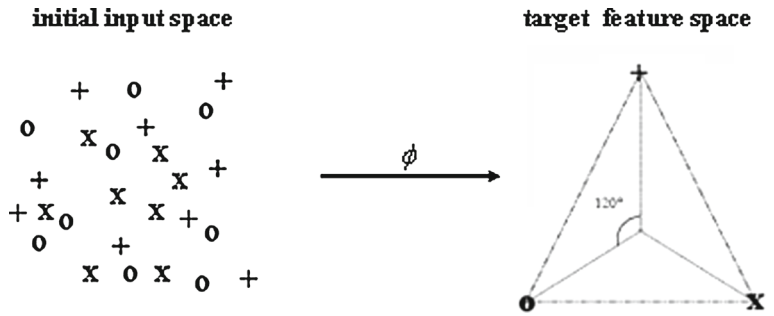

Fig. 1 3-Class problem in 2D: the target matrix performs clustering of the data in the feature space

$$[\boldsymbol{K}^*]_{i,j} = \begin{cases} 1 & y_i = y_j \\ \frac{-1}{m-1} & y_i \neq y_j \end{cases} \tag{5}$$

where $m (m \geq 2)$ is the number of classes. This target matrix corresponds to a mapping ϕ that associates each data point $\boldsymbol{x}$ with one of the m vertices of a $(m-1)$-dimensional centered simplex, according to the class label to which it belongs. Figure 1 shows that the target matrix performs clustering of the data in the feature space for an example of 3-class problem (Guermeur et al. 2004).

3.2 Unbalanced class distribution

The problem of unbalanced class distribution, i.e. data sets are always with an unequal number of class labels, is commonplace in data mining and machine learning community. A drawback of kernel alignment is that it doesn't consider the unbalanced class distribution which may cause the sensitivity of the measure to drop drastically. To deal with this issue, Kandola et al. (2002a) proposed to by adjusting the target matrix $\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}$ using the following transformation:

$$y_i = \begin{cases} \frac{1}{l_+} & y_i = +1 \\ -\frac{1}{l_-} & y_i = -1 \end{cases} \tag{6}$$

where l_+ and l_- denote the number of positive and negative samples in the data set, respectively. This gives a slightly modified definition of kernel alignment.

A more sophisticated and general way to cancel the effect of unbalanced class distribution is to center the kernels or Gram matrices before computing the alignment measure (Cortes et al. 2012; Kawanabe et al. 2009). The centered kernel k_c associated to k is defined for any $\boldsymbol{x}, \boldsymbol{z} \in X$ by

$$k_c(\boldsymbol{x}, \boldsymbol{z}) = (\phi(\boldsymbol{x}) - \mathrm{E}[\phi(X)])^{\mathrm{T}}(\phi(\boldsymbol{z}) - \mathrm{E}[\phi(X)]) \tag{7}$$

where $\mathrm{E}[\phi(X)]$ denotes the expected value of $\phi(\boldsymbol{x})$ for all $\boldsymbol{x} \in X$. Mathematically, centering in the corresponding feature space is achieved by multiplying the matrix $\boldsymbol{C} = \boldsymbol{I} - \mathbf{1}\mathbf{1}^{\mathrm{T}}/l$ to Gram matrix $\boldsymbol{K}$ from both sides, i.e. $\boldsymbol{CKC}$, where $\boldsymbol{I}$ is the identity matrix of size l and $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}} \in R^l$. Let $\boldsymbol{K}_c = \boldsymbol{CKC}$ and $\boldsymbol{K}_c^* = \boldsymbol{CK}^*\boldsymbol{C}$, the empirical centered alignment between the kernel matrix $\boldsymbol{K}$ and the target matrix $\boldsymbol{K}^*$ is defined as:

$$A_c(\boldsymbol{K}, \boldsymbol{K}^*) = \frac{\langle \boldsymbol{K}_c, \boldsymbol{K}_c^* \rangle_{\mathrm{F}}}{\|\boldsymbol{K}_c\|_{\mathrm{F}} \left\|\boldsymbol{K}_c^*\right\|_{\mathrm{F}}} \tag{8}$$

Although this improved definition of alignment may appear to be a technicality, it is in fact a critical difference. Without that centering, the definition of alignment does not correlate well with the performance of learning machine. This phenomenon was also noticed by Meila (2003) and Pothin and Richard (2008). For further discussion of the virtues of centered alignment, see the paper by Cortes et al. (2012).

