The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Compact Multi-Label Learning

Xiaobo Shen,[‡*] Weiwei Liu,[♯*] Ivor W. Tsang,[♮] Quan-Sen Sun,[§†] Yew-Soon Ong[‡]
[‡]School of Computer Science and Engineering, Nanyang Technological University
[♯]School of Computer Science and Engineering, The University of New South Wales
[♮]Center for Artificial Intelligence, University of Technology Sydney
[§]School of Computer and Engineering, Nanjing University of Science and Technology
{njust.shenxiaobo, liuweiwei863}@gmail.com, ivor.tsang@uts.edu.au, sunquansen@njust.edu.cn, asysong@ntu.edu.sg

Abstract

Embedding methods have shown promising performance in multi-label prediction, as they can discover the dependency of labels. Most embedding methods cannot well align the input and output, which leads to degradation in prediction performance. Besides, they suffer from expensive prediction computational costs when applied to large-scale datasets. To address the above issues, this paper proposes a Co-Hashing (CoH) method by formulating multi-label learning from the perspective of cross-view learning. CoH first regards the input and output as two views, and then aims to learn a common latent hamming space, where input and output pairs are compressed into compact binary embeddings. CoH enjoys two key benefits: 1) the input and output can be well aligned, and their correlations are explored; 2) the prediction is very efficient using fast cross-view kNN search in the hamming space. Moreover, we provide the generalization error bound for our method. Extensive experiments on eight real-world datasets demonstrate the superiority of the proposed CoH over the state-of-the-art methods in terms of both prediction accuracy and efficiency.

Introduction

In multi-label learning (Zhang and Zhou 2014; Liu, Tsang, and Müller 2017), each instance is represented by a set of labels. For example, a document can be associated with a range of topics, such as *sports*, *finance*, and *education*; or an image may be tagged with both *beach* and *tree*. The task of multi-label learning is to learn a function that can predict the proper label sets for unseen instances. Nowadays it is becoming more and more relevant to a number of applications, ranging from document classification to gene function prediction and automatic image annotation.

Much of the literature (Chen and Lin 2012; Guo and Schuurmans 2013; Zhang and Zhou 2014) has shown that multi-label learning methods usually achieve better prediction performance by explicitly capturing label dependency. To this end, (Hsu et al. 2009) are the first to propose embedding the label vectors into a low-dimensional subspace using random projection, and then building a regression model in the embedding space. To learn better embeddings, various embedding methods have since been developed, such as canonical correlation analysis (CCA) (Zhang and Schneider 2011), maximum margin output coding (MMOC) (Zhang and Schneider 2012). Such methods have achieved the impressive performance on small datasets. However, their decoding schemes often require solving a quadratic programming (QP) problem in a combinatorial space, which is very computationally expensive.

To improve efficiency, large margin kNN (LM-kNN) (Liu and Tsang 2015) for fast multi-label prediction has been recently proposed. LM-kNN learns a distance metric to discover the label dependency so that instances with very different multiple labels are moved further away. More importantly, LM-kNN is the first approach to employ k-nearest neighbor (kNN) search in the embedding space for prediction and it avoids an expensive decoding procedure. Another state-of-the-art method that also uses kNN search for fast prediction is the recently proposed sparse local embedding for extreme classification (SLEEC) (Bhatia et al. 2015). SLEEC first seeks the embedding of labels by preserving the pairwise distances between a few nearest label neighbors, and then learns the regressor in the embedding space. A gap usually exists between the input and output in the multi-label setting, and the performance of kNN decision rule for multi-label prediction significantly depends on the alignment between the input and output. However, the above two methods fail to explore this correlation, thus their learned embeddings are not well aligned, leading to degradation in prediction performance. In addition, their prediction efficiency relies on the speed of kNN search, which is slow for large-scale application. How to further improve the prediction efficiency also remains less explored.

