Near-Orthogonality Regularization in Kernel Methods

Pengtao Xie[*†], **Barnabas Poczos**[*] **and Eric P. Xing**[†]
[*]Machine Learning Department, Carnegie Mellon University
[†]Petuum Inc.
{pengtaox,bapoczos}@cs.cmu.edu, eric.xing@petuum.com

Abstract

Kernel methods perform nonlinear learning in high-dimensional reproducing kernel Hilbert spaces (RKHSs). Even though their large model-capacity leads to high representational power, it also incurs substantial risk of overfitting. To alleviate this problem, we propose a new regularization approach, *near-orthogonality regularization*, which encourages the RKHS functions to be close to being orthogonal. This effectively imposes a structural constraint over the function space, which reduces model complexity and can improve generalization performance. Besides, encouraging orthogonality reduces the redundancy among functions, which hence can reduce model size without compromising modeling power and better capture infrequent patterns in the data. Here, we define a family of orthogonality-promoting regularizers by encouraging the Gram matrix of the RKHS functions to be close to an identity matrix where the closeness is measured by Bregman matrix divergences. We apply these regularizers to two kernel methods, and develop an efficient ADMM-based algorithm to solve the regularized optimization problems. We analyze how near-orthogonality affects the generalization performance of kernel methods. Our results suggest that the closer the functions are to being orthogonal, the smaller the generalization error is. Experiments demonstrate the efficacy of near-orthogonality regularization in kernel methods.

1 INTRODUCTION

Kernel methods perform learning in reproducing kernel Hilbert spaces (RKHSs) of functions (Schölkopf and Smola, 2002). The RKHS represents a high-dimensional feature space that can capture nonlinear patterns in the lower-dimensional observed data. This Hilbert space is associated with a kernel function k, and the inner product in the RKHS can be implicitly computed by evaluating k in the lower-dimensional input space (known as *kernel trick*). Well-established kernel methods include support vector machine (Schölkopf and Smola, 2002), kernel principal component analysis (Schölkopf et al., 1997), kernel independent component analysis (Bach and Jordan, 2002), to name a few.

One key ingredient in kernel methods is regularization, which reduces overfitting by controlling the complexity of the RKHS functions (Schölkopf and Smola, 2002; Micchelli and Pontil, 2005). Regularizers proposed previously such as RKHS norm, derivatives, green functions, and splines mostly focus on encouraging a small norm (Micchelli and Pontil, 2005) and smoothness of functions (Schölkopf and Smola, 2002). Notably, the most widely-used regularizer is the squared RKHS norm.

In this work, we introduce a new regularization approach that encourages a set of RKHS functions to be close to being orthogonal, so that the correlation and redundancy between these functions can be reduced. Besides alleviating overfitting problems, these regularizers can also (1) reduce model size without sacrificing modeling power (Xie, 2015): near-orthogonal functions bear less redundancy and are highly complementary, and thus a small number of such functions can possess sufficient representational power; (2) capture infrequent latent patterns in the data (Xie et al., 2015a): without near-orthogonality regularization, the majority of RKHS functions are used to capture frequent patterns since these patterns have dominant signals in the dataset; promoting near-orthogonality among the functions can drive them to diversely "spread out", giving both infrequent and frequent patterns a fair treatment. Previously, regularizers that achieved near-orthogonality effects have been investigated in other methods, including

latent Dirichlet allocation (Zou and Adams, 2012), neural networks (Cogswell et al., 2015) and restricted Boltzmann machine (Xie et al., 2015a); however, they have not been explored in kernel methods. We aim to bridge this gap in this work.

To promote near-orthogonality among a set of RKHS functions $\{f_i\}_{i=1}^K$, we compute their Gram matrix $\mathbf{G}$ where $G_{ij} = \langle f_i, f_j \rangle$, and encourage $\mathbf{G}$ to be close to an identity matrix $\mathbf{I}$. The off-diagonal elements of $\mathbf{G}$ and $\mathbf{I}$ are $\langle f_i, f_j \rangle$ and zero, respectively. The elements on the diagonal of $\mathbf{G}$ and $\mathbf{I}$ are $\|f_i\|_{\mathcal{H}}^2$ and one, respectively. Making $\mathbf{G}$ close to $\mathbf{I}$ effectively encourages $\langle f_i, f_j \rangle$ to be close to zero and $\|f_i\|_{\mathcal{H}}$ to be close to one, which drives f_i and f_j to be close to being orthogonal. We measure the closeness between $\mathbf{G}$ and $\mathbf{I}$ using Bregman matrix divergences (Dhillon and Tropp, 2007), and define a family of near-orthogonality regularizers thereupon. We apply the proposed regularizers to two kernel methods – kernel distance metric learning (KDML) (Tsang et al., 2003; Jain et al., 2012) and kernel sparse coding (KSC) (Gao et al., 2010), and develop an optimization algorithm based on alternating direction method of multipliers (ADMM) (Boyd et al., 2011) where the RKHS functions are learned using functional gradient descent (FGD) (Dai et al., 2014). We perform analysis to show that the near-orthogonality regularization can reduce generalization error bounds. Experimental results show that the proposed near-orthogonality regularizers (1) greatly improve the generalization performance of KDML and KSC; (2) can reduce model size without sacrificing modeling power; (3) can better capture infrequent patterns in the data; and (4) outperform other orthogonality-promoting regularizers and the squared Hilbert norm.

The major contributions of this paper are:

- We propose a new regularization approach in kernel methods that encourages the RKHS functions to be close to being orthogonal.
- We define a family of near-orthogonality regularizers based on Bregman matrix divergences.
- We apply these regularizers to two kernel methods, and develop an ADMM-based algorithm to solve the regularized optimization problems.
- We analyze how the near-orthogonality regularization affects the generalization performance of kernel methods.
- Experiments demonstrate the efficacy of the proposed regularizers.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 introduces the near-orthogonality regularizers and an ADMM-based algorithm. Section 4 and 5 present the generalization error analysis and experimental results respectively. Finally, Section 6 concludes the paper.

2 RELATED WORKS

In non-kernel methods, regularizers that achieve near-orthogonality effects have been investigated. Zou and Adams (2012) employ the determinantal point process (DPP) (Kulesza and Taskar, 2012) to encourage the parallelepiped formed by a set of vectors $\{\mathbf{w}_i\}_{i=1}^K$ to have a large volume which indirectly promotes near-orthogonality among these vectors, because making the vectors close to being orthogonal can effectively enlarge the volume. A major drawback of DPP is its sensitivity to the scaling of vectors, e.g., increasing the magnitude of vectors can also enlarge the volume, but it does not promote near-orthogonality. Xie et al. (2015a) propose an angle-based regularizer, which encourages the angles between a set of vectors to have large mean and small variance, hence encouraging these vectors to become orthogonal. The angle between vector $\mathbf{w}_i$ and $\mathbf{w}_j$ is defined as $\theta_{ij} = \arccos(\frac{|\mathbf{w}_i^\top \mathbf{w}_j|}{\|\mathbf{w}_i\|_2 \|\mathbf{w}_j\|_2})$. An absolute value is used here to ensure the angles are driven to $\frac{\pi}{2}$ (orthogonal), rather than π. The absolute value makes this regularizer non-smooth, which brings in difficulty for optimization. In this paper, we aim to define new near-orthogonality regularizers to overcome the limitations of existing ones.

3 METHOD

We consider kernel methods that are parameterized by a set of RKHS functions. Examples include kernel principal component analysis (PCA) (Schölkopf et al., 1997), kernel independent component analysis (ICA) (Bach and Jordan, 2002), kernel distance metric learning (Tsang et al., 2003; Jain et al., 2012) and kernel sparse coding (Gao et al., 2010), to name a few. We propose to regularize these functions from the perspective of *near-orthogonality*, which encourages these functions to be close to being orthogonal. This can reduce the correlation and redundancy among the functions, which can potentially bring in several benefits: (1) reducing overfitting, (2) shrinking model size (the number of functions) without sacrificing modeling power, and (3) capturing infrequent patterns, as we will verify in experiments.