3.3 Regression

The problem of regression is to approximate an unknown real-value function from the observation of a limited sequence of (typically) noise corrupted input/output data pairs. The first algorithm based on kernel alignment is to adapt the kernel matrix to improve the alignment with a target classification (Cristianini et al. 2001). This algorithm performs transduction, and provides a nonparametric way to perform kernel selection. To apply this algorithm for the case of regression, Kandola et al. (2002a) proposed to modify the target matrix $\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}$ using the following transformation:

$$y_i \leftarrow y_i - \bar{y} \tag{9}$$

where $\bar{y}$ represents the mean of the output values over the training set. Note that for regression $y_i \in R$ denotes the output corresponding to the input $\boldsymbol{x}_i \in R^n$.

4 Applications

In kernel methods, the goal of model selection is to ensure a good generalization performance of the learning machine, that is, a minimal risk. The ease with which kernel alignment can be calculated in $O(l^2)$ time complexity using only training data, prior to any computationally intensive training of learning machine, makes it an interesting evaluation measure for model selection. Generally speaking, the existing applications of kernel alignment can be categorized into four groups: kernel parameter tuning, MKL, SKL and feature selection and extraction.

4.1 Kernel parameter tuning

The kernel parameters always have a significant influence on the overall performance of the final obtained learning mode. For example, Keerthi and Lin (2003) analyzed the asymptotic behaviors of SVM with the Gaussian kernel and pointed out that severe overfitting or underfitting would occur when the width parameter σ was not properly tuned.[1]

Given a kernel k_θ depending on a parameter set $\boldsymbol{\theta}$, since the kernel alignment is differentiable with respect to $\boldsymbol{\theta}$ as long as the kernel is, gradient-based optimization algorithms are usually used to choose the parameter set $\boldsymbol{\theta}$ by maximizing the kernel alignment (Camargo and González 2009; Guermeur et al. 2004; Igel et al. 2007; Pothin and Richard 2006, 2007). Formally, it searches for:

$$\boldsymbol{\theta}^* = \arg\max_{\theta} \quad A(\boldsymbol{K}_\theta, \boldsymbol{K}^*) \tag{10}$$

[1] More precisely, the generalization performance of the SVM with a Gaussian kernel is jointly determined by two parameters: the kernel parameter σ and regularization parameter C.

where $\boldsymbol{K}_\theta$ denotes the Gram matrix for kernel k_θ. The gradients with respect to the kernel parameters can be easily calculated. For example, from Eq. (3) we have

$$\begin{aligned}\nabla_\theta A(\boldsymbol{K}_\theta, \boldsymbol{K}^*) &= \frac{\partial A(\boldsymbol{K}_\theta, \boldsymbol{K}^*)}{\partial \boldsymbol{\theta}} = \frac{\boldsymbol{y}^{\mathrm{T}}(\partial \boldsymbol{K}_\theta / \partial \boldsymbol{\theta})\boldsymbol{y}}{l \left\| \boldsymbol{K}_\theta \right\|_{\mathrm{F}}} - \frac{\boldsymbol{y}^{\mathrm{T}} K_\theta \boldsymbol{y} \left[\partial \left(\| K_\theta \|_{\mathrm{F}} \right) / \partial \boldsymbol{\theta}\right]}{l \left\| K_\theta \right\|_{\mathrm{F}}^2} \\ &= \frac{\left\langle \frac{\partial \boldsymbol{K}_\theta}{\partial \boldsymbol{\theta}}, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}} \right\rangle_{\mathrm{F}} \langle K_\theta, \boldsymbol{K}_\theta \rangle_{\mathrm{F}} - \left\langle K_\theta, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}} \right\rangle_{\mathrm{F}} \left\langle \frac{\partial K_\theta}{\partial \boldsymbol{\theta}}, \boldsymbol{K}_\theta \right\rangle_{\mathrm{F}}}{l \langle K_\theta, \boldsymbol{K}_\theta \rangle_{\mathrm{F}}^{3/2}}\end{aligned} \tag{11}$$

Finally, tuning the kernel parameters can be achieved by the following gradient update rule:

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \eta \nabla_\theta A(\boldsymbol{K}_\theta, \boldsymbol{K}^*) \tag{12}$$

where t is the iteration step number and $\eta > 0$ is the gradient step.

