Asymmetric Kernel Learning

Wei Wu[a], Jun Xu[b], Hang Li[b], Satoshi Oyama[c]

[a]*Department of Probability and Statistics, Peking University, No.5 Yiheyuan Road, Beijing, 100871, P. R. China*
[b]*Microsoft Research Asia, 4F Sigma Building, No. 49 Zhichun Road, Beijing, 100190, P. R. China*
[c]*Graduate School of Information Science and Technology, Hokkaido University, Japan*

Abstract

This paper addresses a new *kernel learning* problem, referred to as 'asymmetric kernel learning' (AKL). First, we give the definition of asymmetric kernel and point out that many 'similarity functions' in real applications can be viewed as asymmetric kernels, for example, VSM, BM25, and LMIR in search. Then, we formalize AKL as an optimization problem whose objective function is a regularized loss function on supervised training data. Next, we propose an approach to AKL, which conducts AKL by using kernel methods. In the approach, the space of asymmetric kernels is assumed to be a reproducing kernel Hilbert space (RKHS), and thus existing kernel methods can be employed to learn the optimal asymmetric kernel. We also show that such an RKHS (i.e., space of asymmetric kernels) exists and refer to the kernel generating the RKHS as 'hyper asymmetric kernel' (HAK). We present examples of HAK as well as theoretical basis for constructing HAKs. The proposed approach is applied to search to learn a relevance model from click-through data. Experimental results on web search and enterprise search data show that the model, named 'Robust BM25' can work better than BM25 because it can effectively deal with the term mismatch problem which plagues BM25.

Key words: Asymmetric Kernel, Kernel Learning, Kernel Machines, Web Search

1. Introduction

Kernel methods [32, 17] as well as kernel learning [2, 21, 24, 25] are powerful technologies in machine learning. The key notion in them is kernel function, which is defined as dot product of images of data pairs in feature space (Hilbert space) mapped from input space (Euclidean space or discrete set). A natural interpretation of kernel is similarity function between data points. Conventionally, kernels are symmetric and positive semi-definite. That means kernels are similarity functions over data pairs in a single input space.

We point out that there are many applications in which we need to measure similarities between data pairs from two different input spaces. For example, in web search, we rank documents based on their relevance to the query. We need to measure relevance (similarity) of queries and documents which are from two different spaces: query space and document space. In collaborative filtering, we recommend items to users based on users' preference to items. Preference can be viewed as similarity between items and users which are elements in two different spaces. The same thing can be said to other applications such as image annotation and machine translation. We can represent such similarity functions as 'asymmetric kernels'.

In this paper, we address the problem of learning an asymmetric kernel from training data. As far as we know, this is the first work on the topic. We first formally define asymmetric kernel,

and then explain the importance of the notion by indicating that conventional relevance models in search such as Vector Space Model (VSM), BM25, and Language Models for Information Retrieval (LMIR) are all asymmetric kernels. Asymmetric kernel contains conventional positive semi-definite kernels as special cases. Next, we give a formal definition of asymmetric kernel learning (AKL) and propose performing AKL using kernel methods. AKL is defined as a supervised learning problem in which the optimal asymmetric kernel is selected from the class of possible ones which minimizes the regularized loss function. The key idea of our method is to define the space of asymmetric kernels as a reproducing kernel Hilbert space (RKHS). In this way, existing kernel methods can be employed to perform the learning task and the form of the optimal solution can also be given by the representer theorem. We theoretically demonstrate that such RKHS and the kernel generating the RKHS exist and refer to the kernel as hyper asymmetric kernel (HAK). We also provide theoretical basis for constructing HAKs. Our method for AKL can be viewed as an extension of the method proposed by [24] for positive semi-definite kernel learning. Ong et al.'s method employs hyperkernel and HAKs in this paper are extensions of hyperkernels.

The proposed AKL method is applied to search. Specifically, it is utilized to train a relevance model, to tackle the challenge of term mismatch. The model, named as 'Robust BM25', is based on the traditional BM25 and HAK. The learned Robust BM25 model determines the relevance score of a query document pair on the basis of not only the BM25 score of the query document pair, but also the BM25 scores of similar query and similar document pairs. All the calculations are naturally incorporated in the model of our AKL method. Experimental results on two large scale data sets show that Robust BM25 can indeed solve term mismatch and significantly outperform the baselines.

The contributions of this paper include: 1) formulation of asymmetric kernel learning problem, 2) proposal of a method for asymmetric kernel learning, and 3) demonstration of the usefulness of asymmetric kernel learning in search.

The rest of the paper is organized as follows. A survey of related work is conducted in Section 2, and then asymmetric kernel is defined in Section 3. Section 4 formalizes the problem of AKL and proposes conducting AKL using kernel methods. Section 5 describes how to apply the method to search. Section 6 reports experimental results and Section 7 concludes this paper.

2. Related work

Kernel methods, including the famous Support Vector Machines (SVM) [35], refer to a class of algorithms in machine learning which can be employed in a variety of tasks such as classification, regression, ranking, correlation analysis, and principle component analysis [17, 32]. Kernel methods make use of kernel functions which map a pair of data in the input space (Euclidean space or discrete set) into the feature space (Hilbert space) and compute the dot product between the images in the feature space. Conventional kernels are symmetric and positive semi-definite, in the sense that they are defined over one single input space. The kernel function is called Mercer kernel when it is continuous [32]. In contrast, in this paper, we consider learning of asymmetric kernels [20] which are defined over two different input spaces.

Asymmetric kernels have only been studied by a small number of research groups [34, 31, 20]. For example, in [34], a method of learning an SVM model with asymmetric kernel has been proposed. In [20], asymmetric kernel is defined and applied to Fisher's linear discriminant. We adopt the same definition of asymmetric kernel as in previous work [20], but focus on the learning of it in this paper.

Choosing a suitable kernel function is crucial for all kernel methods. Kernel learning, which aims to automatically learn a kernel function from training data, becomes an important technique [2, 21, 33, 27, 24, 25]. In [21] as well as [2], methods for multiple kernel learning have been proposed, in which the optimal kernel is selected from a class of linear combination of kernels. [24, 25] have proposed learning kernel by using kernel methods, in which the optimal kernel is chosen from the RKHS generated by 'hyperkernel'. The method for asymmetric kernel learning (AKL) in this paper can be viewed as an extension of that of Ong et al.'s.

Term mismatch is one of the major challenges for search, because most of traditional ranking models, including VSM [30], BM25 [28], and LMIR[26, 39], are based on term matching and the ranking results will be inaccurate when term mismatch occurs. To solve the problem, heuristic methods of query expansion or (pseudo) relevance feedback [cf., 29, 37, 30, 3, 23, 7, 40] and Latent Semantic Indexing (LSI) [12] or Probabilistic Latent Semantic Indexing (PLSI) [16] have been proposed and certain progresses have been made. The former approach tackles the problem at the term level and the latter at the topic level. In this paper, we apply AKL to address the term mismatch challenge with a term level approach, and demonstrate that we can learn an asymmetric kernel as ranking model, referred to as Robust BM25.

Click-through data, which records the URLs clicked by the users after their query submissions at a search engine, has been widely used in web search [1, 19, 10]. For example, click-through data has been utilized in training of Ranking SVM model, in which preference pairs over documents given queries are derived from click-through data [19]. Click-through data has also been used for calculating query similarity, because queries which link to the same URL in click-through data may represent the same search intent [5, 11, 36]. In this paper, we utilize click-through data for training Robust BM25 as well as calculating query similarity.

Learning to rank refers to machine learning techniques for constructing ranking models using labeled data [cf., 22]. Several approaches to learning to rank have been proposed and it becomes one of the important technologies in the development of modern search engines [e.g., 8, 38, 14]. The Robust BM25 method proposed in this paper can also be viewed as a learning to rank method. Robust BM25 runs on the top of conventional learning to rank methods. Specifically, it trains a 're-ranking' model online to deal with term mismatch, while conventional learning to rank methods train a ranking model offline for basic ranking. The learning method of Robust BM25 based on SVM proposed in this paper has some similarities to Ranking SVM proposed by [15] and [19], a popular learning to rank algorithm. However, it also has differences from Ranking SVM. For example, the model (kernel function) in Robust BM25 differs from that in Ranking SVM.

3. Asymmetric Kernel

Asymmetric kernel is defined as follows.

3.1. Definition and Properties

Asymmetric kernel measures similarity between two objects from two different spaces. The similarity function is in fact a dot product in the feature space into which the two objects are mapped from their original input spaces, respectively. Asymmetric kernel is formally defined as follows.

Definition 1 (Asymmetric Kernel). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two input spaces, and $\mathcal{H}$ be feature space (Hilbert space). Asymmetric kernel is a function $k : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$, satisfying $k(x, y) = \langle\varphi_X(x), \varphi_Y(y)\rangle_{\mathcal{H}}$*

for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, where φ_X and φ_Y are mapping functions from $\mathcal{X}$ and $\mathcal{Y}$ to $\mathcal{H}$, respectively.

Asymmetric kernel is a natural extension of conventional positive semi-definite kernel. If the two input spaces (also the two mapping functions) are identical in Definition 1, then asymmetric kernel degenerates to positive semi-definite kernel. Asymmetric kernel has the properties as shown below, which enable us to construct more complicated asymmetric kernels from simple asymmetric kernels.

Lemma 1 (Properties of asymmetric kernel). *Let $k_1(x, y)$ and $k_2(x, y)$ be asymmetric kernels on $\mathcal{X} \times \mathcal{Y}$, then the following functions $k : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ are also asymmetric kernels: (1) $\alpha \cdot k_1$ (for all $\alpha \in \mathbb{R}$), (2) $k_1 + k_2$, (3) $k_1 \cdot k_2$.*

Proof Since $k_1(x, y)$ and $k_2(x, y)$ are asymmetric kernels, suppose that $k_1(x, y) = \langle \varphi^1_X(x), \varphi^1_Y(y) \rangle_1$ and $k_2(x, y) = \langle \varphi^2_X(x), \varphi^2_Y(y) \rangle_2$, where $\langle \cdot, \cdot \rangle_1$ is the dot product in N_1-dimensional Hilbert space and $\langle \cdot, \cdot \rangle_2$ is the dot product in N_2-dimensional Hilbert space. N_1 and N_2 are finite or infinite.