3.1 NEAR-ORTHOGONALITY REGULARIZERS

In this section, we first define regularizers to achieve near-orthogonality between two RKHS functions f and g. One way to encourage f and g to be close to being orthogonal is to make their inner product $\langle f, g \rangle$ in the

RKHS close to zero and their norms $\|f\|_{\mathcal{H}}$ and $\|g\|_{\mathcal{H}}$ close to one. In light of this, the near-orthogonality among a set of functions $\mathcal{F} = \{f_i\}_{i=1}^K$ can be achieved in the following manner: computing the Gram matrix $\mathbf{G}$ where $G_{ij} = \langle f_i, f_j \rangle$, then encouraging $\mathbf{G}$ to be close to an identity matrix. Off the diagonal of $\mathbf{G}$ and $\mathbf{I}$ are $\langle f_i, f_j \rangle$ and zero, respectively. On the diagonal of $\mathbf{G}$ and $\mathbf{I}$ are $\|f_i\|_{\mathcal{H}}^2$ and one, respectively. Making $\mathbf{G}$ close to $\mathbf{I}$ effectively encourages $\langle f_i, f_j \rangle$ to be close to zero and $\|f_i\|_{\mathcal{H}}$ close to one, which as a result encourages f_i and f_j to be close to being orthogonal.

Next, we discuss how to measure the closeness between $\mathbf{G}$ and $\mathbf{I}$. One straightforward way is to use the squared Frobenius norm (SFN): $\|\mathbf{G} - \mathbf{I}\|_F^2$. The SFN measures orthogonality of functions in a pairwise manner since it can be factorized into pairwise inner products: $\sum_{i=1}^K \sum_{j \neq i}^K (\langle f_i, f_j \rangle)^2 + \sum_{i=1}^K (\|f_i\|_{\mathcal{H}}^2 - 1)^2$. We conjecture[1] that measuring orthogonality in a global manner is more desirable. To achieve this goal, we resort to another measure: Bregman matrix divergence (BMD) (Dhillon and Tropp, 2007). Let $\mathbb{S}^n$ denote real, symmetric $n \times n$ matrices. Given a strictly convex, differentiable function $\phi : \mathbb{S}^n \to \mathbb{R}$, the BMD is defined as: $D_\phi(\mathbf{X}, \mathbf{Y}) = \phi(\mathbf{X}) - \phi(\mathbf{Y}) - \text{tr}((\bigtriangledown \phi(\mathbf{Y}))^\top (\mathbf{X} - \mathbf{Y}))$ where $\text{tr}(\cdot)$ denotes the trace of a matrix. Under different choices of ϕ, $D_\phi(\mathbf{X}, \mathbf{Y})$ can be specialized to several instances. Under $\phi(\mathbf{X}) = \text{tr}(\mathbf{X} \log \mathbf{X} - \mathbf{X})$, where $\log \mathbf{X}$ is the matrix logarithm, we have the von Neumann divergence (VND) (Kulis et al., 2009): $D_{vN}(\mathbf{X}, \mathbf{Y}) = \text{tr}(\mathbf{X} \log \mathbf{X} - \mathbf{X} \log \mathbf{Y} - \mathbf{X} + \mathbf{Y})$. $\phi(\mathbf{X}) = -\log\det \mathbf{X}$ results in a log-determinant divergence (LDD) (Kulis et al., 2009): $D_{ld}(\mathbf{X}, \mathbf{Y}) = \text{tr}(\mathbf{X}\mathbf{Y}^{-1}) - \log\det(\mathbf{X}\mathbf{Y}^{-1}) - n$. Interestingly, SFN is a special case of BMD when $\phi(\mathbf{X}) = \|\mathbf{X}\|_F^2$. Given the three instances of BMD, we can use them to measure the closeness between $\mathbf{G}$ and $\mathbf{I}$, and define a family of BMD regularizers (constants are dropped) to promote near-orthogonality. Under VND, the regularizer is

$$\Omega_{vN}(\mathcal{F}) = D_{vN}(\mathbf{G}, \mathbf{I}) \propto \text{tr}(\mathbf{G} \log \mathbf{G} - \mathbf{G}). \quad (1)$$

Under LDD, the regularizer is

$$\Omega_{ld}(\mathcal{F}) = D_{ld}(\mathbf{G}, \mathbf{I}) \propto \text{tr}(\mathbf{G}) - \log\det(\mathbf{G}). \quad (2)$$

Under SFN, the regularizer is

$$\Omega_{sfn}(\mathcal{F}) = D_{sfn}(\mathbf{G}, \mathbf{I}) = \|\mathbf{G} - \mathbf{I}\|_F^2. \quad (3)$$

To apply the VND and LDD regularizers, the Gram matrix $\mathbf{G}$ is required to be positive definite. In our experiments, this condition is always satisfied since VND and LDD encourage the RKHS functions to be close to being orthogonal (therefore linearly independent). Different from the SFN regularizer, VND and LDD do not admit a pairwise factorization, and hence allow one to measure orthogonality globally; this benefit will be demonstrated in experiments as shown below (Section 5). Unlike DPP (Kulesza and Taskar, 2012), LDD utilizes an additional term $\text{tr}(\mathbf{G}) = \sum_{i=1}^K \|f_i\|_{\mathcal{H}}^2$ to control the magnitude of RKHS functions, and thus avoiding DPP's sensitivity to scaling. Similarly, VND and SFN are also insensitive to scaling since they encourage $\|f\|_{\mathcal{H}}$ to be close to one. In addition, all three regularizers are smooth and amenable for optimization.

Here, we use these regularizers to encourage near-orthogonality among RKHS functions, and define BMD regularized kernel methods (BMD-KM):

$$\min_{\mathcal{F}} \quad \mathcal{L}(\mathcal{F}) + \lambda \Omega(\mathcal{F}) \quad (4)$$

where $\mathcal{L}(\mathcal{F})$ is the objective function of the kernel method, and λ is the regularization parameter. Compared to kernel PCA and ICA in which the functions are required to be strictly-orthogonal, BMD-KM can be seen as a relaxed counterpart where the functions are encouraged to be close to, but not necessarily strictly, orthogonal. As we will demonstrate in the experiments, strict-orthogonality can compromise performance in certain applications.

3.2 CASE STUDIES

In this section, we apply the BMD regularizers to two instances of kernel methods: kernel distance metric learning and kernel sparse coding.

Kernel Distance Metric Learning with BMD Regularization Distance metric learning (DML) has wide applications in classification, clustering and information retrieval (Xing et al., 2002; Davis et al., 2007; Guillaumin et al., 2009). Given data pairs labeled as similar or dissimilar, DML aims at learning a distance metric such that similar pairs would be placed close to each other and dissimilar pairs are separated apart. Kernel DML (Tsang et al., 2003; Jain et al., 2012) is equipped with K RKHS functions $\mathcal{F} = \{f_i\}_{i=1}^K$ that map a data example $\mathbf{x}$ into a vector $\mathbf{h}^{(x)}$ in a K-dimensional latent space, where $\mathbf{h}_i^{(x)} = f_i(\mathbf{x})$. Given two examples $\mathbf{x}$ and $\mathbf{y}$, their distance is defined as $d_{\mathcal{F}}(\mathbf{x}, \mathbf{y}) = \|\mathbf{h}^{(x)} - \mathbf{h}^{(y)}\|_2^2$, which is parameterized by $\mathcal{F}$. Given N training examples, $\{\mathbf{x}_n, \mathbf{y}_n, t_n\}_{n=1}^N$, where $\mathbf{x}_n$ and $\mathbf{y}_n$ are similar if the label t_n equals to 1 and dissimilar if $t_n = 0$, following (Guillaumin et al., 2009), we learn the distance metric by minimizing $\sum_{n=1}^N \log(1 + \exp((2t_n - 1) d_{\mathcal{F}}(\mathbf{x}_n, \mathbf{y}_n))$. Using $\Omega(\mathcal{F})$ to promote near-orthogonality, we obtain

[1] The conjecture is validated in experiments.

the BMD-regularized KDML (BMD-KDML) problem:

$$\min_{\mathcal{F}} \sum_{n=1}^{N} \log(1+\exp((2t_n-1)d_{\mathcal{F}}(\mathbf{x}_n,\mathbf{y}_n)) + \lambda\Omega(\mathcal{F}). \tag{5}$$