4.2 Multiple kernel learning

In recent years, MKL algorithms have been proposed (Gönen and Alpayın 2011; Kandola et al. 2002b; Lanckriet et al. 2004), for learning a combination k_w of multiple base kernels instead of selecting one:

$$k_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = f_{\boldsymbol{w}} \left(\{k_r(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{r=1}^p \right) \tag{13}$$

where the combination function ($f_{\boldsymbol{w}} : \boldsymbol{R}^P \rightarrow \boldsymbol{R}$) forms a single kernel from the p base kernels using the parameters $\boldsymbol{w}$. Specially, when focusing on learning finite linear combinations of the given base kernels, we can linearly parameterize the combination function as:

$$k_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = f_{\boldsymbol{w}} \left(\{k_r(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{r=1}^p \right) = \sum_{r=1}^{p} w_r k_r(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{14}$$

where $\boldsymbol{w} = (w_1, \ldots, w_p)^{\mathrm{T}}$ is the kernel weight vector. Compared with traditional kernel methods using a single fixed kernel, MKL does exhibit its flexibility of automated kernel learning, and also reflect the fact that typical learning problems often involve multiple, heterogeneous data sources.

How to determine the kernel weights $\boldsymbol{w}$ is the main task of MKL. Different approaches based on kernel alignment differ in the way they put restrictions on w: the linear combination ($\boldsymbol{w} \in R^p$), the conic combination ($\boldsymbol{w} \in R_+^p$) or the convex combination ($\boldsymbol{w} \in R_+^p$ and $\sum_{r=1}^p w_r = 1$).[2] For example: Lanckriet et al. (2004) proposed to maximize the kernel alignment using arbitrary kernel weights, which can be cast as a semidefinite programming (SDP) problem. They also restricted the kernel weights to be nonnegative and their SDP formulation reduced to a quadratically constrained quadratic programming (QCQP) problem. Cortes et al. (2012) proposed to maximize the centered alignment with respect to $\|\boldsymbol{w}\| = 1$, which has an analytical solution. They also restricted the kernel weights to be nonnegative and obtained a QP problem. Kandola et al. (2002b) proposed to maximize the kernel alignment with respect to a conic combination of kernels. The resulting quadratic programming (QP) is very similar to the hard margin SVM optimization problem and is expected to give sparse kernel combination weights. Qiu and Lane (2009) proposed a simple heuristic to select the kernel weights using kernel alignment:

[2] Note that the conic combination is a special case of the linear combination and the convex combination is a special case of the conic combination.

$$w_r = \frac{A(\boldsymbol{K}_r, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}})}{\sum_{r=1}^{p} A(\boldsymbol{K}_r, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}})} \tag{15}$$

where $\boldsymbol{K}_r$ denotes the base kernel matrix for the base kernel k_r. Obviously, the obtained combined kernel is a convex combination of the base kernels.

4.3 Spectral kernel learning

In practice, a good learning model should take advantages on not only the labeled data, but also the unlabeled data when they are available. One popular way to exploit the unlabeled data is semi-supervised learning (Chapelle et al. 2006). A promising family of semi-supervised learning methods can be viewed as constructing kernels by transforming the spectra of some positive semidefinite matrices, such as kernel matrix $\boldsymbol{K}$ or graph Laplacian L both for labeled and unlabeled data, which is known as spectral kernel learning (SKL) (Hoi et al. 2006; Zhu et al. 2004). Formally, the framework of SKL suggests designing the new kernel matrix $\bar{\boldsymbol{K}}$ by a non-negative function g as follows:

$$\bar{\boldsymbol{K}} = \sum_{i=1}^{l} g(\lambda_i)\boldsymbol{v}_i\boldsymbol{v}_i^T \tag{16}$$

where $(\lambda_i, \boldsymbol{v}_i)$ are the eigen-pairs of that positive semidefinite matrix, and the function g can be regarded as a filter function or a transformation function that modifies the spectrum of the matrix. The resulting kernel $\bar{\boldsymbol{K}}$ is also referred as semi-supervised kernel.