Let $\varphi^1_{Xi}(\cdot)$ and $\varphi^1_{Yi}(\cdot)$ be the i^{th} elements of vectors $\varphi^1_X(\cdot)$ and $\varphi^1_Y(\cdot)$, respectively ($i = 1, 2, \cdots, N_1$), and $\varphi^2_{Xi}(\cdot)$ and $\varphi^2_{Yi}(\cdot)$ be the i^{th} elements of vectors $\varphi^2_X(\cdot)$ and $\varphi^2_Y(\cdot)$, respectively ($i = 1, 2, \cdots, N_2$).

(1) Let ${\varphi^1_X}'(x) = \alpha \cdot \varphi^1_X(x)$, we obtain $\alpha \cdot k_1(x, y) = \langle {\varphi^1_X}'(x), \varphi^1_Y(y) \rangle_1$. $\alpha \cdot k_1$ is an asymmetric kernel, $\forall \alpha \in \mathbb{R}$.

(2) Let $\varphi_X(x) = (\varphi^1_X(x), \varphi^2_X(x))$, and $\varphi_Y(y) = (\varphi^1_Y(y), \varphi^2_Y(y))$, we obtain $\langle \varphi_X(x), \varphi_Y(y) \rangle = \langle \varphi^1_X(x), \varphi^1_Y(y) \rangle_1 + \langle \varphi^2_X(x), \varphi^2_Y(y) \rangle_2 = k_1(x, y) + k_2(x, y)$. $k_1 + k_2$ is an asymmetric kernel.

(3) Let $\varphi_X(x) = \varphi^1_X(x) \otimes \varphi^2_X(x)$ and $\varphi_Y(y) = \varphi^1_Y(y) \otimes \varphi^2_Y(y)$. $\varphi_X(x)$ is a vector whose elements are $\{\varphi^1_{Xi}(x)\varphi^2_{Xj}(x)\}$,$1 \leqslant i \leqslant N_1,\ 1 \leqslant j \leqslant N_2$ and $\varphi_Y(y)$ is a vector whose elements are $\{\varphi^1_{Yi}(y)\varphi^2_{Yj}(y)\}$, $1 \leqslant i \leqslant N_1,\ 1 \leqslant j \leqslant N_2$. We obtain

$$\begin{aligned}
\langle \varphi_X(x), \varphi_Y(y) \rangle &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \varphi^1_{Xi}(x)\varphi^2_{Xj}(x)\varphi^1_{Yi}(y)\varphi^2_{Yj}(y) \\
&= \sum_{i=1}^{N_1} \varphi^1_{Xi}(x)\varphi^1_{Yi}(y) \sum_{j=1}^{N_2} \varphi^2_{Xj}(x)\varphi^2_{Yj}(y) \\
&= \sum_{i=1}^{N_1} \varphi^1_{Xi}(x)\varphi^1_{Yi}(y) k_2(x, y) \\
&= k_1(x, y)k_2(x, y).
\end{aligned}$$

$k_1 \cdot k_2$ is an asymmetric kernel.

3.2. Asymmetric Kernel as Similarity Functions

Many similarity functions in practical applications can be viewed as asymmetric kernels. These include document retrieval (search), collaborative filtering, and machine translation. For example, in search given a query the relevance function assigns scores to documents which represent the relevance degrees of the documents with respect to the query. The relevance function is defined over the query and document spaces, and measures the 'similarity' between query and document. Traditionally, VSM, BM25, and LMIR have been employed as relevance models, which can be viewed as asymmetric kernels (VSM is simply a positive semi-definite kernel).

Let $\mathcal{Q}$ and $\mathcal{D}$ denote query and document spaces. Each dimension in the two spaces corresponds to a term, and query and document are respectively represented as vectors in the two spaces. Let $\mathcal{H}$ denote a Hilbert space endowed with dot product $\langle \cdot, \cdot \rangle$ (it is in fact an n-dimensional Euclidean space where n is the number of unique terms).

3.2.1. VSM

Given query $q \in \mathcal{Q}$ and document $d \in \mathcal{D}$, VSM is calculated as

$$\text{VSM}(q, d) = \langle \varphi_Q^{\text{VSM}}(q), \varphi_D^{\text{VSM}}(d) \rangle,$$

where $\varphi_Q^{\text{VSM}}(q)$ and $\varphi_D^{\text{VSM}}(d)$ are mappings to $\mathcal{H}$ from $\mathcal{Q}$ and $\mathcal{D}$, respectively.

$$\varphi_Q^{\text{VSM}}(q)_t = idf(t) \cdot tf(t, q)$$

and

$$\varphi_D^{\text{VSM}}(d)_t = idf(t) \cdot tf(t, d),$$

where t is a term, $tf(t, q)$ is frequency of term t in query q, $tf(t, d)$ is frequency of term t in document d, $idf(t)$ is inverse document frequency of term t.

3.2.2. BM25

Given query $q \in \mathcal{Q}$ and document $d \in \mathcal{D}$, BM25 is calculated as

$$\text{BM25}(q, d) = \langle \varphi_Q^{\text{BM25}}(q), \varphi_D^{\text{BM25}}(d) \rangle, \tag{1}$$

where $\varphi_Q^{\text{BM25}}(q)$ and $\varphi_D^{\text{BM25}}(d)$ are mappings to $\mathcal{H}$ from $\mathcal{Q}$ and $\mathcal{D}$, respectively.

$$\varphi_Q^{\text{BM25}}(q)_t = \frac{(k_3 + 1) \times tf(t, q)}{k_3 + tf(t, q)}$$

and

$$\varphi_D^{\text{BM25}}(d)_t = idf(t) \frac{(k_1 + 1) \times tf(t, d)}{k_1 \left(1 - b + b \cdot \frac{\text{len}(d)}{\text{avgDocLen}}\right) + tf(t, d)},$$

where k_1, k_3, and b are parameters. Moreover, len(d) is the length of document d and avgDocLen is average length of documents in the collection.

3.2.3. LMIR

We employ Dirichlet smoothing as an example. Other smoothing methods such as Jelinek-Mercer (JM) can also be used. Given query $q \in \mathcal{Q}$ and document $d \in \mathcal{D}$, the LMIR with Dirichlet smoothing is calculated as

$$\text{LMIR}(q, d) = \langle \varphi_Q^{\text{LMIR}}(q), \varphi_D^{\text{LMIR}}(d) \rangle,$$

where $\phi_Q^{\text{LMIR}}(q)$ and $\phi_D^{\text{LMIR}}(d)$ are $(n+1)$-dimensional mappings to $\mathcal{H}$ from $\mathcal{Q}$ and $\mathcal{D}$, respectively. For $t = 1, 2, \cdots, n$, $\phi_Q^{\text{LMIR}}(q)_t$ and $\phi_D^{\text{LMIR}}(d)_t$ are defined as

$$\varphi_Q^{\text{LMIR}}(q)_t = tf(t, q)$$

and

$$\varphi_D^{\text{LMIR}}(d)_t = \log\left(1 + \frac{tf(t,d)}{\mu P(t)}\right),$$

where μ is a free smoothing parameter, $P(t)$ is probability of term t in the whole collection. $P(t)$ plays a similar role as inverse document frequency $idf(t)$ in VSM and BM25. The $(n+1)^{th}$ entries of $\varphi_Q^{\text{LMIR}}(q)$ and $\varphi_D^{\text{LMIR}}(d)$ are defined as

$$\varphi_Q^{\text{LMIR}}(q)_{n+1} = \text{len}(q)$$

and

$$\varphi_D^{\text{LMIR}}(d)_{n+1} = \log\frac{\mu}{\text{len}(d)+\mu},$$

where $\text{len}(q)$ and $\text{len}(d)$ are the lengthes of query q and document d, respectively.

One can certainly define asymmetric kernels (e.g., VSM, BM25, and LMIR) and exploit them in practice. When training data is available, it is also desirable to automatically train an asymmetric kernel; this leads to the problem of asymmetric kernel learning.

4. Our Approach to Asymmetric Kernel Learning

We formalize the asymmetric kernel learning (AKL) problem and propose a method for the task using kernel methods.

4.1. Formulation of Asymmetric Kernel Learning

Suppose that we are given training data $S = \{(x_i, y_i), t_i\}_{i=1}^N$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are a pair of objects, and $t_i \in \mathcal{T}$ is their response. The training data can be that for classification, regression, or ranking. AKL aims to select the optimal asymmetric kernel from the class of possible ones which can make accurate prediction on the training data as well as future test data. AKL is formally defined as the following optimization problem:

$$\min_{k\in\mathcal{K}\subseteq\mathcal{A}} \frac{1}{N}\sum_{i=1}^{N} l(k(x_i, y_i), t_i) + \Omega(k),$$

where $\mathcal{A} = \{k : \mathcal{X}\times\mathcal{Y} \to \mathbb{R} | k(x,y) = \langle\varphi_X(x), \varphi_Y(y)\rangle\}$ is the space of all asymmetric kernels, $\mathcal{K}$ is a subspace of $\mathcal{A}$, $l(\cdot,\cdot)$ is a loss function, and Ω is a regularizer.

To conduct AKL, we need to specify (1) function space $\mathcal{K}$, (2) regularizer Ω, and (3) loss function $l(\cdot,\cdot)$. Function space determines the scope of learning, regularizer controls the complexity of function, and loss function measures prediction accuracy.

4.2. Learning Asymmetric Kernels with Kernel Methods

In this paper, we propose performing AKL using kernel methods. The key idea in our approach is to assume that the space of asymmetric kernels $\mathcal{K}$ is also an RKHS.

We first specify the original AKL problem as follows

$$\min_{k\in\mathcal{K}} \frac{1}{N}\sum_{i=1}^{N} l(k(x_i, y_i), t_i) + \frac{\lambda}{2}\|k\|_{\mathcal{K}}^2, \tag{2}$$

where $\lambda > 0$ is a coefficient, $\mathcal{K}$ is a Hilbert space of asymmetric kernels, and $\|k\|_{\mathcal{K}}$ denotes regularization on space $\mathcal{K}$. If $\mathcal{K}$ is also an RKHS generated by a postive semi-definite kernel $\bar{k} : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$, then Eq. (2) is equivalent to the optimization problem of kernel methods (note that the learned function has two arguments.). According to the representer theorem of kernel methods [32], the optimal solution of Eq. (2) is in the form

$$k^*(x, y) = \sum_{i=1}^{N} \alpha_i \bar{k}((x_i, y_i), (x, y)), \tag{3}$$

where $\alpha_i \in \mathbb{R}, 1 \leq i \leq N$, and N denotes number of training instances.

That is to say, if $\mathcal{K}$ is also an RKHS, then the AKL problem can be solved with kernel methods. The question then is whether there exists space $\mathcal{K}$, or equivalently kernel $\bar{k}$. We show below that it is the case and refer to the kernel $\bar{k}$ as hyper asymmetric kernel (HAK).