To address the aforementioned issues, this paper regards the input and output as two distinct views, and provides a new insight into multi-label learning from a cross-view perspective (Quadrianto and Lampert 2011; Dhillon, Foster, and Ungar 2011; Guo and Xiao 2012; Shen et al. 2017b), which aims to integrate the complementary advantages of the two views to accomplish multi-label prediction tasks. Specifically, we propose Co-Hashing (CoH) to learn a common latent hamming space, where the input and output are jointly embedded into compact binary codes and well aligned. In the latent subspace, the correlation between in-

[*]indicates equal contribution.
[†]Corresponding author.

put and output is built, and knowledge can be better transferred. In prediction phrase, we directly search the kNN label embeddings of any testing instance in the constructed latent subspace. The kNN cross-view label embeddings search benefits from the view alignment, leading to better multi-label prediction performance. CoH is also inspired by the success of hashing in large-scale approximate nearest neighbor search (Wang et al. 2016), which supports to improve prediction efficiency by accelerating kNN search in the learned hamming subspace. Extensive experiments on eight real-world datasets demonstrate the proposed CoH outperforms the state-of-the-art methods in terms of both prediction accuracy and efficiency.

Related Work

Multi-label Learning Embedding methods have shown promising results for multi-label learning, especially for many label case. They aim to project label vector into a low-dimensional subspace, and then perform the regression in the embedding subspace. Until now a number of embedding methods have been proposed via various compression and decompression techniques (Hsu et al. 2009; Zhang and Schneider 2011; Tai and Lin 2012; Zhang and Schneider 2012). To name a few, (Tai and Lin 2012) proposed the principal label space transformation (PLST), which uses principal component analysis (PCA) to compress the label vector. Based on canonical correlation analysis (CCA), (Zhang and Schneider 2011) took both feature and label into consideration. After that, maximum margin output coding (MMOC) (Zhang and Schneider 2012) was developed to learn an output coding. Recently, sparse local embedding for extreme classification (SLEEC) is proposed to seek the embedding of labels by preserving the pairwise distances between a few nearest label neighbors. Generally speaking, these methods cannot well establish the alignment between the input and output, which influences the performance of the kNN search for multi-label prediction.

Cross-view Learning Cross-view learning (Guo and Xiao 2012) aims to analyze the relationships among different views, and tries to bridge the gap between these views. Its basic idea is to seek a common latent subspace, where embeddings from different views are well aligned. As a result, cross-view learning allows features from different views to be matched in the latent subspace. Cross-view learning has been successfully applied to many applications, such as image-text retrieval, cross language text classification, person re-identification.

In this work, we try to consider multi-label learning from the perspective of cross-view learning. Specifically, we assume that the input and output share the common subspace, which is reflective of the latent semantics. The common subspace establishes the correlations between the input and output, thus label search and prediction in this subspace is significant. To achieve our goal, we formulate multi-label learning as a model that learns the semantic latent embeddings shared by both the input and output.

Hash Code Learning Hashing technique (Wang et al. 2016) has become a widely-studied solution to approximate nearest neighbor (ANN) search for its great gains in both storage and computation among massive data. The basic idea of hashing (Weiss, Torralba, and Fergus 2009; Gong et al. 2013; Shen et al. 2015; 2017a; 2017b) is to map high-dimensional data into a low-dimensional discrete code space called Hamming space, while preserving similarity structure in the original space. Accordingly, each data point is represented by a short code called hash code consisting of a sequence of bits.

Some supervised hashing methods (Gong et al. 2013; Zhao et al. 2015; Lai et al. 2016) have been proposed in the multi-label setting. However, the proposed CoH is different from them, which are only developed for retrieval, and cannot be applied for multi-label learning. The motivation of developing CoH is to achieve fast multi-label prediction using hashing technique. Benefiting from the efficient Hamming distance computation, CoH can improve efficiency of kNN search. To our knowledge, this is the first work applying hashing technique to accelerate multi-label prediction.

Co-Hashing

Let $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^{n}$ be the training dataset, $\mathbf{x}^{(i)} \in \mathcal{X} \subseteq \mathbb{R}^p$ be an input (instance) vector, $\mathbf{y}^{(i)} \in \mathcal{V} \subseteq \{0,1\}^q$ be the corresponding output (label) vector, and let n denote the number of training instances. Let $\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}\right] \in \mathbb{R}^{p\times n}$ be the instance matrix and $\mathbf{Y} = \left[\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(n)}\right] \in \mathbb{R}^{q\times n}$ be the label matrix. The goal of multi-label learning is to learn a multi-label classifier $f : \mathbb{R}^p \to \{0,1\}^q$ that accurately predicts the label vector for any unseen instance.