Let $\mu_i = g(\lambda_i)$, the goal of SKL algorithm is to find the optimal spectral coefficients μ_i which should result in better generalization performance. Cristianini et al. (2001) proposed to maximize the kernel alignment with the available labels to find the optimal μ_i of the kernel matrix $\boldsymbol{K}$ and obtained an analytical solution. They also suggested a transduction algorithm based on the observation that optimizing kernel alignment with the labeled data and in doing so it will adapt the Gram matrix also for the unlabeled data. Zhu et al. (2004) proposed a semi-supervised kernel learning method, which learns the spectral transformation of the graph Laplacian L by optimizing kernel alignment with the so-called order constraints on $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_l)^{\mathrm{T}}$, namely $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_l$. Hoi et al. (2006) proposed a new SKL algorithm which is also built on the principle of kernel alignment:

$$\begin{aligned} \max\ & A(\bar{\boldsymbol{K}}_{tr}, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}) \\ \text{s.t.}\ & \bar{\boldsymbol{K}} = \sum_{i=1}^{l} \mu_i \boldsymbol{v}_i \boldsymbol{v}_i^T \\ & \mathrm{trace}(\bar{\boldsymbol{K}}) = 1 \\ & \mu_i \geq \beta\mu_{i+1} \geq 0, \quad i = 1, \ldots, l \end{aligned} \tag{17}$$

where β is introduced as a decay factor that satisfiers $\beta \geq 1$, $\boldsymbol{v}_i$ are the eigenvectors of the original kernel matrix $\boldsymbol{K}$, $\bar{\boldsymbol{K}}_{tr}$ is the kernel matrix restricted to the labeled training data. This optimization problem belongs to convex optimization and is usually regarded as a SDP (Lanckriet et al. 2004), which may not be computationally efficient. However, it can be transformed to a QP problem and solved much more efficiently.

4.4 Feature selection and extraction

Feature selection and feature extraction are two major approaches to dimensionality reduction (Ding et al. 2012; Jain et al. 2000). Feature selection refers to selecting features in the

measurement space and the features obtained are a subset of the original input variables, while feature extraction involves a transformation of the original input variables and features provided are a set of new variables in the transformed space. Kernel alignment is successfully applied to feature selection and feature extraction.

For feature selection, Wong and Burkowski (2009) deployed kernel alignment as an evaluation tool, using recursive feature elimination (RFE) to compute a molecular descriptor containing the most important features needed for a classification application. In detail, this algorithm starts with the full set of features, then recursively removes the feature that produces the maximum difference between the kernel alignment value evaluated with the reduced feature set and the kernel alignment value evaluated with the current feature set, that is, at each iteration, it removes the feature q that maximizes the difference:

$$diff(q) = A(\boldsymbol{K}_{removed(q)}, \boldsymbol{K}^*) - A(\boldsymbol{K}_{current}, \boldsymbol{K}^*) \tag{18}$$

It has been shown theoretically and empirically that this algorithm lowers the generalization error bound of an SVM classifier. Ramona et al. (2012) proposed a feature selection algorithm named scaled alignment selection (SAS), which performs an iterative maximization of the kernel alignment through a simple gradient ascent on the scaling factors $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)^{\mathrm{T}}$ of the scaled kernel:

$$k_\sigma(\boldsymbol{x}, \boldsymbol{z}) = k(\boldsymbol{\sigma} \circ \boldsymbol{x}, \boldsymbol{\sigma} \circ \boldsymbol{z}) \tag{19}$$

where $\boldsymbol{M} \circ \boldsymbol{N}$ denotes the entry-wise product of two vectors or matrices. The features are ranked after maximization by descending scale factor order, assuming that the most weighted features contribute the most to the decision function. In this approach, feature selection is essentially converted to a kernel parameter tuning problem.

For feature extraction, Wu and Farquhar (2007) proposed a discriminative subspace kernel learning algorithm to find a low-dimensional subspace of the kernel feature space. In this algorithm, kernel alignment was employed as the learning criterion, resulting in a nonlinear optimization problem, for which the conjugate gradient technique was applied to compute a locally optimal solution. Note that this nonlinear feature extraction is also conducted in the kernel parameter tuning framework. The key observation is that when projecting data into a low dimensional subspace of the feature space, the parameters that are used for describing this subspace can be regarded as the parameters of the kernel function between the projected data.