4.3. Hyper Asymmetric Kernel

4.3.1. Definition of Hyper Asymmetric Kernel

HAK is defined as follows.

Definition 2 (Hyper Asymmetric Kernel). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two input spaces. $\bar{k}((x, y), (x', y'))$ is called Hyper Asymmetric Kernel, if it has the following properties. (1) $\bar{k} : (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \to \mathbb{R}$ is a positive semi-definite kernel. (2) All the elements in the RKHS generated by $\bar{k}$ are also asymmetric kernels on $\mathcal{X}$ and $\mathcal{Y}$.*

If the two input spaces $\mathcal{X}$ and $\mathcal{Y}$ are identical in Definition 2, then HAK degenerates to hyperkernel proposed in [24, 25].

4.3.2. Example of HAK

The following kernel is an HAK.

$$\bar{k}((x, y), (x', y')) = g(x, y)k_X(x, x')k_Y(y, y')g(x', y'), \tag{4}$$

where $g(\cdot, \cdot)$ is an asymmetric kernel function, and k_X and k_Y are two Mercer kernels [32] on spaces $\mathcal{X} \times \mathcal{X}$ and $\mathcal{Y} \times \mathcal{Y}$, respectively. First, we prove that $\bar{k}$ is a positive semi-definite kernel.

Theorem 1. *$k_X : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_Y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ are two positive semi-definite kernels. For any function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, $\bar{k}$ defined in Eq. (4) is a positive semi-definite kernel.*

Proof of Theorem 1 is given in Appendix A. Next, we prove that all of the elements in the RKHS generated by $\bar{k}$ are asymmetric kernels.

Theorem 2. *Suppose $g(x, y)$ is an asymmetric kernel. Given any two Mercer kernels $k_X : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_Y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the RKHS $\mathcal{K}$ generated by $\bar{k}$ defined in Eq. (4) is a subspace of $\mathcal{A} = \{k : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} | k(x, y) = \langle \varphi_X(x), \varphi_Y(y) \rangle\}$.*

Proof of Theorem 2 is given in Appendix B. From Theorems 1 and 2, we conclude that $\bar{k}$ is an HAK. We will show that this HAK has an important application in search.

[4] propose a pairwise kernel for collaborative filtering. Pairwise kernel is defined as $\bar{k}_C((u, i), (u', i')) = k_U(u, u') \cdot k_I(i, i')$, where k_U and k_I are kernels defined on the spaces of users and items, respectively. Obviously, $\bar{k}_C$ is a specific case of the HAK in Eq. (4).

4.3.3. Construction of HAK

More general and complicated HAKs can be constructed on the basis of the following two theorems. Both of them are generalizations of the kernel in Eq. (4).

Theorem 3 (Constructing Power Series). *Given two Mercer kernels $k_X : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_Y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, for any asymmetric kernel $g(x, y)$ and $\{c_i\}_{i=1}^{n} \subset \mathbb{R}^+$, $\bar{k}_P$ defined below is a hyper asymmetric kernel.*

$$\bar{k}_P((x, y), (x', y')) = \sum_{i=0}^{\infty} c_i \cdot g(x, y) \left(k_X(x, x')k_Y(y, y')\right)^i g(x', y'),$$

where the convergence radius of $\sum_{i=0}^{\infty} c_i \xi^i$ is R, $|k_X(x, x')| < \sqrt{R}, |k_Y(y, y')| < \sqrt{R}$, for any x, x', y, y'.

Theorem 4 (Combining Multiple Kernels). *Given two finite sets of Mercer kernels $K_X = \left\{k_i^X(x, x')\right\}_{i=1}^{n}$ and $K_Y = \left\{k_i^Y(y, y')\right\}_{i=1}^{n}$. For any asymmetric kernel $g(x, y)$ and $\{c_i\}_{i=1}^{n} \subset \mathbb{R}^+$, $\bar{k}_M$ defined below is a hyper asymmetric kernel.*

$$\bar{k}_M((x, y), (x', y')) = \sum_{i=1}^{n} c_i \cdot g(x, y)k_i^X(x, x')k_i^Y(y, y')g(x', y').$$

Proofs of Theorem 3 and Theorem 4 are given in Appendix C and Appendix D, respectively.

5. Application to Search

In this section we show how our approach to asymmetric kernel learning can be applied to search, in order to address one of the most critical challenges: term mismatch.

5.1. Term Mismatch in Search

Existing relevance ranking models in search, including VSM, BM25, and LMIR (as explained above, all of them can be viewed as asymmetric kernels) calculate the relevance of the document with respect to the query on the basis of term matching, i.e., the terms (words) shared by the query and document (cf., Eq. (1)). However, a document and a query can still be relevant, even when they do not share any term, for example, the query is 'NY' while the document only contains 'New York'. In such case the document cannot be ranked high by the conventional relevance models, and term mismatch occurs. In fact, term mismatch poses one of the most critical challenges in search.

5.2. Robust BM25

We try to learn a more reliable ranking model from data as an extension of BM25 to deal with term mismatch, called 'Robust BM25'. Robust BM25 is actually an asymmetric kernel. We first give the definition of Robust BM25 and then explain why it has the capability to handle term mismatch.

First, we can specify the HAK in Eq. (4) as follows

$$\bar{k}((q, d), (q', d')) = k_{BM25}(q, d)k_Q(q, q')k_D(d, d')k_{BM25}(q', d'),$$

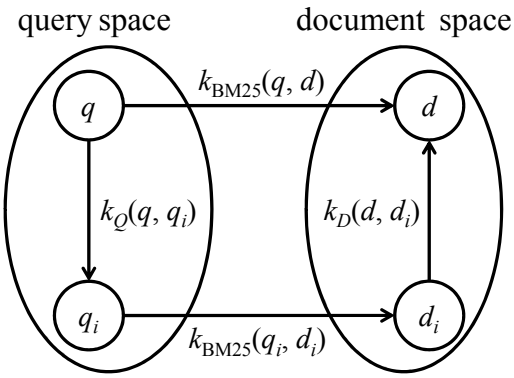

Figure 1: Robust BM25 deals with term mismatch by using the neighbors in query spaces and document spaces.

where $k_{BM25}(q, d)$ is the BM25 asymmetric kernel, $k_Q : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$ and $k_D : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ are positive semi-definite kernels on query space and document space, which represent query similarity and document similarity, respectively.

With training data and our AKL method (cf., Eq. (3)), we can learn the following asymmetric kernel, which is referred to as Robust BM25 (RBM25)[1].

$$k_{RBM25}(q, d) = k_{BM25}(q, d) \cdot \sum_{i=1}^{N} \alpha_i \cdot k_Q(q, q_i) k_D(d, d_i) k_{BM25}(q_i, d_i), \tag{5}$$

where α_i is learned from training data.

Robust BM25 can effectively deal with the term mismatch problem, as shown in Figure 1. Suppose that the query space contains queries as elements and has the kernel function k_Q as similarity function. Given query q, one can find its similar queries q_i based on $k_Q(q, q_i)$ (its neighbors). Similarly, the document space contains documents as elements and has the kernel function k_D as similarity function. Given document d, one can find its similar documents d_i based on $k_D(d, d_i)$ (its neighbors). The relevance model BM25 is defined as an asymmetric kernel between query and document over the two spaces. Term mismatch means that BM25 score $k_{BM25}(q, d)$ is not reliable.

One possible way to deal with the problem is to use the neighboring queries q_i and documents d_i to smooth the BM25 score of q and d, as that of in the k-nearest neighbor algorithm [9, 13]. In other words, we employ the k-nearest neighbor method in both the query and document spaces to calculate the final relevance score (cf., Figure 1). This is exactly what Robust BM25 does. More specifically, Robust BM25 determines the ranking score of query q and document d, not only based on the relevance score between q and d themselves (i.e., $k_{BM25}(q, d)$), but also based on the relevance scores between similar queries q_i and similar documents d_i (i.e., $k_{BM25}(q_i, d_i)$), and it makes a weighted linear combination of the relevance scores (5).

5.3. Implementation

To learn Robust BM25, we need to decide the query similarity kernel, document similarity kernel, training data creation technique, and optimization technique. We explain one way of implementing them using click-through data.

[1]To avoid zero value of $k_{BM25}(q, d)$, one can add a small constant, i.e., to assume $k_{BM25}(q, d) \geq \epsilon$.

First, the document similarity kernel $k_D(d, d')$ between d and d' is simply defined as cosine similarity between the titles and URLs of the two documents. The query similarity kernel $k_Q(q, q')$ between q and q' is defined as Pearson Correlation Coefficient between clicked URLs of the two queries on a click-through bipartite graph:

$$k_Q(q, q') = \frac{\sum_{i=1}^{n}(u_i - \bar{u})(v_i - \bar{v})}{\sqrt{\sum_{i=1}^{n}(u_i - \bar{u})^2}\sqrt{\sum_{i=1}^{n}(v_i - \bar{v})^2}},$$

where u_i and v_i denote numbers of clicks on URL i by query q and q' respectively, $\bar{u}$ and $\bar{v}$ denote average numbers of clicks for q and q', and n denotes total number of clicked URLs by q and q'. Intuitively, if two queries have many co-clicked URLs, then they will be regarded similar, e.g., 'NY' and 'New York' may have many co-clicked URLs. Note that Pearson Correlation Coefficient is a kernel.

Following the proposal in [19], we generate pairwise training data from click-through data. More precisely, For each query q_i we derive preference pairs (d_i^+, d_i^-), where d_i^+ and d_i^- mean that document d_i^+ is more preferred than d_i^- with respect to query q_i.