Formulation

Inspired by the success in cross-view learning (Quadrianto and Lampert 2011; Dhillon, Foster, and Ungar 2011; Guo and Xiao 2012; Shen et al. 2017b), we consider its benefits for multi-label learning. In cross-view learning, it is commonly known that data from the same objects described in different views share a certain common subspace (Guo and Xiao 2012). Accordingly, we assume that a low-dimensional common latent hamming space $\mathcal{B}$ exists for the input and output, where the semantic embeddings $\mathbf{B} = \left[\mathbf{b}^{(1)}, \ldots, \mathbf{b}^{(n)}\right] \in \{-1,1\}^{d\times n}$ appropriately represent the input and output training pairs, and $\mathbf{b}^{(i)} \in \{-1,1\}^d$ denotes the binary embedding of the i-th training pair. The binary constraint on $\mathbf{B}$ is to achieve fast multi-label prediction, benefiting from the efficient computation in Hamming space (Wang et al. 2016). To achieve this goal, we derive the objective function of the proposed CoH as follows

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{B}} \ & \|\mathbf{U}^\top\mathbf{X} - \mathbf{B}\|_F^2 + \|\mathbf{V}^\top\mathbf{Y} - \mathbf{B}\|_F^2 + \alpha \mathrm{Tr}\left(\mathbf{B}\mathbf{L}\mathbf{B}^\top\right) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1,1\}^{d\times n}, \mathbf{B}\mathbf{B}^\top = n\mathbf{I}_d \end{aligned} \tag{1}$$

where the constraint $\mathbf{B}\mathbf{B}^\top = n\mathbf{I}_d$ is introduced to make d bits mutually uncorrelated, such that the redundancy among these bits can be minimized; $\mathbf{L} = \mathbf{I}_n - \mathbf{S} \in \mathbb{R}^{n\times n}$ is the normalized graph Laplacian (Belkin, Niyogi, and Sindhwani 2006). In this work, the normalized similarity graph is defined as $\mathbf{S} = \mathbf{Y}^\top\mathbf{\Lambda}^{-1}\mathbf{Y}$, where $\mathbf{\Lambda} = \mathrm{diag}\left(\mathbf{Y}^\top\mathbf{1}\right) \in \mathbb{R}^{q\times q}$ is

used to normalize each row. In (1), the first two terms model the mapping loss; the last manifold regularization term is employed to preserve the semantic similarity structure between the embeddings in the hamming space.

Generally, (1) is difficult to solve because of the binary constraint. We first define a set $\Omega = \{\mathbf{Z} \in \mathbb{R}^{d\times n} | \mathbf{Z}\mathbf{Z}^\top = n\mathbf{I}_d\}$, then provide a new formulation that softens the orthogonal constraint in (1) as

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{B}} & \|\mathbf{U}^\top\mathbf{X} - \mathbf{B}\|_F^2 + \|\mathbf{V}^\top\mathbf{Y} - \mathbf{B}\|_F^2 \\ & + \alpha \mathrm{Tr}\left(\mathbf{B}\mathbf{L}\mathbf{B}^\top\right) + \frac{\rho}{2}\mathrm{dist}^2\left(\mathbf{B}, \Omega\right) \qquad (2) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{d\times n} \end{aligned}$$

where $\mathrm{dist}\left(\mathbf{B}, \Omega\right) = \min_{\mathbf{Z}\in\Omega} \|\mathbf{B} - \mathbf{Z}\|_F$ measures the distance from $\mathbf{B}$ to the set Ω, and $\rho \geq 0$ is a regularization parameter. In (2), we allow a certain discrepancy between $\mathbf{B}$ and Ω, such that (2) can be more flexible than (1). Due to the fact that $\mathrm{Tr}\left(\mathbf{B}^\top\mathbf{B}\right) = \mathrm{Tr}\left(\mathbf{Z}^\top\mathbf{Z}\right) = nd$, (2) can be further transformed to the following problem

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{B},\mathbf{Z}} & \|\mathbf{U}^\top\mathbf{X} - \mathbf{B}\|_F^2 + \|\mathbf{V}^\top\mathbf{Y} - \mathbf{B}\|_F^2 \\ & + \alpha \mathrm{Tr}\left(\mathbf{B}\mathbf{L}\mathbf{B}^\top\right) - \rho \mathrm{Tr}\left(\mathbf{B}^\top\mathbf{Z}\right) \qquad (3) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{d\times n}, \mathbf{Z} \in \mathbb{R}^{d\times n}, \mathbf{Z}\mathbf{Z}^\top = n\mathbf{I}_d \end{aligned}$$

In the next section, we propose a computationally tractable optimization algorithm to solve (3).