Kernel Sparse Coding with BMD Regularization Sparse coding (SC) (Olshausen and Field, 1997) is widely applied for signal processing, data reconstruction, feature learning, to name a few. To reconstruct an input data $\mathbf{x}$, SC learns a dictionary of basis $\{\mathbf{d}_i\}_{i=1}^K$ and reconstructs $\mathbf{x}$ using a sparse linear combination of the basis: $\mathbf{x} \approx \sum_{i=1}^K \alpha_i \mathbf{d}_i$, where $\{\alpha_i\}_{i=1}^K$ are the sparse coefficients. In the kernel SC (Gao et al., 2010), $\mathbf{x}$ is mapped into $k(\mathbf{x},\cdot)$ in a RKHS induced by kernel function $k(\cdot,\cdot)$ and a dictionary of RKHS functions $\mathcal{F} = \{f_i\}_{i=1}^K$ are learned to reconstruct $k(\mathbf{x},\cdot)$. Given the training data $\{\mathbf{x}_n\}_{n=1}^N$, $\mathcal{F}$ can be learned by minimizing $\frac{1}{2}\sum_{n=1}^N \|k(\mathbf{x}_n,\cdot) - \sum_{i=1}^K a_{ni} f_i\|_{\mathcal{H}}^2 + \lambda_1 \sum_{n=1}^N \|\mathbf{a}_n\|_1 + \frac{\lambda_2}{2}\sum_{i=1}^K \|f_i\|_{\mathcal{H}}^2$, where the reconstruction error is measured by the squared Hilbert norm and $\mathbf{a}_n$ are the linear coefficients. The ℓ_1 regularizer $\|\mathbf{a}_n\|_1$ is applied to encourage the coefficients to be sparse. To avoid the degenerated case where the RKHS functions are of large norm while the coefficients are close to zero, the squared Hilbert norm regularizer $\|f_i\|_{\mathcal{H}}^2$ is applied to the RKHS functions to keep their magnitude small. By adding $\Omega(\mathcal{F})$, we obtain the BMD-regularized KSC (BMD-KSC) problem:

$$\begin{aligned}\min_{\mathcal{F},\mathcal{A}} \quad & \tfrac{1}{2}\sum_{n=1}^N \|k(\mathbf{x}_n,\cdot) - \sum_{i=1}^K a_{ni} f_i\|_{\mathcal{H}}^2 \\ & + \lambda_1 \sum_{n=1}^N \|\mathbf{a}_n\|_1 + \tfrac{\lambda_2}{2}\sum_{i=1}^K \|f_i\|_{\mathcal{H}}^2 + \lambda_3\Omega(\mathcal{F})\end{aligned} \tag{6}$$

where $\mathcal{A}$ denotes all the sparse codes.

3.3 ALGORITHM

In this section, we develop an ADMM (Boyd et al., 2011) based algorithm to solve the BMD-KM problem. First, by introducing auxiliary variables $\widehat{\mathcal{F}} = \{\widehat{f}_i\}_{i=1}^K$ which are a set of RKHS functions and $\mathbf{A} \in \mathbb{R}^{K\times K}$, we rewrite the BMD-KM problem into an equivalent form that is amenable for developing ADMM-based algorithms.

$$\begin{aligned}\min_{\mathcal{F},\widehat{\mathcal{F}},\mathbf{A}} \quad & \mathcal{L}(\mathcal{F}) + \lambda D_\phi(\mathbf{A},\mathbf{I}) \\ s.t. \quad & \forall i, f_i = \widehat{f}_i \\ & \forall i,j, \langle f_i, \widehat{f}_j\rangle = A_{ij}, \langle f_i, \widehat{f}_j\rangle = A_{ji}\end{aligned} \tag{7}$$

where $\mathbf{A}$ is required to be positive definite when $D_\phi(\mathbf{A},\mathbf{I})$ is an VND or LDD regularizer. The constraints $\langle f_i, \widehat{f}_j\rangle = A_{ij}$ and $\langle f_i, \widehat{f}_j\rangle = A_{ji}$ imply that $\mathbf{A}$ is symmetric. We define augmented Lagrangian with parameter $\rho > 0$: $\mathcal{L}(\mathcal{F}) + \lambda D_\phi(\mathbf{A},\mathbf{I}) + \sum_{i=1}^K \langle g_i, f_i - \widehat{f}_i\rangle + \sum_{i=1}^K\sum_{j=1}^K (P_{ij}(\langle f_i,\widehat{f}_j\rangle - A_{ij}) + Q_{ij}(\langle f_i,\widehat{f}_j\rangle - A_{ji}) + \frac{\rho}{2}(\langle f_i,\widehat{f}_j\rangle - A_{ij})^2 + \frac{\rho}{2}(\langle f_i,\widehat{f}_j\rangle - A_{ji})^2)$, where $\mathcal{G} = \{g_i\}_{i=1}^K$ is another set of RKHS functions, and $\mathbf{P},\mathbf{Q} \in \mathbb{R}^{K\times K}$ are Lagrange multipliers. Then we minimize this Lagrangian function by alternating among $\mathcal{F}$, $\widehat{\mathcal{F}}$, $\mathcal{G}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{Q}$.

Solve A Given $\mathbf{H} \in \mathbb{R}^{K\times K}$ where $H_{ij} = \langle f_i, \hat{f}_j\rangle$, we learn $\mathbf{A}$ by minimizing $\lambda D_\phi(\mathbf{A},\mathbf{I}) - \langle \mathbf{P},\mathbf{A}\rangle - \langle \mathbf{Q}^\top,\mathbf{A}\rangle + \frac{\rho}{2}\|\mathbf{H}-\mathbf{A}\|_F^2 + \frac{\rho}{2}\|\mathbf{H}^\top - \mathbf{A}\|_F^2$, which is a convex problem. $D_\phi(\mathbf{A},\mathbf{I})$ has three cases, which we discuss separately.

When $D_\phi(\mathbf{A},\mathbf{I})$ is VND, we first perform an eigendecomposition of $\mathbf{D} = \mathbf{P} + \mathbf{Q}^\top + \rho(\mathbf{H}+\mathbf{H}^\top)$: $\mathbf{D} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Phi}^{-1}$, then the optimal solution of $\mathbf{A}$ can be obtained as $\mathbf{A} = \mathbf{\Phi}\widehat{\mathbf{\Sigma}}\mathbf{\Phi}^{-1}$ where

$$\widehat{\Sigma}_{ii} = \frac{\lambda\omega\left(\frac{\Sigma_{ii}}{\lambda} - \log\left(\frac{\lambda}{2\rho}\right)\right)}{2\rho} \tag{8}$$

and $\omega(\cdot)$ is the Wright omega function (Gorenflo et al., 2007). It can be shown that $\mathbf{A}$ is positive definite.

When $D_\phi(\mathbf{A},\mathbf{I})$ is LDD, the optimal solution is:

$$\mathbf{A} = -\tfrac{1}{2}\mathbf{B} + \tfrac{1}{2}\sqrt{\mathbf{B}^2 - 4\mathbf{C}} \tag{9}$$

where $\mathbf{B} = \frac{1}{\rho}(\lambda\mathbf{I} - \mathbf{P} - \mathbf{Q}^\top - \rho(\mathbf{H}+\mathbf{H}^\top))$ and $\mathbf{C} = -\frac{\lambda}{\rho}\mathbf{I}$. It can be verified that $\mathbf{A}$ is positive definite.

When $D_\phi(\mathbf{A},\mathbf{I})$ is SFN, the optimal solution for $\mathbf{A}$ is:

$$\mathbf{A} = (2\lambda\mathbf{I} + \mathbf{P} + \mathbf{Q}^\top + \rho(\mathbf{H}+\mathbf{H}^\top))/(2\lambda + 2\rho). \tag{10}$$

Please refer to the supplements for a detailed derivation.