Finally, we take the pairwise training data as input to AKL method and learn the optimal asymmetric kernel (Robust BM25). We use the Hinge loss as loss function, the objective function then becomes

$$\min_{k \in \mathcal{K}} \sum_{i=1}^{M} \left[1 - (k(q_i, d_i^+) - k(q_i, d_i^-))\right]_+ + \frac{\lambda}{2}\|k\|_{\mathcal{K}}^2,$$

where M is number of preference pairs in training data (note that this is similar to Ranking SVM [15]). The dual problem may be written as

$$\max_{\vec{\theta}} \sum_{i=1}^{M} \theta_i - \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} \theta_i \theta_j \mathcal{W}(i, j) \quad s.t. \quad 0 \le \theta_i \le \frac{1}{\lambda},$$

where

$$\begin{aligned}\mathcal{W}(i, j) = k_Q(q_i, q_j) \cdot \Big[& k_D(d_i^+, d_j^+) k_{BM25}(q_i, d_i^+) k_{BM25}(q_j, d_j^+) \\ & - k_D(d_i^+, d_j^-) k_{BM25}(q_i, d_i^+) k_{BM25}(q_j, d_j^-) \\ & - k_D(d_i^-, d_j^+) k_{BM25}(q_i, d_i^-) k_{BM25}(q_j, d_j^+) \\ & + k_D(d_i^-, d_j^-) k_{BM25}(q_i, d_i^-) k_{BM25}(q_j, d_j^-) \Big].\end{aligned}$$

We can solve the dual problem and obtain the following solution

$$k_{RBM25}(q, d) = k_{BM25}(q, d) \cdot \sum_{i=1}^{M} \theta_i \cdot k_Q(q, q_i) \left[k_{BM25}(q_i, d_i^+) k_D(d_i^+, d) - k_{BM25}(q_i, d_i^-) k_D(d_i^-, d)\right].$$

In online search, given a query, we first retrieve the queries similar to it, then individually retrieve documents with the original query and the similar queries, combine the retrieved documents, train a Robust BM25 model using click-through data, and rank the documents with their Robust BM25 scores (note that we need a Robust BM25 for each query). The time complexities of training Robust BM25 is of order $O(M^2)$, where M is number of preference pairs. Since the number of retrieved documents is small, search with Robust BM25 can be carried out efficiently.

Table 1: Statistics on web search and enterprise search datasets.

	Web search	Enterprise search
# of judged queries	8,294	2,864
# of judged query-URL pairs	1,715,844	282,130
# of search impressions in click-through	490,085,192	17,383,935
# of unique queries in click-through	14,977,647	2,368,640
# of unique URLs in click-through	30,166,304	2,419,866
# of clicks in click-through	2,605,404,156	4,996,027

6. Experiment

6.1. Experiment Setting

We conducted experiments to test the performances of Robust BM25. In our experiments, we used two large scale datasets from a commercial web search engine and an enterprise search engine running in an IT company. The two datasets consist of query-URL pairs and their relevance judgments. The relevance judgments can be 'Perfect', 'Excellent', 'Good', 'Fair', or 'Bad'. Besides, we also collected large scale click-through data from both search engines. Table 1 shows the statistics on the two datasets. The click-through data in both datasets was split into two parts, one for learning query similarity kernel and the other for learning Robust BM25.

BM25 and query expansion [37] were selected as baselines. The pairwise kernel, which was initially proposed for collaborative filtering [4], was chosen as another baseline. As evaluation measures, we used MAP [3] and NDCG [18] at positions 1, 3, and 5. When calculating MAP, we define the ranks 'Perfect' and 'Excellent' as relevant and the other three ranks as irrelevant.

6.2. Experimental Results

Table 2 reports the results on the web search data and enterprise data. We can see that Robust BM25 outperforms the baselines, in terms of all measures on both datasets. We conducted significant tests (t-test) on the improvements. The results show that the improvements are all statistically significant (p-value < 0.05). We have conducted analysis on the cases in which Robust BM25 performs better and found that the reason is that Robust BM25 can indeed effectively address the term mismatch problem. Pairwise kernel outperforms BM25 and query expansion, which indicates that it is better to learn a ranking model in search. However, its performance is still lower than Robust BM25, suggesting that it is better to include BM25 in the final relevance model, as in Robust BM25.

6.3. Discussions

We investigated the reasons that Robust BM25 can outperform the baselines, using the experiments on web search data as examples. It seems that Robust BM25 can effectively deal with term mismatch with its mechanisms: using query similarity and document similarity.

Table 3 gives an example. The query, web page, and label are 'wal**l**mart', which is a typo, 'http://www.walmart.com' with title 'Walmart.com: Save money. Live better', and 'Perfect', which means that the page should be ranked at top one position, respectively. BM25 cannot give a high score to the page, as there is a mismatch between query and page. (Note that there is difference between the query term 'wallmart' and the document term 'walmart'.) In contrast,

Table 2: Ranking accuracies on web search and enterprise search data.

		MAP	NDCG@1	NDCG@3	NDCG@5
Web search	Robust BM25	0.1192	0.2480	0.2587	0.2716
	Pairwise Kernel	0.1123	0.2241	0.2418	0.2560
	Query Expansion	0.0963	0.1797	0.2061	0.2237
	BM25	0.0908	0.1728	0.2019	0.2180
Enterprise search	Robust BM25	0.3122	0.4780	0.5065	0.5295
	Pairwise Kernel	0.2766	0.4465	0.4769	0.4971
	Query Expansion	0.2755	0.4076	0.4712	0.4958
	BM25	0.2745	0.4246	0.4531	0.4741

Table 3: Example 1 from web search.

Query	wallmart
Similar queries	'walmart', 'wal mart', 'walmarts'
Page	http://www.walmart.com
Title	Walmart.com: Save money. Live better
Rate	Perfect

Robust BM25 can effectively leverage the similar queries such as 'walmart', 'wal mart', and 'walmarts' and rank the web page to position one.

Table 4 gives another example. The query is 'mensmagazines', which is a tail query and does not have a similar query found in the click-through data. The web page is 'http://en.wikipedia.org/wiki/List_of_men's_magazines' (referred to as Page1) and the relevance label is 'Excellent'. There is a mismatch, because there is no sufficient knowledge to break query 'mensmagazines' into 'mens' and 'magazines'. As a result, BM25 cannot rank Page1 high. In contrast, Robust BM25 uses similar documents to calculate the relevance. Specifically, it utilizes a similar web page 'http://www.askmen.com/links/sections/mensmagazines.html' (referred to as Page2), which contains the term 'mensmagazines' in its URL. The original query can match well with Page2. Besides, Page1 and Page2 are also similar because they have the common terms 'men' and 'magazines' in titles. Therefore, Robust BM25 can assign a high score to Page1.

7. Conclusion

In the paper, we have studied the problem of asymmetric kernel learning (AKL). We have pointed out the importance of asymmetric kernel functions in practice by showing the relevance models in search such as VSM, BM25, and LMIR are actually asymmetric kernels. We have then proposed a method for AKL using kernel methods. The key idea is to assume that the space of asymmetric kernels is also an RKHS, and employ existing kernel methods to perform the learning task. We refer to the kernel generating the RKHS as hyper asymmetric kernel (HAK). We have then given examples of HAK and provided theoretical basis for constructing HAK. Finally, we have shown that we can apply our method of AKL to search in order to effectively deal with the term mismatch problem. The learned model Robust BM25 is a natural and more reliable extension of conventional BM25 model.

Table 4: Example 2 from web search.

Query	mensmagazines
Page1	http://en.wikipedia.org/wiki/List_of_men's_magazines
Title1	List of men's magazines - Wikipedia, the free encyclopedia
Rate1	Excellent
Page2	http://www.askmen.com/links/sections/mensmagazines.html
Title2	AskMen.com - Men's magazines

References

[1] Agichtein, E., Brill, E., Dumais, S., 2006. Improving web search ranking by incorporating user behavior information. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 19–26.

[2] Bach, F. R., Lanckriet, G. R. G., Jordan, M. I., 2004. Multiple kernel learning, conic duality, and the smo algorithm. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning. ACM, New York, NY, USA, p. 6.

[3] Baeza-Yates, R. A., Ribeiro-Neto, B., 1999. Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[4] Basilico, J., Hofmann, T., 2004. Unifying collaborative and content-based filtering. In: ICML '04: Proceedings of the twenty-first international conference on Machine learning. ACM, New York, NY, USA, pp. 65–72.

[5] Beeferman, D., Berger, A., 2000. Agglomerative clustering of a search engine query log. In: KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp. 407–416.

[6] Berg, C., Christensen, J., Ressel, P., 1984. Harmonic analysis on semigroups: theory of positive definite and related functions. Springer.

[7] Broder, A., Ciccolo, P., Gabrilovich, E., Josifovski, V., Metzler, D., Riedel, L., Yuan, J., 2009. Online expansion of rare queries for sponsored search. In: WWW '09: Proceedings of the 18th international conference on World wide web. ACM, New York, NY, USA, pp. 511–520.

[8] Burges, C. J., Ragno, R., Le, Q. V., 2007. Learning to rank with nonsmooth cost functions. In: Schölkopf, B., Platt, J., Hoffman, T. (Eds.), Advances in Neural Information Processing Systems 19. MIT Press, Cambridge, MA, pp. 193–200.

[9] Cover, T., Hart, P., 1967. Nearest neighbor pattern classification 13 (1), 21– 27.

[10] Craswell, N., Szummer, M., 2007. Random walks on the click graph. In: SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 239–246.

[11] Cui, H., Wen, J.-R., Nie, J.-Y., Ma, W.-Y., 2003. Query expansion by mining user logs. IEEE Trans. on Knowl. and Data Eng. 15 (4), 829–839.

[12] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R., 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407.

[13] Dudani, S., April 1976. The distance-weighted k-nearest-neighbor rule 6 (4), 325–327.

[14] Geng, X., Liu, T.-Y., Qin, T., Arnold, A., Li, H., Shum, H.-Y., 2008. Query dependent ranking using k-nearest neighbor. In: SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 115–122.

[15] Herbrich, R., Graepel, T., Obermayer, K., 2000. Large margin rank boundaries for ordinal regression. In: Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D. (Eds.), Advances in Large Margin Classifiers. MIT Press, Cambridge, MA, pp. 115–132.

[16] Hofmann, T., 1999. Probabilistic latent semantic indexing. In: SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 50–57.

[17] Hofmann, T., Scholkopf, B., Smola, A., 2008. Kernel methods in machine learning. Annals of Statistics 36 (3), 1171.

[18] Järvelin, K., Kekäläinen, J., 2000. Ir evaluation methods for retrieving highly relevant documents. In: SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 41–48.

[19] Joachims, T., 2002. Optimizing search engines using clickthrough data. In: KDD '02: Proceedings of the eighth

ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, NY, USA, pp. 133–142.

[20] Koide, N., Yamashita, Y., 2006. Asymmetric kernel method and its application to fisher's discriminant. In: ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition. IEEE Computer Society, Washington, DC, USA, pp. 820–824.

[21] Lanckriet, G. R. G., Cristianini, N., Bartlett, P., Ghaoui, L. E., Jordan, M. I., 2004. Learning the kernel matrix with semidefinite programming. J. Mach. Learn. Res. 5, 27–72.

[22] Liu, T.-Y., 2009. Learning to rank for information retrieval. Found. Trends Inf. Retr. 3 (3), 225–331.