Optimization

In essence, (3) is a nonlinear mixed-integer optimization problem involving the discrete constraint on $\mathbf{B}$ and non-convex orthogonal constraint on $\mathbf{Z}$, which is generally NP-hard. We propose a tractable alternating algorithm to iteratively optimize each variable. The flowchart of CoH is described by Algorithm 1.

Update U, V For a given $\mathbf{B}$, (3) is reduced to a least square minimization problem with respect to $\mathbf{U}$ and $\mathbf{V}$. We can obtain the closed-form solution of $\mathbf{U}$ as $\mathbf{U} = \left(\mathbf{X}\mathbf{X}^\top + \epsilon\mathbf{I}_p\right)^{-1}\mathbf{X}\mathbf{B}^\top$, where ϵ is a small non-negative parameter to avoid overfitting, and we simply set $\epsilon = 0.001$. The solution of $\mathbf{V}$ can be similarly obtained.

Update B By dropping some irrelevant terms to $\mathbf{B}$, we have

$$\begin{aligned} \max_{\mathbf{B}} \quad & f(\mathbf{B}) = \mathrm{Tr}\left(\mathbf{B}\mathbf{S}\mathbf{B}^\top\right) \qquad (4) \\ & + \frac{1}{\alpha}\mathrm{Tr}\Big(2(\mathbf{U}^\top\mathbf{X} + \mathbf{V}^\top\mathbf{Y})\mathbf{B}^\top + \rho\mathbf{Z}\mathbf{B}^\top\Big) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{d\times n} \end{aligned}$$

In this work, (4) can be solved via the majorization method, which is widely used in many statistic areas. The main idea of majorization methodology is to iteratively optimize a surrogate function. In specific, we iteratively optimize a local function $\hat{f}_i(\mathbf{B})$ that linearizes $f(\mathbf{B})$ at the point $\mathbf{B}^{(i)}$, and employ $\hat{f}_i(\mathbf{B})$ as a surrogate of $f(\mathbf{B})$. Given $\mathbf{B}^{(i)}$, the next discrete point $\mathbf{B}^{(i+1)}$ can be derived by optimizing the following objective function

$$\begin{aligned} \max_{\mathbf{B}} \quad & \hat{f}_i(\mathbf{B}) = f(\mathbf{B}^{(i)}) + \langle \nabla f(\mathbf{B}^{(i)}), \mathbf{B} - \mathbf{B}^{(i)} \rangle \qquad (5) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{d\times n} \end{aligned}$$

where $\nabla f(\mathbf{B}^{(i)}) = 2\mathbf{B}^{(i)}\mathbf{S} + \frac{1}{\alpha}\left(2(\mathbf{U}^\top\mathbf{X} + \mathbf{V}^\top\mathbf{Y}) + \rho\mathbf{Z}\right)$. Note that $\nabla f(\mathbf{B}^{(i)})$ may contain many entries, thus multiple solutions for $\mathbf{B}^{(i+1)}$ may exist. To avoid this ambiguity, we introduce the function $\mathcal{C}\left(x, y\right) = \begin{cases} x, x \neq 0 \\ y, x = 0 \end{cases}$. Then the updating rule for $\mathbf{B}^{(i+1)}$ can be defined as

$$\mathbf{B}^{(i+1)} = \mathrm{sign}\left(\mathcal{C}(\nabla f(\mathbf{B}^{(i)}), \mathbf{B}^{(i)})\right) \qquad (6)$$

where $\mathrm{sign}(\cdot)$ is the sign function, $\mathcal{C}$ is applied in an element-wise manner.