Solve f_i We solve f_i by minimizing $\Upsilon = \mathcal{L}(\mathcal{F}) + \langle g_i, f_i\rangle + \sum_{j=1}^K (P_{ij}+Q_{ij})\langle f_i,\widehat{f}_j\rangle + \frac{\rho}{2}\sum_{j=1}^K((\langle f_i,\widehat{f}_j\rangle - A_{ij})^2 + (\langle f_i,\widehat{f}_j\rangle - A_{ji})^2)$. The first issue we need to address is how to represent f_i. When $f \in \mathcal{F}$ is regularized by the RKHS norm, according to the representer theorem (Schölkopf and Smola, 2002), the optimal solution f^* can be expressed as a linear combination of kernel functions evaluated at training data: $f^*(\mathbf{x}) = \sum_{n=1}^N \alpha_n k(\mathbf{x}_n,\mathbf{x})$, which we refer to as *representer theorem representation* (RTR). This endows f^* an explicit parametrization that greatly eases learning: the search space of f^* is reduced from the infinite-dimensional RKHS $\mathcal{H}$ to an N-dimensional space of coefficients $\{\alpha_n\}_{n=1}^N$. However, in Υ, due to the presence of inner products between f_i with other functions, the representer theorem does not hold and f_i does not admit a RTR form. To address this issue, we learn f_i directly using functional gradient descent (Dai et al., 2014). A *functional* $F : \mathcal{H} \to \mathbb{R}$ maps functions in $\mathcal{H}$ to real numbers. A *functional gradient* $\nabla F[f]$ is

defined implicitly as the linear term of the change in a function due to a small perturbation ϵ in its input: $F[f + \epsilon g] = F[f] + \epsilon\langle \nabla F[f], g\rangle + O(\epsilon^2)$. Of particular interest is the *evaluation functional* $F_{\mathbf{x}}[f]$ which is parameterized by an input vector $\mathbf{x}$ and evaluates f at $\mathbf{x}$: $F_{\mathbf{x}}[f] = f(\mathbf{x})$. The functional gradient of $F_{\mathbf{x}}[f]$ is $k(\mathbf{x}, \cdot)$ (Dai et al., 2014) where k is the kernel associated with the RKHS. The gradient of an inner product functional $F_g[f] = \langle f, g\rangle$ is g.

$\mathcal{L}(\mathcal{F})$ depends on a specific kernel method. Here, we consider KDML where $\mathcal{L}(\mathcal{F})$ is given in Eq.(5) while leaving the derivation of KSC to the supplements. In KDML, f_i appears in two types of functionals: evaluation functionals in $d_{\mathcal{F}}(\mathbf{x}_n, \mathbf{y}_n)$ (such as $f_i(\mathbf{x}_n)$) and inner product functionals (such as $\langle g_i, f_i\rangle$). The functional gradient of Υ is $\triangle f_i = 2\sum_{n=1}^{N} \sigma((2t_n - 1)d_{\mathcal{F}}(\mathbf{x}_n, \mathbf{y}_n))(2t_n - 1)(f_i(\mathbf{x}_n) - f_i(\mathbf{y}_n))(k(\mathbf{x}_n, \cdot) - k(\mathbf{y}_n, \cdot)) + g_i + \sum_{j=1}^{K}(P_{ij} + Q_{ij} + \rho(2\langle f_i, \widehat{f}_j\rangle - A_{ij} - A_{ji}))\widehat{f}_j$, where $\sigma(x) = 1/(1 + \exp(-x))$ is a sigmoid function. Given this functional gradient, we can perform gradient descent to update f_i until convergence: $f_i \leftarrow f_i - \eta\triangle f_i$, where η is the learning rate. In the algorithm, we initialize f_i, g_i and $\widehat{f}_j$ as zero functions. Then as will be proven in Section 3.3.1, during the algorithm execution, f_i, g_i and $\widehat{f}_j$ are all in the form of RTR. So updating f_i amounts to updating the linear coefficients in the RTR.

Solve $\widehat{f}_j$ The sub-problem defined over $\widehat{f}_j$ is (**T**): $\min_{\widehat{f}_j} \; -\langle g_j, \widehat{f}_j\rangle + \sum_{i=1}^{K}((P_{ij} + Q_{ij})\langle f_i, \widehat{f}_j\rangle + \frac{\rho}{2}((\langle f_i, \widehat{f}_j\rangle - A_{ij})^2 + (\langle f_i, \widehat{f}_j\rangle - A_{ji})^2))$ which is a convex problem. Setting the derivative of the objective function to zero, we get an equation (**E**): $(2\rho\sum_{i=1}^{K} f_i \otimes f_i)\widehat{f}_j = \sum_{i=1}^{K}(\rho(A_{ij} + A_{ji}) - (P_{ij} + Q_{ij}))f_i + g_j$, where $\otimes$ denotes the outer product in RKHS. As will be proven in Section 3.3.1, $\widehat{f}_j$, g_j and f_i are all in the form of RTR. Let $\Phi = [k(\mathbf{x}_1, \cdot), \cdots, k(\mathbf{x}_N, \cdot)]$, then $f_i = \Phi\mathbf{a}_i$, $\sum_{i=1}^{K}(\rho(A_{ij} + A_{ji}) - (P_{ij} + Q_{ij}))f_i + g_j = \Phi\mathbf{b}$, $\widehat{f}_j = \Phi\mathbf{c}$, where $\mathbf{a}_i$, $\mathbf{b}$, $\mathbf{c}$ are coefficient vectors. $\mathbf{a}_i$ and $\mathbf{b}$ are known and $\mathbf{c}$ is to be estimated. Then (**E**) can be written as $(2\rho\sum_{i=1}^{K}\mathbf{a}_i\mathbf{a}_i^\top)\Phi^\top\Phi\mathbf{c} = \mathbf{b}$, where $(\Phi^\top\Phi)_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{c} = ((2\rho\sum_{i=1}^{K}\mathbf{a}_i\mathbf{a}_i^\top)\Phi^\top\Phi)^{-1}\mathbf{b}$. In practice, inverting the $N \times N$ matrix $(2\rho\sum_{i=1}^{K}\mathbf{a}_i\mathbf{a}_i^\top)\Phi^\top\Phi$ is computationally prohibitive when N is large. In that case, we can switch to a stochastic FGD method to solve this problem.

The update rules for $\mathbf{P}$, $\mathbf{Q}$ and g_j are simple:

Update P $\quad \mathbf{P} = \mathbf{P} + \rho(\mathbf{H} - \mathbf{A})$

Update Q $\quad \mathbf{Q} = \mathbf{Q} + \rho(\mathbf{H} - \mathbf{A}^\top)$

Update g_j $\quad g_j = g_j + \rho(f_j - \hat{f}_j)$

3.3.1 RTR Form of RKHS Functions

Next, we present the proof that as long as the RKHS functions f_i, g_i and $\widehat{f}_j$ are initialized to be zero, they are always in the RTR form during the entire execution of the algorithm. We prove this by induction. For the base case (iteration $t = 0$), these functions are all zero, hence admitting the RTR form. For the inductive step, assuming the statement is true at iteration $t - 1$, we prove it holds for iteration t. We begin with f_i, which is solved by FGD. At iteration t, the input of the algorithm is $f_i^{(t-1)}$ and output is $f_i^{(t)}$. The first term of the functional gradient $\triangle f_i$ is in the RTR form, so are g_i and $\widehat{f}_j$ (according to the inductive hypothesis). Then $\triangle f_i$ is in the RTR form. Starting from $f_i^{(t-1)}$ which is in the RTR form according to the inductive hypothesis, f_i is updated iteratively in the following way: $f_i^{(s)} \leftarrow f_i^{(s-1)} - \eta \triangle f_i^{(s-1)}$ (where s indexes FGD iterations), resulting in $f_i^{(t)}$ which is also in the RTR form.

Next, we prove that if $g_j^{(t-1)}$ and $f_i^{(t-1)}$ are in the RTR form, so will be $\widehat{f}_j$. The proof is similar to that of the representer theorem (Schölkopf and Smola, 2002). We decompose $\widehat{f}_j$ into $\widehat{f}_j^{\|}$ and $\widehat{f}_j^{\perp}$, where $\widehat{f}_j^{\|}$ is in $S = \{\sum_{n=1}^{N}\alpha_n k(\mathbf{x}_n, \mathbf{x}), \{\alpha_n\}_{n=1}^{N} \subset \mathbb{R}\}$ (i.e., in the RTR form) and $\widehat{f}_j^{\perp}$ is perpendicular to S, hence $\langle g_j, \widehat{f}_j^{\perp}\rangle = 0$, $\langle f_i, \widehat{f}_j^{\perp}\rangle = 0$. Problem (**T**) can be equivalently written as: $\min_{\widehat{f}_j^{\|}} \; -\langle g_j, \widehat{f}_j^{\|}\rangle + \sum_{i=1}^{K}((P_{ij} + Q_{ij})\langle f_i, \widehat{f}_j^{\|}\rangle + \frac{\rho}{2}((\langle f_i, \widehat{f}_j^{\|}\rangle - A_{ij})^2 + (\langle f_i, \widehat{f}_j^{\|}\rangle - A_{ji})^2))$. Hence the optimal solution of $\widehat{f}_j$ is in the RTR form. For g_j, from its update equation $g_j = g_j + \rho(f_j - \hat{f}_j)$, it is easy to see that if $f_j^{(t-1)}$ and $\widehat{f}_j^{(t-1)}$ are in the RTR form, so will be $g_j^{(t)}$. Note that these RKHS functions are in the RTR form because of the algorithmic procedure (namely, initializing these functions as zero and using FGD to solve f) rather than the representer theorem. If we choose another way of initialization, f may not be in the RTR form.