5 Relationship with other kernel evaluation measures

Kernel alignment evaluates how well the kernel matrix K aligns to an ideal target matrix $\boldsymbol{K}^*$. The higher the kernel alignment value the more aligned both matrices, hence the kernel provides high class separability and good classification results. This measure also has some similarities with other related measures.

5.1 Kernel polarization

Kernel polarization (Baram 2005; Wang et al. 2009) is defined as the Frobenius inner product between two matrices, which simplifies the kernel alignment by ridding its denominator:

$$KP(\boldsymbol{K}, \boldsymbol{K}^*) = \langle \boldsymbol{K}, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}} \rangle_{\mathrm{F}} = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{y} = \sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{20}$$

It can be written as:

$$KP(\boldsymbol{K}, \boldsymbol{K}^*) = \sum_{y_i = y_j} k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{y_i \neq y_j} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{21}$$

Obviously, $KP(\boldsymbol{K}, \boldsymbol{K}^*)$ will increase if the similarity represented by the kernel is large for the input points of the same class and small for the points from different classes.

We argue here that if a normalized kernel is used, kernel polarization is a lower bound of the kernel alignment multiplied by l^2. Indeed, a normalized kernel is given by:

$$k_{norm}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{k(\boldsymbol{x}_i, \boldsymbol{x}_i) k(\boldsymbol{x}_j, \boldsymbol{x}_j)}} \tag{22}$$

It is obvious that $-1 \leq k_{\text{norm}}(\boldsymbol{x}_i, \boldsymbol{x}_j) \leq 1$, hence we have

$$l^2 A(\boldsymbol{K}_{norm}, \boldsymbol{K}^*) = \frac{l \boldsymbol{y}^{\mathrm{T}} \boldsymbol{K}_{norm} \boldsymbol{y}}{\|\boldsymbol{K}_{norm}\|_{\mathrm{F}}} = \frac{l \boldsymbol{y}^{\mathrm{T}} \boldsymbol{K}_{norm} \boldsymbol{y}}{\sqrt{\sum_{i=1}^{l} \sum_{j=1}^{l} k_{norm}(\boldsymbol{x}_i, \boldsymbol{x}_j)^2}} \geq \boldsymbol{y}^{\mathrm{T}} \boldsymbol{K}_{norm} \boldsymbol{y} = KP(\boldsymbol{K}_{norm}, \boldsymbol{K}^*) \tag{23}$$

where $\boldsymbol{K}_{norm}$ denotes the kernel matrix for the normalized kernel $k_{\text{norm}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

5.2 Kernel class separability

Kernel class separability (Wang 2008; Xiong et al. 2005) takes the following expression:

$$J = \frac{\text{trace}\,(\boldsymbol{S}_B)}{\text{trace}\,(\boldsymbol{S}_W)} \tag{24}$$

where $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ denote the between-class scatter matrix and within-class scatter matrix in the feature space, respectively:

$$\begin{aligned} \boldsymbol{S}_B &= \sum_{i=1}^{m} l_i (\boldsymbol{a}_i - \boldsymbol{a})(\boldsymbol{a}_i - \boldsymbol{a})^{\mathrm{T}} \\ \boldsymbol{S}_W &= \sum_{i=1}^{m} \left(\sum_{j:y_j = i} (\phi(\boldsymbol{x}_j) - \boldsymbol{a}_i)(\phi(\boldsymbol{x}_j) - \boldsymbol{a}_i)^{\mathrm{T}} \right) \end{aligned} \tag{25}$$

where m, l_i, $\boldsymbol{a}_i$ and $\boldsymbol{a}$ denote the number of classes, the number of samples in the ith class, the mean vector for the ith class and mean vector for all classes in the feature space, respectively. Obviously, a high J means small within-class scatter and large between-class scatter in the feature space.