Update Z The sub-problem with respect to $\mathbf{Z}$ is defined as follows

$$\max_{\mathbf{Z}} \quad \mathrm{Tr}\left(\mathbf{B}^\top\mathbf{Z}\right) \quad \text{s.t. } \mathbf{Z}^\top\mathbf{Z} = n\mathbf{I}_d \qquad (7)$$

Although there is a non-convex constraint on $\mathbf{Z}$, we fortunately show that (7) admits a closed-form solution. The singular value decomposition (SVD) of $\mathbf{B}$ is defined as $\mathbf{B} = \mathbf{P}\Sigma\mathbf{Q}^\top = \sum_{k=1}^{r}\sigma_k\mathbf{p}_k\mathbf{q}_k^\top$, where r is the rank of $\mathbf{B}$, $\sigma_1, \cdots, \sigma_r$ are the positive singular values, and $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_r]$, and $\mathbf{Q} = [\mathbf{q}_1, \cdots, \mathbf{q}_r]$ contain the left and right singular vectors, respectively. The closed-form solution of (7) can be characterized by the following theorem.

Theorem 1. *$\mathbf{Z} = \sqrt{n}\mathbf{P}\mathbf{Q}^\top$ is an optimal solution to optimization problem in* (7)*.*

Proof. (7) is the classic Orthogonal Procrustes problem. The proof of this theorem can be adapted from (Schönemann 1966). □

Algorithm 1 Co-Hashing (CoH)

Input: training dataset: $\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^n$, embedding dimensionality: d, regularization parameter: α.
Output: $\mathbf{U}, \mathbf{V}, \mathbf{B}$.

```
Initialize Z ∈ R^{n×d} by performing PCA on Y.
Initialize B = sign(Z).
repeat
    Compute the closed-form solution of U as U = (XX^⊤ + εI_p)^{-1} XB^⊤;
    Compute the closed-form solution of V as V = (YY^⊤ + εI_q)^{-1} YB^⊤;
    Update B via (6);
    Compute SVD decomposition of B, B = PΣQ^⊤;
    Update Z as Z = √n PQ^⊤;
until ε-optimal
```

Convergence Analysis

We theoretically analyze the convergence of CoH. The convergence of $\mathbf{B}$ sub-problem in CoH is presented as follows.

Lemma 1. *For the* $\mathbf{B}$ *sub-problem,* $\left\{\mathbf{B}^{(i)}\right\}$ *is the sequence of binary codes generated by the discrete optimization, i.e.,* (6)*, then* $\left\{f\left(\mathbf{B}^{(i)}\right)\right\}$ *converges.*

Proof. Since $\mathbf{S}$ is positive semidefinite, f is a convex function. Then, we have $f(\mathbf{B}) \geq \hat{f}_i(\mathbf{B})$. Besides, $\hat{f}_i(\mathbf{B}^{(i+1)}) \geq \hat{f}_i(\mathbf{B}^{(i)})$ and $\hat{f}_i(\mathbf{B}^{(i)}) = \hat{f}(\mathbf{B}^{(i)})$ hold. Based on them, we have $f(\mathbf{B}^{(i+1)}) \geq f(\mathbf{B}^{(i)})$, and $f(\mathbf{B}^{(i)})$ monotonically increases the objective function value of (4). Together with the fact $f(\mathbf{B}^{(i)})$ is upper bounded, it turns out that the sequence $\left\{f(\mathbf{B}^{(i)})\right\}$ converges. □

Based on the above lemma, we further have the following convergence theorem of CoH.

Theorem 2. *The alternate updating rules in Algorithm 1 monotonically decrease the objective function value of CoH, i.e.,* (3) *in each iteration, and Algorithm 1 will converge to a local minimum of CoH.*

Proof. Let $\{\mathbf{U}_\ell, \mathbf{V}_\ell, \mathbf{B}_\ell, \mathbf{Z}_\ell\}$ be the solution in the ℓ-th iteration generated by Algorithm 1, and $\varrho_\ell = \varrho(\mathbf{U}_\ell, \mathbf{V}_\ell, \mathbf{B}_\ell, \mathbf{Z}_\ell)$ be the corresponding objective function value. In Algorithm 1, at the $(\ell+1)$-th iteration, since each sub-problem is solved exactly, we have

$$\begin{aligned}\varrho\left(\mathbf{U}_\ell, \mathbf{V}_\ell, \mathbf{B}_\ell, \mathbf{Z}_\ell\right) &\geq \varrho\left(\mathbf{U}_{\ell+1}, \mathbf{V}_\ell, \mathbf{B}_\ell, \mathbf{Z}_\ell\right) \\ &\geq \varrho\left(\mathbf{U}_{\ell+1}, \mathbf{V}_{\ell+1}, \mathbf{B}_\ell, \mathbf{Z}_\ell\right) \geq \varrho\left(\mathbf{U}_{\ell+1}, \mathbf{V}_{\ell+1}, \mathbf{B}_{\ell+1}, \mathbf{Z}_\ell\right) \\ &\geq \varrho\left(\mathbf{U}_{\ell+1}, \mathbf{V}_{\ell+1}, \mathbf{B}_{\ell+1}, \mathbf{Z}_{\ell+1}\right) \qquad (8)\end{aligned}$$