[23] Mitra, M., Singhal, A., Buckley, C., 1998. Improving automatic query expansion. In: SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 206–214.

[24] Ong, C. S., Smola, A. J., Williamson, R. C., ???? Hyperkernels. In: S. Becker, S. T., Obermayer, K. (Eds.), Advances in Neural Information Processing Systems 15. MIT Press.

[25] Ong, C. S., Smola, A. J., Williamson, R. C., 2005. Learning the kernel with hyperkernels. J. Mach. Learn. Res. 6, 1043–1071.

[26] Ponte, J. M., Croft, W. B., 1998. A language modeling approach to information retrieval. In: SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 275–281.

[27] Rakotomamonjy, A., Bach, F. R., Canu, S., Grandvalet, Y., November 2008. Simplemkl. Journal of Machine Learning Research 9, 2491–2521.

[28] Robertson, S. E., Hull, D. A., 2000. The TREC-9 filtering track final report. In: TREC-9. pp. 25–40.

[29] Salton, G., Buckley, C., 1997. Improving retrieval performance by relevance feedback, 355–364.

[30] Salton, G., McGill, M. J., 1986. Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA.

[31] Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K. R., Rätsch, G., Smola, A. J., 1999. Input space versus feature space in kernel-based methods. IEEE Transactions on Neural Networks 10 (5), 1000–1017.

[32] Schölkopf, B., Smola, A. J., 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA.

[33] Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B., 2006. Large scale multiple kernel learning. J. Mach. Learn. Res. 7, 1531–1565.

[34] Tsuda, K., 1998. Support vector classifier with asymmetric kernel functions. In: in European Symposium on Artificial Neural Networks (ESANN). pp. 183–188.

[35] Vapnik, V. N., 1995. The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA.

[36] Wen, J., Nie, J., Zhang, H., 2002. Query clustering using user logs. ACM Trans. Inf. Syst. 20 (1), 59–81.

[37] Xu, J., Croft, W. B., 1996. Query expansion using local and global document analysis. In: SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 4–11.

[38] Xu, J., Li, H., 2007. Adarank: a boosting algorithm for information retrieval. In: SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, New York, NY, USA, pp. 391–398.

[39] Zhai, C., Lafferty, J., 2004. A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22 (2), 179–214.

[40] Zhuang, Z., Cucerzan, S., 2006. Re-ranking search results using query logs. In: CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management. ACM, New York, NY, USA, pp. 860–861.

A. Proof of Theorem 1

Proof We need to show $\bar{k}(\cdot,\cdot)$ is symmetric and positive semi-definite.

Symmetric Since $k_X(\cdot,\cdot)$ and $k_Y(\cdot,\cdot)$ are symmetric, $\forall x, x' \in \mathcal{X}$ and $y, y' \in \mathcal{Y}$, we have

$$\begin{aligned}\bar{k}((x,y),(x',y')) &= g(x,y)k_X(x,x')k_Y(y,y')g(x',y') \\ &= g(x',y')k_X(x',x)k_Y(y',y)g(x,y) \\ &= \bar{k}((x',y'),(x,y)).\end{aligned}$$

Positive semi-definite $\forall\{\alpha_i\}_{i=1}^n \subset \mathbb{R}, \{(x_i,y_i)\}_{i=1}^n \subset \mathcal{X}\times\mathcal{Y}$, we have

$$\sum_{i,j=1}^{n} \alpha_i\alpha_j g(x_i,y_i)g(x_j,y_j) = \left(\sum_{i=1}^{n}\alpha_i g(x_i,y_i)\right)^2 \geqslant 0.$$

Since $k_X(\cdot,\cdot)$ and $k_Y(\cdot,\cdot)$ are positive semi-definite, we conclude that $g(x,y)g(x',y')k_X(x,x')k_Y(y,y')$ is positive semi-definite [cf., 6, Theorem 1.12].

B. Proof of Theorem 2

To prove the theorem, we need the following two lemmas:

Lemma 2. *Suppose that $g(x,y)$ is an asymmetric kernel, and k_X and k_Y are two Mercer kernels. Given any finite example set $\{(x_i,y_i)\}_{i=1}^N \subset \mathcal{X}\times\mathcal{Y}$, $k_N(x,y) = \sum_{i=1}^N \alpha_i g(x,y)k_X(x,x_i)k_Y(y,y_i)g(x_i,y_i)$ is an asymmetric kernel.*

Proof Since $g(x,y)$ is an asymmetric kernel, according to Lemma 1, we only need to show that $\sum_{i=1}^N \alpha_i k_X(x,x_i)k_Y(y,y_i)g(x_i,y_i)$ is an asymmetric kernel.

Since $k_X(x,x')$ and $k_Y(y,y')$ are Mercer kernels, we obtain

$$\begin{aligned}k_X(x,x') &= \langle\psi_X(x),\psi_X(x')\rangle_{\mathcal{H}_X}, \\ k_Y(y,y') &= \langle\psi_Y(y),\psi_Y(y')\rangle_{\mathcal{H}_Y},\end{aligned}$$

where $\psi_X : \mathcal{X}\to\mathcal{H}_X$ and $\psi_Y : \mathcal{Y}\to\mathcal{H}_Y$ are corresponding feature mappings, and $\mathcal{H}_X$ and $\mathcal{H}_Y$ are Hilbert spaces with respect to ψ_X and ψ_Y, respectively.

Let $\mathcal{H} = \mathcal{H}_X$ and $\varphi_X(x) = \psi_X(x)$,

$$\varphi_Y^N(y) = \sum_{i=1}^{N}\alpha_i g(x_i,y_i)\psi_X(x_i)\psi_Y^\top(y_i)\psi_Y(y).$$

Note that $\varphi_Y^N = \gamma_N\circ\psi_Y$, where $\gamma_N = \sum_{i=1}^N \alpha_i g(x_i,y_i)\psi_X(x_i)\psi_Y^\top(y_i)$ is a linear operator from $\mathcal{H}_Y$ to

$\mathcal{H}_X = \mathcal{H}$. Thus, we have

$$\begin{aligned}\langle \varphi_X(x), \varphi_Y^N(y)\rangle_{\mathcal{H}} &= \langle \psi_X(x), \sum_{i=1}^{N} \alpha_i g(x_i, y_i)\psi_X(x_i)\psi_Y^\top(y_i)\psi_Y(y)\rangle_{\mathcal{H}_X} \\ &= \psi_X^\top(x) \sum_{i=1}^{N} \alpha_i g(x_i, y_i)\psi_X(x_i)\psi_Y^\top(y_i)\psi_Y(y) \\ &= \sum_{i=1}^{N} \alpha_i g(x_i, y_i)\psi_X^\top(x)\psi_X(x_i)\psi_Y^\top(y_i)\psi_Y(y) \\ &= \sum_{i=1}^{N} \alpha_i k_X(x, x_i)k_Y(y, y_i)g(x_i, y_i).\end{aligned}$$

This means $\sum_{i=1}^{N} \alpha_i k_X(x, x_i)k_Y(y, y_i)g(x_i, y_i)$ is an asymmetric kernel. Thus, we conclude that $k_N(x, y)$ is an asymmetric kernel.

Lemma 3. *Given any two positive semi-definite kernels $k_X : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $k_Y : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Suppose $\psi_Y : \mathcal{Y} \to \mathcal{H}_Y$ is the feature mapping of $k_Y(\cdot, \cdot)$. $\mathcal{H}_Y$ is a Hilbert space endowed with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_Y}$. Given any sets $\{x_i\}_{i=1}^N \subset \mathcal{X}$ and $\{y_i\}_{i=1}^N \subset \mathcal{Y}$, for an arbitrary $z \in \mathcal{H}_Y$, the following matrix inequality holds:*

$$\left(k_X(x_i, x_j)\langle \psi_Y(y_i), z\rangle_{\mathcal{H}_Y}\langle \psi_Y(y_j), z\rangle_{\mathcal{H}_Y}\right)_{N\times N} \preccurlyeq \left(k_X(x_i, x_j)k_Y(y_i, y_j)\langle z, z\rangle_{\mathcal{H}_Y}\right)_{N\times N}.$$

Proof Since $k_X(\cdot, \cdot)$ is a positive semi-definite kernel, following the conclusion given in Proposition 4 in [17], we only need to prove

$$\left(k_Y(y_i, y_j)\langle z, z\rangle_{\mathcal{H}_Y} - \langle \psi_Y(y_i), z\rangle_{\mathcal{H}_Y}\langle \psi_Y(y_j), z\rangle_{\mathcal{H}_Y}\right)_{N\times N}$$

is positive semi-definite, which means given any $\{\alpha_i\}_{i=1}^N \subset \mathbb{R}$, we need to prove

$$\sum_{i,j=1}^{N} \alpha_i\alpha_j \left(k_Y(y_i, y_j)\langle z, z\rangle_{\mathcal{H}_Y} - \langle \psi_Y(y_i), z\rangle_{\mathcal{H}_Y}\langle \psi_Y(y_j), z\rangle_{\mathcal{H}_Y}\right) \geqslant 0.$$

Since

$$\begin{aligned}\sum_{i,j=1}^{N} \alpha_i\alpha_j k_Y(y_i, y_j)\langle z, z\rangle_{\mathcal{H}_Y} &= \sum_{i,j=1}^{N} \alpha_i\alpha_j \langle \psi_Y(y_i), \psi_Y(y_j)\rangle_{\mathcal{H}_Y}\langle z, z\rangle_{\mathcal{H}_Y} \\ &= \left\langle \sum_{i=1}^{N} \alpha_i\psi_Y(y_i), \sum_{i=1}^{N} \alpha_i\psi_Y(y_i)\right\rangle_{\mathcal{H}_Y} \langle z, z\rangle_{\mathcal{H}_Y},\end{aligned}$$

and

$$\sum_{i,j=1}^{N} \alpha_i\alpha_j\langle \psi_Y(y_i), z\rangle_{\mathcal{H}_Y}\langle \psi_Y(y_j), z\rangle_{\mathcal{H}_Y} = \left(\langle \sum_{i=1}^{N} \alpha_i\psi_Y(y_i), z\rangle_{\mathcal{H}_Y}\right)^2,$$

according to the Cauchy inequality, we get the conclusion.

We prove Theorem 2 on the basis of Lemma 1, Lemma 2, and Lemma 3.