3.3.2 Scalable Representation of RKHS Functions Based on Random Fourier Features

When the RKHS functions are in the RTR form, $O(N^2D)$ computational cost is incurred where N is the number of training examples and D is the input feature dimension. On large-sized datasets, this is not scalable. In this section, we investigate a scalable representation of RKHS functions based on random Fourier features (RFFs) (Rahimi and Recht, 2007). Given a shift-invariant kernel $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} - \mathbf{y})$ such as the radial basis function (RBF) kernel, it can be approximated with RFFs: $k(\mathbf{x}, \mathbf{y}) = \langle k(\mathbf{x}, \cdot), k(\mathbf{y}, \cdot)\rangle \approx z(\mathbf{x})^\top z(\mathbf{y})$, where $z(\mathbf{x}) \in \mathbb{R}^Q$ is the RFF transformation of $\mathbf{x}$, and can

be seen as an approximation of $k(\mathbf{x}, \cdot)$. $z(\mathbf{x})$ is generated in the following way: (1) compute the Fourier transform $p(\boldsymbol{\omega})$ of the kernel k; (2) draw Q i.i.d samples $\boldsymbol{\omega}_1, \cdots, \boldsymbol{\omega}_Q \in \mathbb{R}^D$ from $p(\boldsymbol{\omega})$ and Q i.i.d samples $b_1, \cdots, b_Q \in \mathbb{R}$ from the uniform distribution on $[0, 2\pi]$; (3) let $z(\mathbf{x}) = \sqrt{\frac{2}{Q}}[\cos(\boldsymbol{\omega}_1^\top \mathbf{x} + b_1), \cdots, \cos(\boldsymbol{\omega}_Q^\top \mathbf{x} + b_Q)]^\top$. For $f \in \mathcal{H}$ where $\mathcal{H}$ is a RKHS induced by a shift-invariant kernel, we know that $f(\mathbf{x}) = \langle f, k(\mathbf{x}, \cdot)\rangle$. Using $z(\mathbf{x})$ to approximate $k(\mathbf{x}, \cdot)$ and $\mathbf{w} \in \mathbb{R}^Q$ to approximate f, we get $f(\mathbf{x}) \approx \mathbf{w}^\top z(\mathbf{x})$. As such, the infinite-dimensional function f can be approximately represented as a finite-dimensional vector $\mathbf{w}$, and the BMD-KM problem defined in Eq.(4) can be written as $\min_{\mathcal{W}} \mathcal{L}(\mathcal{W}) + \lambda\Omega(\mathcal{W})$ where $\mathcal{W} = \{\mathbf{w}_i\}_{i=1}^K$ and $\Omega(\mathcal{W}) = D(\mathbf{G}, \mathbf{I})$ with $G_{ij} = \mathbf{w}_i^\top \mathbf{w}_j$. Now, learning can be conducted over $\mathcal{W}$, and the computational cost is reduced from $O(N^2 D)$ to $O(NQ)$, where Q is the number of RFFs and is much smaller than ND.

4 ANALYSIS

In this section, we analyze how near-orthogonality regularization affects the generalization performance of kernel methods. Specifically, we choose the LDD regularizer to conduct the study while leaving the other two regularizers (SFN and VND) for future work. Inspired by (Xie et al., 2015a), we perform the analysis in two steps. First, we prove that decreasing LDD amounts to decreasing the absolute value of the cosine similarity (AVCS) of RKHS functions, therefore making the RKHS functions close to being orthogonal. Then we show that the upper bound of generalization error is an increasing function of the AVCS. Combining the two pieces together, we conclude that reducing LDD can decrease the generalization error bound.

We begin with the first step. Given the RKHS function set $\mathcal{F} = \{f_i\}_{i=1}^K$, let $s_{ij} = \frac{|\langle f_i, f_j\rangle|}{\|f_i\|_{\mathcal{H}}\|f_j\|_{\mathcal{H}}}$ be the AVCS between f_i and f_j, and $s(\mathcal{F}) = \max_{1\leq i<j\leq K} s_{ij}$ be the maximal AVCS among all pairs of RKHS functions. Drawing inspiration from (Xie et al., 2015a), we prove that the gradient of LDD $\Omega_{ld}(\mathcal{F})$ is an ascent direction of $s(\mathcal{F})$, which is formally given in the following lemma.

Lemma 1 *Let $\widehat{\mathcal{F}} = \{\widehat{f}_i\}_{i=1}^K$ be a RKHS function set where $\widehat{f}_i = f_i + \eta g_i$ and g_i is the functional gradient of $\Omega_{ld}(\mathcal{F})$ w.r.t f_i. Then $\exists \delta > 0$ such that $\forall \eta \in (0, \delta)$, $s(\widehat{F}) \geq s(\mathcal{F})$.*

This implies that $\Omega_{ld}(\mathcal{F})$ and $s(\mathcal{F})$ are closely aligned. Decreasing $\Omega_{ld}(\mathcal{F})$ effectively decreases $s(\mathcal{F})$. Next, we show that the generalization error bounds of BMD-KDML and BMD-KSC are increasing functions of $s(\mathcal{F})$, using technique developed in (Xie et al., 2015b).

Kernel Distance Metric Learning In KDML, the hypothesis function is $u(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^K (f_i(\mathbf{x}) - f_i(\mathbf{y}))^2$ and the loss function ℓ is the logistic loss $\ell(u(\mathbf{x}, \mathbf{y}), t) = \log(1 + \exp((2t-1)u(\mathbf{x}, \mathbf{y})))$. Let $\mathcal{U} = \{u : (\mathbf{x}, \mathbf{y}) \mapsto \sum_{i=1}^K (f_i(\mathbf{x}) - f_i(\mathbf{y}))^2, \{f_i\}_{i=1}^K \subset \mathcal{H}\}$ denote the hypothesis set and $\mathcal{A} = \{\ell : (\mathbf{x}, \mathbf{y}, t) \mapsto \ell(u(\mathbf{x}, \mathbf{y}), t), u \in \mathcal{U}\}$ denote the loss class, which is the composition of the loss function with each of the hypotheses. We assume $\|\mathbf{x}\|_2 \leq C$, $\|f\|_{\mathcal{H}}$ is upper bounded by $B(k)$ which depends on the kernel function k and $|k(\mathbf{x}, \mathbf{y})|$ is upper bounded by $B'(k, C)$, which depends on k and C.

Given the joint distribution p^* of input data pair $(\mathbf{x}, \mathbf{y})$ and the binary label t indicating whether this data pair is similar or dissimilar, the risk of the hypothesis u is $L(u) = \mathbb{E}_{(\mathbf{x},\mathbf{y},t)\sim p^*}[\ell(u(\mathbf{x}, \mathbf{y}), t)]$. Its empirical counterpart (training error) can be defined as $\widehat{L}(u) = \frac{1}{N}\sum_{n=1}^N \ell(u(\mathbf{x}_n, \mathbf{y}_n), t_n)$. The generalization error of a hypothesis u is defined as $L(u) - \widehat{L}(u)$, which represents how well the algorithm can learn and usually depends on the complexity of the hypothesis class and the number of training examples. The generalization error of non-kernelized DML was previously analyzed in (Bellet and Habrard, 2015; Verma and Branson, 2015), focusing on model complexity and sample complexity. Our analysis aims at revealing how near-orthogonality affects the generalization performance.

Next, we analyze how $s(\mathcal{F})$ affects the generalization error bound (GEB) of BMD-KDML. The major result is presented in Theorem 1[2].