Kernel alignment is somehow close to the kernel class separability. Indeed, suppose that the number of classes is two ($m = 2$) and the numbers of training samples from each class are the same, it has been proven that (Wang 2008; Lee and Bien 2010):

$$A(\boldsymbol{K}, \boldsymbol{K}^*) = \frac{\text{trace}\,(\boldsymbol{S}_B)}{\|\boldsymbol{K}\|_{\mathrm{F}}} \tag{26}$$

Note that this equation still holds with the matrix $\boldsymbol{K}^*$ defined by Eq. (6) in the case of unbalanced class distribution (Ramona et al. 2012). Hence kernel alignment is a special case of the kernel class separability as shown in the following relationship:

$$J = \left(\frac{\|\boldsymbol{K}\|_{\mathrm{F}}}{\text{trace}\,(\boldsymbol{S}_W)} \right) A(\boldsymbol{K}, \boldsymbol{K}^*) \tag{27}$$

5.3 Feature space-based kernel matrix evaluation measure

Nguyen and Ho (2008) showed that having a high kernel alignment value is only a sufficient condition to be a good kernel matrix, but not a necessary condition. It is possible for a kernel matrix to have a very good performance even though its kernel alignment value is low. They then proposed a surrogate measure named feature space-based kernel matrix evaluation measure (FSM), which is defined as the ratio of the total within-class standard deviation in the direction between the class centers to the distance between these centers:

$$FSM(\boldsymbol{K},\boldsymbol{y}) = \frac{std_+ + std_-}{\|\boldsymbol{a}_+ - \boldsymbol{a}_-\|} \tag{28}$$

where $\boldsymbol{a}_+$ and $\boldsymbol{a}_-$ are class centers in the feature space, std_+ and std_- are standard derivation of the positive and negative class in the direction between the class centers, respectively. FSM has been shown to be advantageous over kernel alignment because it takes into account the within-class data variance (namely standard derivation) at a finer scale, and relaxes the strict conditions of kernel alignment by considering relative positions of classes in the feature space.

Furthermore, if the label vector $\boldsymbol{y}$ is adjusted by Eq. (6), it is easy to show that (Neumann et al. 2005; Ramona et al. 2012):

$$\langle \boldsymbol{K}, \boldsymbol{y}\boldsymbol{y}^{\mathrm{T}} \rangle_{\mathrm{F}} = \|\boldsymbol{a}_+ - \boldsymbol{a}_-\| = \frac{l}{l_+ l_-} \mathrm{trace}\,(\boldsymbol{S}_B) \tag{29}$$

Hence the reciprocal of FSM can be represented as:[3]

$$R\_FSM(\boldsymbol{K},\boldsymbol{y}) = \frac{1}{FSM(\boldsymbol{K},\boldsymbol{y})} = \frac{l}{l_+ l_-}\left(\frac{\mathrm{trace}\,(\boldsymbol{S}_B)}{std}\right) \tag{30}$$

where std is the total within-class standard derivation, i.e. $std = std_+ + std_-$. This shows a close relationship between R_FSM and kernel alignment. Compared to kernel class separability, the main advantage of R_FSM is that it only takes the total within-class data variance in the direction between the class centers into account based on the observation that varying data (in the feature space) along the separating hyperplane will not affect the margin (Nguyen and Ho 2008).

6 Conclusion and future work

The main advantage of kernel methods stems from the implicit transformation of data to a high-dimensional feature space with a kernel function, thus the choice of a kernel is of crucial importance. Learning a kernel from the data and model selection require evaluation measures to assess the quality of the kernel. In this paper, the research progress and applications of the kernel alignment, known as the most extensively used kernel evaluation measure in practice, is systematically reviewed. The relationship of this measure to kernel polarization, kernel class separability, and feature space-based kernel matrix evaluation measure (FSM) is further analyzed. This helps us in understanding the advantages and disadvantages of the kernel alignment for kernel learning and model selection.

The following issues are worthy of exploring in the future work:

[3] For a classification task, a good kernel matrix should be characterized with small FSM value, which is opposite to the other mentioned measures, i.e. kernel alignment, kernel polarization and kernel class separability, hence we consider the reciprocal of FSM for the sake of consistency.

(1) Although both theoretical and empirical results do suggest the existence of accurate predictors with a high kernel alignment value, we are not provided that a high kernel alignment value is necessarily needed for a good classifier. Moreover, the work presented in (Chudzian 2012) shows this measure doesn't result in the kernel parameters corresponding to the classifiers that achieve the minimal error rate. It will be important to thoroughly explore these imperfect results.