Proof Given a function $k(x, y)$ in $\mathcal{K}$, there is a sequence $\{k_N(x, y)\}$ in $\mathcal{K}$ such that

$$k_N(x, y) = \sum_{i=1}^{N} \alpha_i g(x, y) k_X(x, x_i) k_Y(y, y_i) g(x_i, y_i) \quad k(x, y) = \lim_{N\to\infty} k_N(x, y).$$

We need to prove that $k(x, y)$ is an asymmetric kernel.

Define $\tilde{k}_N(x, y) = \sum_{i=1}^{N} \alpha_i k_X(x, x_i) k_Y(y, y_i) g(x_i, y_i)$, we have $k(x, y) = \lim_{N\to\infty} k_N(x, y) = \lim_{N\to\infty} \tilde{k}_N(x, y) g(x, y) = \tilde{k}(x, y) g(x, y)$, where $\tilde{k}(x, y) = \lim_{N\to\infty} \tilde{k}_N(x, y)$.

From the proof of Lemma 2, we know $\tilde{k}_N(x, y) = \langle \varphi_X(x), \varphi_Y^N(y) \rangle_{\mathcal{H}}$, where $\mathcal{H}$ is a Hilbert space determined by k_X, $\varphi_Y^N = \gamma_N \circ \psi_Y$, and $\gamma_N = \sum_{i=1}^{N} \alpha_i g(x_i, y_i) \psi_X(x_i) \psi_Y^T(y_i)$. Here $\psi_X : \mathcal{X} \to \mathcal{H}_X$ and $\psi_Y : \mathcal{Y} \to \mathcal{H}_Y$ are feature mappings for k_X and k_Y, respectively.

Next, we prove $\{\gamma_N\}$ is a Cauchy sequence. $\forall z \in \mathcal{H}_Y$,

$$\|\gamma_N(z)\|_{\mathcal{H}_X}^2 = \sum_{i,j=1}^{N} \alpha_i \alpha_j g(x_i, y_i) g(x_j, y_j) k_X(x_i, x_j) \langle \psi_Y(y_i), z \rangle_{\mathcal{H}_Y} \langle \psi_Y(y_j), z \rangle_{\mathcal{H}_Y}.$$

According to Lemma 3, we have

$$\begin{aligned}
&\sum_{i,j=1}^{N} \alpha_i \alpha_j g(x_i, y_i) g(x_j, y_j) k_X(x_i, x_j) \langle \psi_Y(y_i), z \rangle_{\mathcal{H}_Y} \langle \psi_Y(y_j), z \rangle_{\mathcal{H}_Y} \\
\leqslant &\sum_{i,j=1}^{N} \alpha_i \alpha_j g(x_i, y_i) g(x_j, y_j) k_X(x_i, x_j) k_Y(y_i, y_j) \langle z, z \rangle_{\mathcal{H}_Y} \\
= &\sum_{i,j=1}^{N} \alpha_i \alpha_j g(x_i, y_i) g(x_j, y_j) k_X(x_i, x_j) k_Y(y_i, y_j) \parallel z \parallel_{\mathcal{H}_Y}^2 .
\end{aligned}$$

Therefore,

$$\parallel \gamma_N \parallel^2 \leqslant \sum_{i,j=1}^{N} \alpha_i \alpha_j g(x_i, y_i) g(x_j, y_j) k_X(x_i, x_j) k_Y(y_i, y_j).$$

Note that $\sum_{i,j=1}^{N} \alpha_i \alpha_j g(x_i, y_i) g(x_j, y_j) k_X(x_i, x_j) k_Y(y_i, y_j)$ is just the square of the norm of $k_N(x, y)$ in $\mathcal{K}$. Given the fact that $\{k_N(x, y)\}$ is a Cauchy sequence in $\mathcal{K}$, we conclude that $\{\gamma_N\}$ is also a Cauchy sequence. Then, there exists a linear operator γ which satisfies $\gamma = \lim_{N\to\infty} \gamma_N$ and $\tilde{k}(x, y) = \langle \varphi_X(x), \varphi_Y(y) \rangle$, where $\varphi_Y = \gamma \circ \psi_Y$. Thus, $k(x, y) = g(x, y)\tilde{k}(x, y)$ is an asymmetric kernel and therefore $\mathcal{K} \subset \mathcal{A}$.

C. Proof of Theorem 3

To prove the theorem, first we need to prove $\bar{k}_P((x, y), (x', y'))$ is a kernel. According to Theorem 1, we know that for any $i \in \mathbb{Z}^+$, $c_i g(x, y) \left(k_X(x, x') k_Y(y, y')\right)^i g(x', y')$ is a kernel (considering $\sqrt{c_i} k_X^i$ and $\sqrt{c_i} k_Y^i$ as 'k_X' and 'k_Y' in that theorem, respectively). Since the summation of kernels is also a kernel, we know that $\bar{k}_P((x, y), (x', y'))$ is a kernel.

Second, given $g(x, y)$ is an asymmetric kernel, suppose $\mathcal{K}_P$ is the reproducing kernel Hilbert space generated by $\bar{k}_P$. Since $\bar{k}_P$ is a kernel, we only need to prove that $\mathcal{K}_P \subset \mathcal{A} = \{k : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} | k(x, y) = \langle \varphi_X(x), \varphi_Y(y) \rangle\}$.

To prove the theorem, we need the following lemma.

Lemma 4. *Suppose that $g(x,y)$ is an asymmetric kernel, and k_X and k_Y are Mercer kernels. Given any finite example set $\{(x_j,y_j)\}_{j=1}^N \subset \mathcal{X}\times\mathcal{Y}$, and any $\{\alpha_j\}_{j=1}^N \subset \mathbb{R}$, $\sum_{j=1}^N \alpha_j \bar{k}_P((x,y),(x_j,y_j))$ is an asymmetric kernel.*

Proof

$$\begin{aligned}\sum_{j=1}^N \alpha_j \bar{k}_P((x,y),(x_j,y_j)) &= \sum_{j=1}^N \alpha_j g(x,y) \sum_{i=0}^{\infty} c_i \left(k_X(x,x_j)k_Y(y,y_j)\right)^i g(x_j,y_j) \\ &= g(x,y)\sum_{j=1}^N \alpha_j \tilde{\bar{k}}_P((x,y),(x_j,y_j)),\end{aligned}$$

where

$$\tilde{\bar{k}}_P((x,y),(x_j,y_j)) = \sum_{i=0}^{\infty} c_i \left(k_X(x,x_j)k_Y(y,y_j)\right)^i g(x_j,y_j).$$

Since $g(x,y)$ is an asymmetric kernel, according to Lemma 1, to prove $\sum_{j=1}^N \alpha_j \bar{k}_P((x,y),(x_j,y_j))$ is an asymmetric kernel, we only need to show that $\sum_{j=1}^N \alpha_j \tilde{\bar{k}}_P((x,y),(x_j,y_j))$ is an asymmetric kernel.

For any $i \geqslant 0$, $i \in \mathbb{Z}^+$, since $\sqrt{c_i}k_X^i(x,x')$ and $\sqrt{c_i}k_Y^i(y,y')$ are both Mercer kernels, we obtain

$$\sqrt{c_i}k_X^i(x,x') = \langle \psi_X^i(x), \psi_X^i(x')\rangle_{\mathcal{H}_x^i}$$

and

$$\sqrt{c_i}k_Y^i(y,y') = \langle \psi_Y^i(y), \psi_Y^i(y')\rangle_{\mathcal{H}_Y^i},$$

where $\psi_X^i(\cdot) : \mathcal{X} \to \mathcal{H}_X^i$ and $\psi_Y^i(\cdot) : \mathcal{Y} \to \mathcal{H}_Y^i$ are feature mappings, and $\mathcal{H}_X^i$ and $\mathcal{H}_Y^i$ are Hilbert spaces with respect to ψ_X^i and ψ_Y^i, respectively.

Let $\mathcal{H}_i = \mathcal{H}_X^i$, $\varphi_X^i(x) = \psi_X^i(x)$, and $\varphi_{YN}^i(y) = \sum_{j=1}^N \alpha_j g(x_j,y_j)\psi_X^i(x_j){\psi_Y^i}^\top(y_j)\psi_Y^i(y)$. Note that $\varphi_{YN}^i = \gamma_N^i \circ \psi_Y^i$, where $\gamma_N^i = \sum_{j=1}^N \alpha_j g(x_j,y_j)\psi_X^i(x_j){\psi_Y^i}^\top(y_j)$ is a linear operator from $\mathcal{H}_Y^i$ to $\mathcal{H}_X^i = \mathcal{H}_i$. Thus, we have

$$\sum_{j=1}^N \alpha_j c_i (k_X(x,x_j)k_Y(y,y_j))^i g(x_j,y_j) = \langle \varphi_X^i(x), \varphi_{YN}^i(y)\rangle_{\mathcal{H}_i}.$$

Let $\mathcal{H} = \mathcal{H}_0 \times \mathcal{H}_1 \times \cdots \mathcal{H}_k \times \cdots$,

$$\begin{aligned}\varphi_X :&\mathcal{X} \to \mathcal{H} \\ &x \mapsto (\varphi_X^0(x), \varphi_X^1(x), \cdots, \varphi_X^k(x), \cdots),\end{aligned}$$

and

$$\begin{aligned}\varphi_{YN} :&\mathcal{Y} \to \mathcal{H} \\ &y \mapsto (\varphi_{YN}^0(y), \varphi_{YN}^1(y), \cdots, \varphi_{YN}^k(y), \cdots),\end{aligned}$$

we have

$$\begin{aligned}\sum_{j=1}^{N}\alpha_j\tilde{\bar{k}}_P((x,y),(x_j,y_j)) &= \sum_{j=1}^{N}\alpha_j\sum_{i=0}^{\infty}c_i(k_X(x,x_j)k_Y(y,y_j))^i g(x_j,y_j)\\ &= \sum_{i=0}^{\infty}\sum_{j=1}^{N}\alpha_j c_i(k_X(x,x_j)k_Y(y,y_j))^i g(x_j,y_j)\\ &= \sum_{i=0}^{\infty}\langle\varphi_X^i(x),\varphi_{YN}^i(y)\rangle_{\mathcal{H}_i}\\ &= \langle\varphi_X(x),\varphi_{YN}(y)\rangle_{\mathcal{H}},\end{aligned}$$

where the inner product in $\mathcal{H}$ is naturally defined as $\sum_{i=0}^{\infty}\langle\cdot,\cdot\rangle_{\mathcal{H}_i}$. Note that $\sum_{i=0}^{\infty}\langle\cdot,\cdot\rangle_{\mathcal{H}_i}$ can be defined only when $(z_0,\cdots,z_k,\cdots)\in\mathcal{H}$, $\sum_{i=0}^{\infty}\langle z_i,z_i\rangle_{\mathcal{H}_i}<\infty$. Obviously, for any $x\in\mathcal{X}$ and $y\in\mathcal{Y}$, $\varphi_X(x)$ and $\varphi_{YN}(y)$ satisfy this condition.