Theorem 1 *With probability at least* $1 - \delta$

$$\begin{aligned} &L(u) - \widehat{L}(u) \\ &\leq \tfrac{8B(k)^2 B'(k,C)^2 K}{(1+\exp(-J))\sqrt{N}} + \log(1 + \exp(J))\sqrt{\tfrac{2\log(2/\delta)}{N}} \end{aligned} \tag{11}$$

where $J = 4B(k)^2 B'(k, C)^2((K-1)s(\mathcal{F}) + 1)$.

From the GEB (right hand side of Eq.(11)), we can see four implications. First, near-orthogonality can reduce GEB. The smaller $s(\mathcal{F})$ is (which indicates stronger near-orthogonality), the smaller the GEB is. As discussed earlier, decreasing LDD can decrease $s(\mathcal{F})$, hence can reduce the GEB. This is the major insight of this analysis. Second, the GEB admits a $O(N^{-1/2})$ rate of convergence in terms of sample size N, which matches the same rate as the analysis in (Bellet and Habrard, 2015; Verma and Branson, 2015). Third, the GEB grows linearly with K – the number of RKHS functions. In (Verma and Branson, 2015), the GEB is $O(\sqrt{D})$ where D is the feature dimension. As shown in experiments, K is roughly in the same scale as $\sqrt{D}$. Fourth,

[2]Please refer to the supplements for the proof.

Table 1: Statistics of the Datasets

	#Train	#Test	Dim.	#Class
MIMIC-III	40K	18K	7207	2833
Cars	8144	8041	4096	196
Birds	9000	2788	4096	200
Scenes-15	3140	1345	–	15
Caltech-256	20846	8934	–	256
UIUC-Sports	1254	538	–	8

the GEB is affected by the properties of the kernel function via the kernel-dependent bounds including $B(k)$ and $B'(k, C)$.

Kernel Sparse Coding We assume the basis functions of the KSC are in the image of the reproducing kernel feature map $\phi(\cdot)$ associated with the kernel k, i.e., $f = \phi(\mathbf{d})$, where $\mathbf{d}$ is a D-dimensional vector in the input space $\mathcal{R}$. Similar to KDML, we assume $\|\phi(\mathbf{d})\|_{\mathcal{H}} \leq B(k)$. The risk function $L(\Phi)$ of the dictionary $\Phi = \{\phi(\mathbf{d}_i)\}_{i=1}^K$ is defined as $\mathbb{E}_{\mathbf{x}\sim p^*}[\min_{\|\mathbf{a}\|_0 \leq m} \|k(\mathbf{x}, \cdot) - \sum_{i=1}^K a_i \phi(\mathbf{d}_i)\|_{\mathcal{H}}]$, where m is a parameter that controls the sparsity of linear coefficients $\mathbf{a}$. Let $\widetilde{L}(\Phi)$ denote the empirical risk function on N training examples. We have the following results on the generalization error of KSC:

Theorem 2 *Let $\mathcal{R}$ have ϵ covers of order $(C/\epsilon)^D$ where C is a constant. Let ϕ be uniformly H-Holder of order $\alpha > 0$ over $\mathcal{R}$ and let $\gamma = \max_{\mathbf{d}\in\mathcal{R}} \|\phi(\mathbf{d})\|_{\mathcal{H}}$. Let ν be any distribution on $\mathcal{R}$, then with probability at least $1 - e^{-\delta}$, for all dictionaries Φ, we have:*

$$L(\Phi) - \widehat{L}(\Phi) \leq \gamma\left(\sqrt{\frac{DK\ln\left(\sqrt{N}C^{\alpha}\frac{m\gamma^2 H}{1-mB^2(k)s(\Phi)}\right)}{2\alpha N}} + \sqrt{\frac{\delta}{2N}}\right) + \sqrt{\frac{4}{N}}.$$

This generalization bound is an increasing function of $s(\Phi)$. Hence, if we encourage the RKHS functions to approach orthogonal, i.e., decreasing $s(\Phi)$, then this generalization bound can be reduced.

5 EXPERIMENTS

In this section, we present experimental results on BMD-KDML and BMD-KSC.

Datasets We used six datasets in the experiments: an electronic health record dataset MIMIC-III (Johnson et al., 2016); five image datasets including Stanford-Cars (Cars) (Krause et al., 2013), Caltech-UCSD-Birds (Birds) (Welinder et al., 2010), Scenes-15 (Lazebnik et al., 2006), Caltech-256 (Griffin et al., 2007) and UIUC-Sports (Li and Fei-Fei, 2007). The first three were used for KDML and the last three for KSC. Their statistics are summarized in Table 1. For each dataset, five random train/test splits were performed, and the results were averaged over the five runs. For the MIMIC-III dataset, we extracted features from demographics (including age and gender), clinical notes (including bag-of-words and Word2Vec (Mikolov et al., 2013)) and lab tests (including zero-order, first-order, and second-order temporal features). The total feature dimension is 7207. The features for Cars and Birds datasets were extracted using the VGG16 (Simonyan and Zisserman, 2015) convolutional neural network trained on the ImageNet (Deng et al., 2009) dataset, which were the outputs of the second fully-connected layer with 4096 dimensions. For Scenes-15, Caltech-256 and UIUC-Sport, we extracted pixel-level dense SIFT (Lowe, 2004) features where the step size and patch size were 8 and 16, respectively.

Experimental Setup For BMD-KDML and BMD-KSC, we experimented with six combinations between three regularizers including SFN, LDD and VND, and two representations of RKHS functions including RTR and RFF. In DML experiments, two data examples were labeled as similar if belonging to the same class and dissimilar otherwise. The learned distance metrics were applied for retrieval whose performance was evaluated using precision@k. Precision@k is defined as n/k where n is the number of examples (among the top k retrieved examples) that have the same class label with the query. We compared with three groups of baseline methods: (1) KDML (Eq.(5) without the regularizer $\Omega(\mathcal{F})$) and its variants under different regularizers including squared Hilbert norm (SHN), DPP (Zou and Adams, 2012) and Angle (Xie et al., 2015a); (2) other kernel DML methods including the ones proposed in (Tsang et al., 2003) (Tsang) and (Jain et al., 2012) (Jain), multiple kernels DML (MK-DML) (Wang et al., 2011) and pairwise constrained component analysis (PCCA) (Mignon and Jurie, 2012); (3) non-kernel metric learning (ML) methods, including information theoretic ML (ITML) (Davis et al., 2007), logistic discriminant ML (LDML) (Guillaumin et al., 2009), DML with eigenvalue optimization (DML-Eig) (Ying and Li, 2012), information-theoretic semi-supervised ML via entropy regularization (Seraph) (Niu et al., 2012) and geometric mean ML (GMML) (Zadeh et al., 2016); (4) Euclidean distance (EUC). For methods in group (1), the RKHS functions are represented in the RTR form. In sparse coding experiments, on top of the SIFT features, we used kernel sparse coding to learn a set of RKHS functions and represented each SIFT feature into a sparse code. To obtain image-level features, we applied max-pooling (Yang et al., 2009) and spatial pyramid matching (Lazebnik et al., 2006; Yang et al., 2009) over the pixel-level sparse codes. The following baselines were compared with: sparse coding (SC) (Yang et al., 2009), unregularized kernel SC (KSC) (Gao et al., 2010), KSC regularized by SHN, DPP and Angle. In these methods, the RKHS functions are represented in the RTR form. We used 5-fold