(2) It is ideally expected that the data is linearly separable in the feature space, thus the kernel alignment can be used as an evaluation measure for kernel learning and model selection. However, the data may not be linearly separable even after kernel transformation in many applications, e.g., the data may exist a multimodally distributed structure. In this case, by a similar analyzing way presented in (Chen et al. 2008; Wang et al. 2009), we can see that employing the kernel alignment as an evaluation tool may result in undesired results. This phenomenon implies that we should take into account finer data distribution models in the feature space to improve the current work.

(3) Most of the existing kernel evaluation measures follow the conventional wisdom of measuring within-class and between-class spreading (specifically, standard derivations) in an appropriate way. Motivated by this, we may design a unified kernel evaluation measure in a more generalized sense. For example, from Sect. 5 we can find that, although the denominators of the four mentioned measures are different, the numerator is common. From the point of view of the normalization, kernel polarization has no normalization, kernel class separability normalization aims at minimizing the within-class scatter in all directions, R_FSM normalization aims at minimizing the within-class scatter in the direction between the class centers, while kernel alignment normalization results in minimizing the global scatter, regardless of the class labels.

As a final remark, it should be emphasized that, in general, having a good kernel evaluation measure, we can leverage the work of kernel learning and model selection, which is certainly of the central interest in kernel methods.

Acknowledgments We would like to thank all the referees for their constructive and insightful comments on this paper. This work is supported in part by the National High Technology Research & Development Program of China (863 Program) (No. 2012AA011101), the National Natural Science Foundation of China (No. 61202265), the China Postdoctoral Science Foundation Funded Project (No. 2012M510275) and the Natural Science Foundation of Jiangxi Province of China (No. 20114BAB211021).

References

Amayri O, Bouguila N (2010) A study of spam filtering using support vector machines. Artif Intell Revi 34(1):73–108

Baram Y (2005) Learning by kernel polarization. Neural Comput 17(6):1264–1275

Camargo JE, González FA (2009) A multi-class kernel alignment method for image collection summarization. In: Proceedings of the 14th Iberoamerican conference on pattern recognition: progress in pattern recognition, image analysis, computer vision, and applications, Guadalajara, Mexico, pp 545–552

Chapelle O, Vapnik V, Mukherjee S (2002) Choosing multiple parameters for support vector machines. Mach Learn 46(1):131–159

Chapelle O, Zien A, Schölkopf B (2006) Semi-supervised learning. MIT Press, Cambridge, MA

Chen B, Liu H, Bao Z (2008) A kernel optimization method based on the localized kernel Fisher criterion. Pattern Recognit 41(3):1098–1109

Chudzian P (2012) Evaluation measures for kernel optimization. Pattern Recognit Lett 33(9):1108–1116

Chung KM, Kao WC, Sun T, Wang LL, Lin CJ (2003) Radius margin bounds for support vector machines with the RBF kernel. Neural Comput 15(11):2463–2681

Cortes C, Mohri M, Rostamizadeh A (2012) Algorithms for learning kernels based on centered alignment. J Mach Learn Res 13:795–828

Cristianini N, Shawe-Taylor J, Elisseeff A, Kandola J (2001) On kernel-target alignment. In: Dietterich TG, Becker S, Ghahraman Z (eds) Advances in Neural Information Processing Systems 14, MIT Press, Cambridge, MA, pp 367–373

Ding S, Zhu H, Jia W, Su C (2012) A survey on feature extraction for pattern recognition. Artif Intell Rev 37(3):169–180

Duan KB, Keerthi SS, Poo AN (2003) Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing 51:41–59

Girolami M, Rogers S (2005) Hierarchic Bayesian models for kernel learning. In: Proceedings of the 22nd international conference on machine learning, Bonn, Germany, pp 241–248

Gönen M, Alpayın E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268

Guermeur Y, Lifchitz A, Vert R (2004) A kernel for protein secondary structure prediction. In: Schölkopf B, Tsuda K, Vert JP (eds) Kernel Methods in Computational Biology, MIT Press, Cambridge, MA, pp 193–206

Hoi SCH, Lyu MR, Chang EY (2006) Learning the unified kernel machines for classification. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, Philadelphia, USA, pp 187–196

Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425

Igel C, Glasmachers T, Mersch B, Pfeifer N, Meinicke P (2007) Gradient-based optimization of kernel-target alignment for sequence kernels applied to bacterial gene start detection. IEEE/ACM Trans Comput Biol Bioinform 4(2):216–226

Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37

Kandola J, Shawe-Taylor J, Cristianini N (2002a) On the extensions of kernel alignment. Technical report 120, Department of Computer Science, University of London

Kandola J, Shawe-Taylor J, Cristianini N (2002b) Optimizing kernel alignment over combinations of kernels. Technical report 121, Department of Computer Science, University of London

Kawanabe M, Nakajima S, Binder A (2009) A procedure of adaptive kernel combination with kernel-target alignment for object classification. In: Proceedings of the 8th ACM international conference on image and video retrieval, Santorini Island, Greece

Keerthi SS, Lin CJ (2003) Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Comput 15(7):1667–1689

Lanckriet GRG, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI (2004) Learning the kernel matrix with semidefinite programming. J Mach Learn Res 5:27–72

Lee SW, Bien Z (2010) Representation of Fisher criterion function in a kernel feature space. IEEE Trans Neural Netw 21(2):333–339

Liu Y, Liao S, Hou Y (2011) Learning kernels with upper bounds of leave-one-out error. In: Proceedings of the 20th ACM conference on information and knowledge management, Glasgow, UK, pp 2205–2208

Meila M (2003) Data centering in feature space. In: Proceedings of the 9th international workshop on artificial intelligence and statistics, Key West, USA

Neumann J, Schnörr C, Steidl G (2005) Combined SVM-based feature selection and classification. Mach Learn 61(1–3):129–150

Nguyen CH, Ho TB (2008) An efficient kernel matrix evaluation measure. Pattern Recognit 41(11):3366–3372

Ong CS, Williamson RC (2005) Learning the kernel with hyperkernels. J Mach Learn Res 6:1043–1071

Pothin J-B, Richard C (2006) A greedy algorithm for optimizing the kernel alignment and the performance of kernel machines. In: Proceedings of 14th European signal processing conference, Florence, Italy, pp 4–8

Pothin J-B, Richard C (2007) Optimal feature representation for kernel machines using kernel-target alignment criterion. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing, Honolulu, USA, vol 3, pp 1065–1068

Pothin J-B, Richard C (2008) Optimizing kernel alignment by data translation in feature space. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing, Las Vegas, USA, pp 3345–3348

Qiu S, Lane T (2009) A framework for multiple kernel support vector regression and its applications to siRNA efficacy prediction. IEEE/ACM Trans Comput Biol Bioinf 6(2):190–199

Ramona M, Richard G, David B (2012, to appear) Multiclass feature selection with kernel Gram-matrix-based criteria. IEEE Trans Neural Netw Learn Syst

Rifkin R, Klautau A (2004) In defense of one-vs-all classification. J Mach Learn Res 5:101–141
Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, New York
Tsochantaridis I, Joachims T, Hofmann T, Altun Y (2005) Large margin methods for structured and interdependent output variables. J Mach Learn Res 6:1453–1484
Vapnik V (1998) Statistical learning theory. Wiley, New York
Vert R (2002) Designing a m-SVM kernel for protein secondary structure prediction. Master's thesis, DEA Informatique de Lorraine
Wang L (2008) Feature selection with kernel class separability. IEEE Trans Pattern Anal Mach Intell 30(9):1534–1546
Wang L, Xue P, Chan KL (2008) Two criteria for model selection in multiclass support vector machines. IEEE Trans Syst Man Cybern B Cybern 38(6):1432–1448
Wang T, Tian S, Huang H, Deng D (2009) Learning by local kernel polarization. Neurocomputing 72(13–15):3077–3084
Wong WWL, Burkowski FJ (2009) Using kernel alignment to select features of molecular descriptors in a QSAR study. IEEE/ACM Trans Comput Biol Bioinform 8(5):1373–1384
Wu M, Farquhar J (2007) A subspace kernel for nonlinear feature extraction. In: Proceedings of the 20th international joint conference on artificial intelligence, Hyderabad, India, pp 1125–1130
Xiong H, Swamy MNS, Ahmad MO (2005) Optimizing the kernel in the empirical feature space. IEEE Trans Neural Netw 16(2):460–474
Zhu X, Kandola J, Ghahramani Z, Lafferty J (2004) Nonparametric transforms of graph kernels for semi-supervised learning. In: Saul LK, Weiss Y, Bottou L (eds) Advances in Neural Information Processing Systems 17, MIT Press, Cambridge, MA