We prove $\bar{k}_P$ is a hyper asymmetric kernel on the basis of above lemma.

Proof Given a function $k(x,y)$ in $\mathcal{K}_P$, there is a sequence $\{k_N(x,y)\}$ in $\mathcal{K}_P$ such that

$$\begin{aligned}k_N(x,y) &= \sum_{j=1}^{N}\alpha_j\bar{k}_P((x,y),(x_j,y_j))\\ &= \sum_{j=1}^{N}\alpha_j g(x,y)\sum_{i=0}^{\infty}c_i\left(k_X(x,x_j)k_Y(y,y_j)\right)^i g(x_j,y_j)\\ &= g(x,y)\sum_{j=1}^{N}\alpha_j\tilde{\bar{k}}_P((x,y),(x_j,y_j));\\ k(x,y) &= \lim_{N\to\infty}k_N(x,y),\end{aligned}$$

where $\tilde{\bar{k}}_P((x,y),(x_j,y_j)) = \sum_{i=0}^{\infty}c_i\left(k_X(x,x_j)k_Y(y,y_j)\right)^i g(x_j,y_j)$.

We try to prove that $k(x,y)$ is an asymmetric kernel. Let $\tilde{k}_N(x,y)=\sum_{j=1}^{N}\alpha_j\tilde{\bar{k}}_P((x,y),(x_j,y_j))$, $k(x,y)=\lim_{N\to\infty}k_N(x,y)=\lim_{N\to\infty}\tilde{k}_N(x,y)g(x,y)=\tilde{k}(x,y)g(x,y)$, where $\tilde{k}(x,y)=\lim_{N\to\infty}\tilde{k}_N(x,y)$.

According to Lemma 1, we only need to prove that $\tilde{k}(x,y)$ is an asymmetric kernel. From the proof of Lemma 4, we know $\tilde{k}_N(x,y)=\langle\varphi_X(x),\varphi_{YN}(y)\rangle_{\mathcal{H}}$, where $\mathcal{H}$ is a Hilbert space determined by $\{\sqrt{c_i}k_X^i\}_{i=0}^{\infty}$.

$$\varphi_{YN}(y)=(\varphi_{YN}^0(y),\varphi_{YN}^1(y),\cdots,\varphi_{YN}^k(y),\cdots),$$

and we define $\mathcal{H}_Y=\mathcal{H}_Y^0\times\cdots\mathcal{H}_Y^k\times\cdots$, $\mathcal{H}_X=\mathcal{H}_X^0\times\cdots\mathcal{H}_X^k\times\cdots$, and

$$\begin{aligned}\gamma_N:&\mathcal{H}_Y\to\mathcal{H}_X\\ &z=(z_0,\cdots,z_k,\cdots)\mapsto(\gamma_N^0(z_0),\cdots,\gamma_N^k(z_k),\cdots),\end{aligned}$$

where $\mathcal{H}_X^i$ and $\mathcal{H}_Y^i$ are the Hilbert spaces with respect to feature mappings $\psi_X^i(\cdot)$ and $\psi_Y^i(\cdot)$ defined by Mercer kernels $\sqrt{c_i}k_X^i$ and $\sqrt{c_i}k_Y^i$, respectively, $\langle\cdot,\cdot\rangle_{\mathcal{H}_Y^i}$ is the inner product defined in $\mathcal{H}_Y^i$, and $\gamma_N^k(z_k)=\sum_{j=1}^{N}\alpha_j g(x_j,y_j)\psi_X^k(x_j)\langle\psi_Y^k(y_j),z_k\rangle_{\mathcal{H}_Y^k}$. The inner products for $\mathcal{H}_X=\mathcal{H}_X^0\times\cdots\mathcal{H}_X^k\times\cdots$

and $\mathcal{H}_Y = \mathcal{H}_Y^0 \times \cdots \mathcal{H}_Y^k \times \cdots$ are naturally defined as $\sum_{i=0}^{\infty}\langle\cdot,\cdot\rangle_{\mathcal{H}_X^i}$ and $\sum_{i=0}^{\infty}\langle\cdot,\cdot\rangle_{\mathcal{H}_Y^i}$, respectively. Note that to make the inner products well defined, we require that input $(z_0, \cdots, z_k, \cdots)$ satisfies $\sum_{i=0}^{\infty}\langle z_i, z_i\rangle_{\mathcal{H}_Y^i} < \infty$. From the following proof, we will see that this condition will guarantee that $\sum_{i=0}^{\infty}\langle\gamma_N^i(z_i), \gamma_N^i(z_i)\rangle_{\mathcal{H}_X^i} < \infty$. Thus, γ_N is well defined.

Then the key point we need to prove is that $\{\gamma_N\}$ is a Cauchy sequence. $\forall z \in \mathcal{H}_Y$, $\|z\|_{\mathcal{H}_Y} < \infty$,

$$\| \gamma_N(z) \|^2_{\mathcal{H}_X} = \sum_{i=0}^{\infty}\sum_{k,j=1}^{N} \alpha_k\alpha_j g(x_k, y_k)g(x_j, y_j)\sqrt{c_i}k_X^i(x_k, x_j)\langle\psi_Y^i(y_k), z_i\rangle_{\mathcal{H}_Y^i}\langle\psi_Y^i(y_j), z_i\rangle_{\mathcal{H}_Y^i}.$$

Using the conclusion given by Lemma 3, we have

$$\begin{aligned}
&\sum_{i=0}^{\infty}\sum_{k,j=1}^{N} \alpha_k\alpha_j g(x_k, y_k)g(x_j, y_j)\sqrt{c_i}k_X^i(x_k, x_j)\langle\psi_Y^i(y_k), z_i\rangle_{\mathcal{H}_Y^i}\langle\psi_Y^i(y_j), z_i\rangle_{\mathcal{H}_Y^i} \\
\leqslant &\sum_{i=0}^{\infty}\sum_{k,j=1}^{N} \alpha_k\alpha_j g(x_k, y_k)g(x_j, y_j)c_i(k_X(x_k, x_j)k_Y(y_k, y_j))^i\langle z_i, z_i\rangle_{\mathcal{H}_Y^i} \\
\leqslant &\left(\sum_{i=0}^{\infty}\sum_{k,j=1}^{N} \alpha_k\alpha_j g(x_k, y_k)g(x_j, y_j)c_i(k_X(x_k, x_j)k_Y(y_k, y_j))^i\right)\left(\sum_{i=0}^{\infty}\langle z_i, z_i\rangle_{\mathcal{H}_Y^i}\right).
\end{aligned}$$

Thus,

$$\| \gamma_N(z) \|^2_{\mathcal{H}_X} \leqslant \sum_{i=0}^{\infty}\sum_{k,j=1}^{N} \alpha_k\alpha_j g(x_k, y_k)g(x_j, y_j)c_i(k_X(x_k, x_j)k_Y(y_k, y_j))^i \| z \|^2_{\mathcal{H}_Y} .$$

So

$$\begin{aligned}
\| \gamma_N \|^2 &\leqslant \sum_{i=0}^{\infty}\sum_{k,j=1}^{N} \alpha_k\alpha_j g(x_k, y_k)g(x_j, y_j)c_i(k_X(x_k, x_j)k_Y(y_k, y_j))^i \\
&= \sum_{k,j=1}^{N} \alpha_k\alpha_j \bar{k}_P((x_k, y_k), (x_j, y_j)).
\end{aligned}$$

Note that $\sum_{k,j=1}^{N} \alpha_k\alpha_j \bar{k}_P((x_k, y_k), (x_j, y_j))$ is just the square of the norm of $k_N(x, y)$ in $\mathcal{K}_P$. From the fact that $\{k_N(x, y)\}$ is a Cauchy sequence in $\mathcal{K}_P$, we know that $\{\gamma_N\}$ is also a Cauchy sequence. Then there is a linear operator γ which satisfies $\gamma = \lim_{N\to\infty}\gamma_N$ and $\tilde{k}(x, y) = \langle\varphi_X(x), \varphi_Y(y)\rangle_{\mathcal{H}}$, where φ_Y is given by

$$(\gamma^0 \circ \psi_Y^0, \cdots, \gamma^k \circ \psi_Y^k, \dots).$$

D. Proof of Theorem 4

We first prove that $\bar{k}_M((x, y), (x', y'))$ is a kernel. Using the conclusion given by Theorem 1, we know that $\forall i$, $c_i g(x, y)k_i^X(x, x')k_i^Y(y, y')g(x', y')$ is a kernel (considering $\sqrt{c_i}k_i^X$ and $\sqrt{c_i}k_i^Y$ as 'k_X' and 'k_Y' in that theorem, respectively). Since summation of kernels is also a kernel, we obtain our conclusion that $\bar{k}_M((x, y), (x', y'))$ is a kernel on $(\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y})$.

Let $\mathcal{K}_M$ be the reproducing kernel Hilbert space generated by $\bar{k}_M$, we try to prove that $\mathcal{K}_M$ is a subset of $\mathcal{A} = \{k : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} | k(x, y) = \langle\varphi_X(x), \varphi_Y(y)\rangle\}$.

To prove the theorem, we need the following lemma.

Lemma 5. *Suppose that $g(x,y)$ is an asymmetric kernel, and k_X and k_Y are Mercer kernels. Given any $\{(x_j,y_j)\}_{j=1}^N \subset \mathcal{X}\times\mathcal{Y}$, and any $\{\alpha_j\}_{j=1}^N \subset \mathbb{R}$, $\sum_{j=1}^N \alpha_j \bar{k}_M((x,y),(x_i,y_i))$ is an asymmetric kernel.*

Proof

$$\begin{aligned}\sum_{j=1}^N \alpha_j \bar{k}_M((x,y),(x_j,y_j)) &= \sum_{j=1}^N \alpha_j g(x,y) \sum_{i=1}^n c_i k_i^X(x,x_j) k_i^Y(y,y_j) g(x_j,y_j)\\ &= g(x,y) \sum_{j=1}^N \alpha_j \tilde{\bar{k}}_M((x,y),(x_j,y_j)).\end{aligned}$$

Since $g(x,y)$ is an asymmetric kernel, according to Lemma 1, to prove $\sum_{j=1}^N \alpha_j \bar{k}_M((x,y),(x_j,y_j))$ is an asymmetric kernel, we only have to prove that $\sum_{j=1}^N \alpha_j \tilde{\bar{k}}_M((x,y),(x_j,y_j))$ is an asymmetric kernel.