Table 2: Retrieval Precision@10 (%) on Three Datasets

	MIMIC-III	Cars	Birds
EUC	58.3 ± 0.1	37.8 ± 0.0	43.2 ± 0.0
ITML	69.3 ± 0.4	50.1 ± 0.0	52.9 ± 0.3
LDML	70.9 ± 0.9	51.3 ± 0.0	52.1 ± 0.2
DML-Eig	70.6 ± 0.7	50.7 ± 0.0	53.3 ± 0.8
Seraph	71.7 ± 0.2	53.6 ± 0.0	52.9 ± 0.2
GMML	71.2 ± 0.3	54.2 ± 0.0	53.7 ± 0.6
Tsang	73.5 ± 0.2	55.8 ±0.0	53.9 ± 0.4
MKDML	75.1 ± 0.9	53.5 ±0.0	54.4 ± 0.1
Jain	74.9 ± 1.1	53.9 ±0.0	55.9 ± 0.6
PCCA	73.4 ± 0.5	56.4 ±0.0	55.1 ± 0.9
KDML	73.8 ± 0.9	54.9 ±0.0	54.7 ± 0.5
KDML-SHN	74.2 ± 0.6	55.4 ±0.0	54.8 ± 0.9
KDML-DPP	75.5 ± 0.8	56.4 ±0.0	57.3 ± 0.3
KDML-Angle	75.9 ± 0.2	56.8 ±0.0	57.1 ± 0.6
KDML-SFN-RTR	76.3 ± 0.7	56.6 ± 0.0	56.4 ± 0.1
KDML-VND-RTR	**77.1** ± 0.6	**57.7** ± 0.0	58.9 ± 0.7
KDML-LDD-RTR	76.7 ± 0.3	57.4 ± 0.0	**59.2** ± 0.3
KDML-SFN-RFF	75.9 ± 0.1	56.5 ± 0.0	56.0 ± 0.2
KDML-VND-RFF	76.9 ± 0.4	57.2 ± 0.0	58.8 ± 0.6
KDML-LDD-RFF	76.8 ± 0.8	57.1 ± 0.0	58.5 ± 0.4

Table 3: The number of RKHS functions that achieves the precision@10 in Table 2

	MIMIC-III	Cars	Birds	Average
KDML	300	400	300	333
KDML-SHN	300	400	300	333
KDML-DPP	200	300	300	267
KDML-Angle	200	300	200	233
KDML-SFN-RTR	200	200	200	200
KDML-VND-RTR	100	200	200	167
KDML-LDD-RTR	100	200	200	167
KDML-SFN-RFF	100	200	200	167
KDML-VND-RFF	100	200	200	167
KDML-LDD-RFF	100	200	200	167

cross validation to tune the regularization parameters in $\{10^{-5}, 10^{-4}, \cdots, 10^{5}\}$, the number of RKHS functions in $\{50, 100, 200, \cdots, 500\}$ and the dimension of RFF in $\{1, 2, \cdots, 10\} \times D$ where D is the feature dimension of input data. The kernel function was chosen to be the radial basis function (RBF) $\exp(-\gamma\|\mathbf{x}-\mathbf{y}\|_2^2)$ where the scale parameter γ is tuned in $\{10^{-3}, 10^{-2}, \cdots, 10^{3}\}$. The parameter ρ in ADMM-based algorithm was set to 1. The learning rate in functional gradient descent was set to 0.001.

Results Table 2 shows the retrieval precision@10 on three datasets, where we observe the following. First, BMD-KDML methods including KDML-(SFN,VND,LDD)-(RTR,RFF) greatly outperform unregularized and SHN-regularized KDML, which demonstrates that near-orthogonality regularization is an effective way to reduce overfitting. Second, BMD regularizers including SFN, VND and LDD outperform other near-orthogonality regularizers including DPP and Angle, possibly because they are insensitive to vector scaling and amenable for optimization. Third, VND and LDD achieve comparable performance and outperform SFN, possibly because they measure near-orthogonality in a global way while SFN conducts that in a pairwise fashion. Fourth, RFF representation of RKHS functions performs comparably to RTR, in spite of the fact

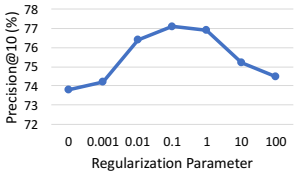

Figure 1: Precision@10 versus the Regularization Parameter λ on MIMIC-III

that it is an approximation method. Finally, the BMD-KDML methods achieve better performance than non-kernel DML methods and other kernel DML methods, suggesting their competitive ability in learning effective distance metrics.

Table 3 shows the number of RKHS functions under which the precision@10 in Table 2 is achieved. It can be seen that the BMD-KDML methods utilize much fewer functions than KDML while achieving better precision@10. For instance, KDML-VND-RTR achieves 77.1% precision@10 with 100 functions on the MIMIC-III dataset while KDML achieves 73.8% precision@10 with 300 functions. These results demonstrate the ability of the BMD regularizers in reducing model size without sacrificing modeling power. By encouraging the functions to be near-orthogonal, the BMD regularizers decrease the redundancy among functions and make the functions highly complementary. As a result, a small number of such functions are able to capture the patterns in the data sufficiently well. In addition, the BMD regularizers achieve better precision@10 with fewer functions than other near-orthogonality regularizers including DPP and Angle, suggesting their better efficacy in promoting near-orthogonality.

In the next experiment, we investigate whether near-orthogonality regularization can better capture infrequent patterns. We select 3 frequent diseases (patterns) and 5 infrequent ones from the MIMIC-III dataset. A disease is regarded as *frequent* if the number of hospital admissions diagnosed with this disease is greater than 300. Table 4 shows the precision@10 on the 8 diseases, from which we observe that: (1) on the 5 infrequent diseases (labeled as D4–D8), the BMD-KDML methods achieve much higher precision@10 than the unregularized KDML, suggesting that by encouraging the functions to be close to being orthogonal, the BMD regularizers can better capture infrequent patterns; (2) on the 3 frequent diseases (labeled as D1–D3), the precision@10 achieved by the BMD-KDML methods is comparable with that achieved by the unregularized KDML, indicating that the BMD regularizers do not compromise the modeling effects on the frequent patterns. On the infrequent diseases, the BMD-KDML methods outperform KDML-DPP and KDML-Angle, suggesting that the BMD regularizers have better abilities in promoting

Table 4: Retrieval precision@10 (%) on three frequent and five infrequent diseases in the MIMIC-III dataset. The number next to a disease ID is its frequency. Note that diseases D1–D3 are frequent diseases, while that D4–D8 are infrequent ones.

	D1 (3566)	D2 (3498)	D3 (2757)	D4 (204)	D5 (176)	D6 (148)	D7 (131)	D8 (121)
Tsang	80.3 ± 0.3	82.8 ± 0.7	81.9 ± 0.4	6.3 ± 0.7	3.9 ± 0.5	4.5 ± 0.5	6.7 ± 0.9	5.1 ± 0.2
MKDML	83.1 ± 0.2	83.4 ± 0.7	82.3 ± 0.6	3.7 ± 1.2	5.5 ± 0.1	9.3 ± 0.8	10.0 ± 0.5	3.7 ± 0.4
Jain	82.7 ± 0.7	84.6 ± 0.5	**82.9** ± 0.4	7.2 ± 0.4	8.2 ± 0.4	3.4 ± 0.9	6.2 ± 0.7	8.7 ± 0.3
PCCA	82.2 ± 0.2	82.1 ± 0.6	82.1 ± 0.3	9.4 ± 0.8	7.7 ± 0.2	5.2 ± 0.4	6.1 ± 0.1	3.2 ± 0.4
KDML	82.6 ± 0.7	83.9 ± 1.2	81.7 ± 0.6	7.4 ± 1.0	5.3 ± 0.9	5.7 ± 0.3	3.8 ± 0.8	3.5 ± 0.4
KDML-SHN	82.1 ± 0.5	83.6 ± 0.4	82.4 ± 0.9	8.3 ± 0.1	5.1 ± 0.8	4.7 ± 0.2	3.4 ± 0.9	3.7 ± 0.8
KDML-DPP	83.4 ± 0.4	**84.7** ± 0.7	82.7 ± 1.0	11.5 ± 0.3	9.7 ± 0.5	10.4 ± 0.4	7.3 ± 0.2	7.9 ± 0.1
KDML-Angle	83.7 ± 0.1	84.3 ± 0.1	81.8 ± 0.3	10.6 ± 0.2	10.2 ± 0.8	9.5 ± 0.6	8.8 ± 0.5	7.2 ± 0.3
KDML-SFN-RTR	82.5 ± 0.8	84.2 ± 1.2	82.2 ± 0.1	15.0 ± 0.3	13.4 ± 0.1	13.8 ± 0.2	12.1 ± 0.6	10.9 ± 0.5
KDML-VND-RTR	**83.9** ± 0.9	84.5 ± 0.8	82.6 ± 0.2	15.5 ± 0.9	16.2 ± 0.7	14.3 ± 0.7	12.4 ± 0.8	14.7 ± 0.8
KDML-LDD-RTR	83.7 ± 0.1	83.8 ± 0.9	82.2 ± 0.6	14.8 ± 0.4	14.2 ± 0.1	13.7 ± 0.3	10.3 ± 0.1	12.8 ± 0.2
KDML-SFN-RFF	82.3 ± 0.9	84.1 ± 0.6	81.1 ± 0.5	15.1 ± 0.2	14.9 ± 0.5	15.2 ± 0.9	13.5 ± 0.1	10.8 ± 0.3
KDML-VND-RFF	83.4 ± 0.2	83.6 ± 0.5	82.7 ± 0.4	15.2 ± 0.5	15.6 ± 0.9	10.6 ± 0.8	12.0 ± 1.1	10.3 ± 0.7
KDML-LDD-RFF	82.9 ± 0.2	84.0 ± 1.0	82.5 ± 0.9	14.4 ± 0.9	13.9 ± 1.2	15.4 ± 0.5	13.9 ± 0.2	14.4 ± 0.1