Thus, the sequence $\{\varrho_\ell\}$ is monotonically decreasing. In addition, the objective function $\varrho(\mathbf{U}, \mathbf{V}, \mathbf{B}, \mathbf{Z})$ is lower-bounded by zero. Therefore, Algorithm 1 is guaranteed to converge to a local minimum of CoH. □

In addition, our empirical study shows that CoH quickly converges within around 30 iterations.

Computational Complexity Analysis

For the training part of CoH, the updates of $\mathbf{U}$ and $\mathbf{V}$ require $\mathcal{O}(p^3 + p^2 n)$, and $\mathcal{O}(q^3 + q^2 n)$, respectively. Updating $\mathbf{B}$ requires $\mathcal{O}\left(dpn + dqn\right)$. Besides, updating $\mathbf{Z}$ takes $\mathcal{O}\left(d^2 n + rdn\right)$. Given $n \gg p, q > d$, the total training computational cost of CoH is $\mathcal{O}((p^2 + q^2)n)$. In the testing stage, CoH requires $\mathcal{O}(pd)$ to generate the binary embedding, and then it takes $\mathcal{O}(\zeta n)$ to predict the d-bit code, where $\mathcal{O}(\zeta)$ denotes the computational complexity for d-bit calculations, and is obviously more efficient than d real-valued distance calculations.

Multi-Label Prediction via Cross-view Search

This section transforms multi-label prediction into a cross-view search problem. We first obtain its semantic binary embedding, i.e., $\text{sign}(\mathbf{U}^\top \mathbf{x})$ for a testing instance $\mathbf{x}$, and then perform cross-view kNN search among the labels in the constructed latent hamming space. The hamming distance between $\mathbf{x}$ and the label of the i-th training instance can be computed as $\|\text{sign}(\mathbf{U}^\top \mathbf{x}) - \text{sign}(\mathbf{V}^\top \mathbf{y}^{(i)})\|_F^2$. In practice, the distance could alternatively be computed as $\|\text{sign}(\mathbf{U}^\top \mathbf{x}) - \mathbf{b}^{(i)}\|_F^2$. The prediction scheme is detailed in Algorithm 2.

Algorithm 2 Testing Algorithm

Input: Testing instance: $\mathbf{x}$, number of NN: k, mappings of the input and output: $\mathbf{U}$, $\mathbf{V}$.
Output: $\mathbf{y}$.

1: Compute the embedding of $\mathbf{x}$, $\mathbf{b} = \text{sign}(\mathbf{U}^\top \mathbf{x})$;
2: Compute the embedding for training labels, $\mathbf{B}^{(l)} = \text{sign}(\mathbf{V}^\top \mathbf{Y})$; or directly use the embedding in training stage, $\mathbf{B}^{(l)} = \mathbf{B}$;
3: Obtain the set $\mathcal{N}_x$ by finding the k nearest neighbors of $\mathbf{x}$ in $\mathbf{B}^{(l)}$;
4: Obtain the prediction $\mathbf{y}$ by voting from $\mathcal{N}_x$.

Generalization Error Bound

CoH is characterized by a distribution $\mathcal{D}$ on the space of input and output $\mathcal{X} \times \{0,1\}^q$, where $\mathcal{X} \subseteq \mathbb{R}^p$. Let a sample $\{(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})\}$ be drawn i.i.d. from the distribution $\mathcal{D}$, where $\mathbf{y}^{(j)} \in \{0,1\}^q$ $(j \in \{1, \ldots, n\})$ is the ground-truth label vector. Assume n samples $D = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \cdots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}$ be drawn i.i.d. n times from the distribution $\mathcal{D}$, which is denoted by $D \sim \mathcal{D}^n$. Let $f^D_{knn_i}(\mathbf{x})$ represent the prediction of the i-th label for input $\mathbf{x}$ using CoH+kNN, which is trained on D. The performance of CoH+kNN: $(f^D_{knn_1}(\cdot), \cdots, f^D_{knn_q}(\cdot)) : \mathcal{X} \to \{0,1\}^q$ is measured in terms of its generalization error, which is its expected loss on a new example $(\mathbf{x}, \mathbf{y})$ drawn according to $\mathcal{D}$