For any $0 \leqslant i \leqslant n, i \in \mathbb{Z}$, since $\sqrt{c_i}k_i^X(x,x')$ and $\sqrt{c_i}k_i^Y(y,y')$ are both Mercer kernels, we obtain

$$\sqrt{c_i}k_i^X(x,x') = \langle \psi_X^i(x), \psi_X^i(x')\rangle_{\mathcal{H}_X^i}, \qquad \sqrt{c_i}k_i^Y(y,y') = \langle \psi_Y^i(y), \psi_Y^i(y')\rangle_{\mathcal{H}_Y^i},$$

where $\psi_X^i(\cdot) : \mathcal{X} \to \mathcal{H}_X^i$ and $\psi_Y^i(\cdot) : \mathcal{Y} \to \mathcal{H}_Y^i$ are feature mappings, and $\mathcal{H}_X^i$ and $\mathcal{H}_Y^i$ are Hilbert spaces with respect to ψ_X^i and ψ_Y^i, respectively.

Let $\mathcal{H}_i = \mathcal{H}_X^i$, $\varphi_X^i(x) = \psi_X^i(x)$, and $\varphi_{YN}^i(y) = \sum_{j=1}^N \alpha_j g(x_j,y_j)\psi_X^i(x_j){\psi_Y^i}^\top(y_j)\psi_Y^i(y)$. Note that $\varphi_{YN}^i = \gamma_N^i \circ \psi_Y^i$, where $\gamma_N^i = \sum_{j=1}^N \alpha_j g(x_j,y_j)\psi_X^i(x_j){\psi_Y^i}^\top(y_j)$ is a linear operator from $\mathcal{H}_Y^i$ to $\mathcal{H}_X^i = \mathcal{H}_i$. Thus, we have

$$\sum_{j=1}^N \alpha_j c_i k_i^X(x,x_j)k_i^Y(y,y_j)g(x_j,y_j) = \langle \varphi_X^i(x), \varphi_{YN}^i(y)\rangle_{\mathcal{H}_i}.$$

Let $\mathcal{H} = \mathcal{H}_1 \times \mathcal{H}_2 \times \cdots \mathcal{H}_n$,

$$\begin{aligned}\varphi_X(x) :&\mathcal{X} \to \mathcal{H}\\ &x \mapsto (\varphi_X^1(x), \varphi_X^2(x), \cdots, \varphi_X^n(x)),\end{aligned}$$

and

$$\begin{aligned}\varphi_{YN}(y) :&\mathcal{Y} \to \mathcal{H}\\ &y \mapsto (\varphi_{YN}^1(y), \varphi_{YN}^2(y), \cdots, \varphi_{YN}^n(y)),\end{aligned}$$

we have

$$\begin{aligned}\sum_{j=1}^N \alpha_j \tilde{\bar{k}}_M((x,y),(x_j,y_j)) &= \sum_{j=1}^N \alpha_j \sum_{i=1}^n c_i k_i^X(x,x_j)k_i^Y(y,y_j)g(x_j,y_j)\\ &= \sum_{i=1}^n \sum_{j=1}^N \alpha_j c_i k_i^X(x,x_j)k_i^Y(y,y_j)g(x_j,y_j)\\ &= \sum_{i=1}^n \langle \varphi_X^i(x), \varphi_{YN}^i(y)\rangle_{\mathcal{H}_i}\\ &= \langle \varphi_X(x), \varphi_{YN}(y)\rangle_{\mathcal{H}},\end{aligned}$$

where the inner product in $\mathcal{H}$ is naturally defined as $\sum_{i=1}^{n}\langle\cdot,\cdot\rangle_{\mathcal{H}_i}$.

We prove that $\bar{k}_M$ is a hyper asymmetric kernel on the basis of Lemma 5.

Proof Given a function $k(x,y)$ in $\mathcal{K}_M$, there is a sequence $\{k_N(x,y)\}$ in $\mathcal{K}_M$ such that

$$\begin{aligned}
k_N(x,y) &= \sum_{j=1}^{N}\alpha_j\bar{k}_M((x,y),(x_j,y_j)) \\
&= \sum_{j=1}^{N}\alpha_j g(x,y)\sum_{i=1}^{n}c_i k_i^X(x,x_j)k_i^Y(y,y_j)g(x_j,y_j) \\
&= g(x,y)\sum_{j=1}^{N}\alpha_j\tilde{\bar{k}}_M((x,y),(x_j,y_j)); \\
k(x,y) &= \lim_{N\to\infty}k_N(x,y).
\end{aligned}$$

We try to prove that $k(x,y)$ is an asymmetric kernel. Let $\tilde{k}_N(x,y) = \sum_{j=1}^{N}\alpha_j\tilde{\bar{k}}_M((x,y),(x_j,y_j))$, $k(x,y) = \lim_{N\to\infty}k_N(x,y) = \lim_{N\to\infty}\tilde{k}_N(x,y)g(x,y) = \tilde{k}(x,y)g(x,y)$, where $\tilde{k}(x,y) = \lim_{N\to\infty}\tilde{k}_N(x,y)$.

According to Lemma 1, we only have to prove that $\tilde{k}(x,y)$ is an asymmetric kernel. From the proof of Lemma 5, we know $\tilde{k}_N(x,y) = \langle\varphi_X(x),\varphi_{YN}(y)\rangle_{\mathcal{H}}$, where $\mathcal{H} = \mathcal{H}_1\times\mathcal{H}_2\times\cdots\mathcal{H}_n$ is a Hilbert space and $\mathcal{H}_i$ is the Hilbert space of the feature mapping of $\sqrt{c_i}k_i^X(\cdot,\cdot)$.

$$\varphi_{YN}(y) = (\varphi_{NY}^1(y),\varphi_{NY}^2(y),\cdots,\varphi_{NY}^n(y)),$$

and we define $\mathcal{H}_Y = \mathcal{H}_Y^1\times\cdots\mathcal{H}_Y^n$, $\mathcal{H}_X = \mathcal{H}_X^1\times\cdots\mathcal{H}_X^n$, and

$$\begin{aligned}
\gamma_N :&\mathcal{H}_Y\to\mathcal{H}_X \\
&z = (z_1,\cdots,z_n)\mapsto(\gamma_N^1(z_1),\cdots,\gamma_N^n(z_n)),
\end{aligned}$$

where $\mathcal{H}_X^i$ and $\mathcal{H}_Y^i$ are the Hilbert spaces with respect to feature mappings $\psi_X^i(\cdot)$ and $\psi_Y^i(\cdot)$ of Mercer kernels $\sqrt{c_i}k_i^X$ and $\sqrt{c_i}k_i^Y$, respectively, $\langle\cdot,\cdot\rangle_{\mathcal{H}_Y^i}$ is the inner product defined in $\mathcal{H}_Y^i$, and $\gamma_N^i(z_i) = \sum_{j=1}^{N}\alpha_j g(x_j,y_j)\psi_X^i(x_j)\langle\psi_Y^i(y_j),z_i\rangle_{\mathcal{H}_Y^i}$.

Then the key point we need to prove is that $\{\gamma_N\}$ is a Cauchy sequence. $\forall z\in\mathcal{H}_Y$,

$$\|\gamma_N(z)\|_{\mathcal{H}_X}^2 = \sum_{i=1}^{n}\sum_{k,j=1}^{N}\alpha_k\alpha_j g(x_k,y_k)g(x_j,y_j)\sqrt{c_i}k_i^X(x_k,x_j)\langle\psi_Y^i(y_k),z_i\rangle_{\mathcal{H}_Y^i}\langle\psi_Y^i(y_j),z_i\rangle_{\mathcal{H}_Y^i}.$$

Using the conclusion given by Lemma 3, we have

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{k,j=1}^{N}\alpha_k\alpha_j g(x_k,y_k)g(x_j,y_j)\sqrt{c_i}k_i^X(x_k,x_j)\langle\psi_Y^i(y_k),z_i\rangle_{\mathcal{H}_Y^i}\langle\psi_Y^i(y_j),z_i\rangle_{\mathcal{H}_Y^i} \\
\leqslant&\sum_{i=1}^{n}\sum_{k,j=1}^{N}\alpha_k\alpha_j g(x_k,y_k)g(x_j,y_j)c_i k_i^X(x_k,x_j)k_i^Y(y_k,y_j)\langle z_i,z_i\rangle_{\mathcal{H}_Y^i} \\
\leqslant&\left(\sum_{i=1}^{n}\sum_{k,j=1}^{N}\alpha_k\alpha_j g(x_k,y_k)g(x_j,y_j)c_i k_i^X(x_k,x_j)k_i^Y(y_k,y_j)\right)\left(\sum_{i=1}^{n}\langle z_i,z_i\rangle_{\mathcal{H}_Y^i}\right).
\end{aligned}$$

Thus,

$$\| \gamma_N(z) \|^2_{\mathcal{H}_X} \leqslant \sum_{i=1}^{n} \sum_{k,j=1}^{N} \alpha_k \alpha_j g(x_k, y_k) g(x_j, y_j) c_i k_i^X(x_k, x_j) k_i^Y(y_k, y_j) \| z \|^2_{\mathcal{H}_Y} .$$

So

$$\| \gamma_N \|^2 \leqslant \sum_{i=1}^{n} \sum_{k,j=1}^{N} \alpha_k \alpha_j g(x_k, y_k) g(x_j, y_j) c_i k_i^X(x_k, x_j) k_i^Y(y_k, y_j) = \sum_{k,j=1}^{N} \alpha_k \alpha_j \bar{k}_M((x_k, y_k), (x_j, y_j)).$$

Note that $\sum_{k,j=1}^{N} \alpha_k \alpha_j \bar{k}_M((x_k, y_k), (x_j, y_j))$ is just the square of the norm of $k_N(x, y)$ in $\mathcal{K}_M$. From the fact that $\{k_N(x, y)\}$ is a Cauchy sequence in $\mathcal{K}_M$, we know that $\{\gamma_N\}$ is also a Cauchy sequence. Then there is a linear operator γ which satisfies $\gamma = \lim_{N\to\infty} \gamma_N$ and $\tilde{k}(x, y) = \langle \varphi_X(x), \varphi_Y(y) \rangle_{\mathcal{H}}$, where φ_Y is given by

$$(\gamma^1 \circ \psi_Y^1, \cdots, \gamma^n \circ \psi_Y^n).$$