Table 5: Convergence Time (Hours) on Three Datasets

	MIMIC-III	Cars	Birds
KDML-SFN-RTR	69.4	17.2	18.6
KDML-VND-RTR	69.8	17.4	18.9
KDML-LDD-RTR	69.9	17.5	18.9
KDML-SFN-RFF	12.6	2.7	2.9
KDML-VND-RFF	12.9	2.8	3.1
KDML-LDD-RFF	12.8	2.8	3.1

Table 6: Classification Accuracy (%) on Three Datasets

	Scenes-15	Caltech-256	UIUC-Sports
SC	83.6 ± 0.2	42.3 ± 0.4	87.4 ± 0.5
KSC	85.4 ± 0.5	44.7 ± 0.8	88.2 ± 0.1
KSC-SHN	85.8 ± 0.6	45.4 ± 0.5	88.3 ± 0.3
KSC-DPP	86.3 ± 0.3	47.3 ± 0.8	89.3 ± 0.2
KSC-Angle	86.8 ± 0.1	46.1 ± 0.8	89.5 ± 0.5
KSC-SFN-RTR	87.1 ± 0.5	47.2 ± 0.5	89.9 ± 0.3
KSC-VND-RTR	**87.9** ± 0.7	**48.6** ± 0.3	**91.3** ± 0.5
KSC-LDD-RTR	87.4 ± 0.4	48.1 ± 0.6	90.7 ± 0.2
KSC-SFN-RFF	86.8 ± 0.6	46.5 ± 0.1	89.5 ± 0.7
KSC-VND-RFF	87.5 ± 0.5	48.2 ± 0.4	90.4 ± 0.8
KSC-LDD-RFF	87.2 ± 0.1	47.8 ± 0.2	90.2 ± 0.4

near-orthogonality than DPP and Angle.

Further, Figure 1 shows how the precision@10 on MIMIC-III varies as we increase the regularization parameter λ in KDML-VND-RTR. As can be seen, the best precision@10 is achieved under a modest λ. A very large λ would make the functions strictly orthogonal, as kernel PCA and ICA do, which would result in excessively strong regularization and therefore poor performance.

We also compares the convergence time of BMD-KDML under two representations: RTR and RFF. As shown in Table 5, RFF results in much faster convergence since this representation does not depend on the training data. While computationally efficient, RFF does not sacrifice much modeling power. Table 2 shows that RFF achieves precision@10 that is comparable to RTR.

Next, we present the kernel sparse coding results. Table 6 shows the classification accuracy on three datasets, from which we observe similar results as in the KDML experiments. First, KSC-(SFN,VND,LDD)-(RTR,RFF) achieve better accuracy than the unregularized and the SHN-regularized KSC. Second, the BMD regularizers outperform other near-orthogonality regularizers including DPP and Angle. Third, VND and LDD are superior to SFN. Fourth, the RFF representation of RKHS functions performs comparably to RTR. Finally, the BMD-KSC methods outperform the non-kernel SC methods and other kernel SC methods. These observations demonstrate the efficacy of the BMD regularizers in reducing overfitting and promoting near-orthogonality.

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a new regularization approach for kernel methods – near-orthogonality regularization, which encourages the RKHS functions to be close to being orthogonal, for the sake of reducing overfitting, decreasing model size without compromising modeling power and effectively capturing infrequent patterns. We design a family of near-orthogonality regularizers based on minimizing the Bregman matrix divergences between functions' Gram matrix and an identity matrix, and apply them to promote near-orthogonality in two kernel methods. An efficient ADMM-based optimization algorithm is developed where the RKHS functions are learned using functional gradient descent. The analysis reveals that the near-orthogonality regularization can reduce the generalization error of kernel methods. Experiments demonstrate the effectiveness of the proposed regularizers.

For future work, we plan to apply near-orthogonality regularization to "deep" kernel methods (Cho and Saul, 2009; Wilson et al., 2016) which bridge kernel methods with deep learning.

Acknowledgements

We would like to thank the anonymous reviewers for the helpful suggestions. This research is supported in part by National Institutes of Health P30DA035778, R01GM114311, National Science Foundation IIS1617583, IIS1563887, DARPA FA872105C0003.

References

Francis R Bach and Michael I Jordan. Kernel independent component analysis. *JMLR*, 2002.

A. Bellet and A. Habrard. Robustness and generalization for metric learning. *Neurocomputing*, 2015.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 2011.

Youngmin Cho and Lawrence K Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, 2009.

M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, and D. Batra. Reducing overfitting in deep networks by decorrelating representations. *ICLR*, 2015.

Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina F Balcan, and Le Song. Scalable kernel methods via doubly stochastic gradients. In *NIPS*, 2014.

J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *ICML*, 2007.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Inderjit S Dhillon and Joel A Tropp. Matrix nearness problems with Bregman divergences. *Matrix Analysis and Applications*, 2007.

Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Kernel sparse representation for image classification and face recognition. In *ECCV*, 2010.

Rudolf Gorenflo, Yuri Luchko, and Francesco Mainardi. Analytical properties and applications of the Wright function. *arXiv preprint math-ph/0701069*, 2007.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Is that you? Metric learning approaches for face identification. In *ICCV*, 2009.

P. Jain, B. Kulis, J. Davis, and I. Dhillon. Metric and kernel learning using a linear transformation. *JMLR*, 2012.

A. Johnson, T. Pollard, L. Shen, L. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Celi, and R. Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3, 2016.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshops*, 2013.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 2012.

Brian Kulis, Mátyás A Sustik, and Inderjit S Dhillon. Low-rank kernel learning with Bregman matrix divergences. *JMLR*, 2009.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing scene categories. In *CVPR*, 2006.

Li-Jia Li and Li Fei-Fei. What, where and who? Classifying events by scene recognition. In *ICCV*, 2007.

David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

Charles A Micchelli and Massimiliano Pontil. Learning the kernel function via regularization. *JMLR*, 2005.

A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *CVPR*, 2012.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

G. Niu, B. Dai, M. Yamada, and M. Sugiyama. Information-theoretic semisupervised metric learning via entropy regularization. *Neural Computation*, 2012.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 1997.

Ali Rahimi and Ben Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

B. Schölkopf, A. Smola, and K. Müller. Kernel principal component analysis. In *ICANN*, 1997.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

Ivor W Tsang, James T Kwok, C Bay, and H Kong. Distance metric learning with kernels. In *ICANN*, 2003.

Nakul Verma and Kristin Branson. Sample complexity of learning Mahalanobis distance metrics. *NIPS*, 2015.

J. Wang, H. Do, A. Woznica, and A. Kalousis. Metric learning with multiple kernels. In *NIPS*, 2011.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P Perona. Caltech-UCSD birds. 2010.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *AISTATS*, 2016.

P. Xie, Y. Deng, and E. Xing. Diversifying restricted Boltzmann machine for document modeling. In *SIGKDD*, 2015a.

P. Xie, Y. Deng, and E. Xing. On the generalization error bounds of neural networks under mutual angular regularization. *arXiv:1511.07110*, 2015b.

Pengtao Xie. Learning compact and effective distance metrics with diversity regularization. In *ECML*, 2015.

Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, 2002.

Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

Yiming Ying and Peng Li. Distance metric learning with eigenvalue optimization. *JMLR*, 2012.

Pourya Habib Zadeh, Reshad Hosseini, and Suvrit Sra. Geometric mean metric learning. In *ICML*, 2016.

James Y. Zou and Ryan P. Adams. Priors for diversity in generative latent variable models. In *NIPS*, 2012.