$$E_{D\sim\mathcal{D}^n, (\mathbf{x},\mathbf{y})\sim\mathcal{D}}\Big(\sum_{i=1}^{q} \ell(y_i, f^D_{knn_i}(\mathbf{x}))\Big) \qquad (9)$$

where y_i denotes the i-th label and $\ell(y_i, f^D_{knn_i}(\mathbf{x}))$ represents the loss function of the i-th label. We then define the following loss function for analysis

$$\ell(y_i, f^D_{knn_i}(\mathbf{x})) = P(y_i \neq f^D_{knn_i}(\mathbf{x})) \qquad (10)$$

For i-th label, we further define the function as follows

$$\nu^i_j(\mathbf{x}) = P(y_i = j|\mathbf{x}),\ j \in \{0,1\} \qquad (11)$$

The Bayes optimal classifier b^* for i-th label is defined as

$$b^*_i(\mathbf{x}) = \arg\max_{j\in\{0,1\}} \nu^i_j(\mathbf{x}) \qquad (12)$$

Before deriving our results, we first present several important definitions and theorems.

Definition 1 (Covering Numbers, (Shawe-Taylor et al. 1998))**.** *Let* $(\mathcal{X}, d)$ *be a metric space,* A *be a subset of* $\mathcal{X}$ *and* $\varepsilon > 0$*. A set* $B \subseteq X$ *is an* ε*-cover for* A*, if for every* $a \in A$*, there exists* $b \in B$ *such that* $d(a,b) < \varepsilon$*. The* ε*-covering number of* A*,* $\mathcal{N}(\varepsilon, A, d)$*, is the minimal cardinality of an* ε*-cover for* A *(if there is no such finite cover then it is defined as* ∞*).*

Table 1: Statistics of eight real-world datasets.

Datasets	#Dataset	#Training	#Testing	#Features	#Labels	#Card-Label	Domain
CAL500	502	452	50	68	174	26.044	music
COREL5K	5,000	4,500	500	499	374	3.522	images
DELICIOUS	16,105	14,495	1,610	500	983	19.020	text
EUR-LEX	19,348	17,413	1,935	5,000	3,993	1.292	text
NUS-WIDE-V	269,648	161,789	107,859	128	81	1.869	images
NUS-WIDE-B	269,648	161,789	107,859	500	81	1.869	images
WIKI10	20,762	14,146	6,616	101,938	30,938	18.64	text
DELICIOUS-L	296,701	196,606	100,095	782,585	205,443	75.54	text

Definition 2 (Doubling Dimension, (Krauthgamer and Lee 2004; Kontorovich and Weiss 2014)). *Let $(\mathcal{X}, d)$ be a metric space, and let $\bar{\lambda}$ be the smallest value such that every ball in $\mathcal{X}$ can be covered by $\bar{\lambda}$ balls of half the radius. The doubling dimension of $\mathcal{X}$ is defined as : $ddim(\mathcal{X}) = \log_2(\bar{\lambda})$.*

Theorem 3 ((Krauthgamer and Lee 2004; Kontorovich and Weiss 2014)). *Let $(\mathcal{X}, d)$ be a metric space. The diameter of $\mathcal{X}$ is defined as $diam(\mathcal{X}) = \sup_{\mathbf{x},\mathbf{x}' \in \mathcal{X}} d(\mathbf{x}, \mathbf{x}')$. The ε-covering number of $\mathcal{X}$, $\mathcal{N}(\varepsilon, \mathcal{X}, d)$, is bounded by*

$$\mathcal{N}(\varepsilon, \mathcal{X}, d) \leq \Big(\frac{2diam(\mathcal{X})}{\varepsilon}\Big)^{ddim(\mathcal{X})} \tag{13}$$

We provide the following generalization error bound for CoH+1NN:

Theorem 4. *Given a metric space $(\mathcal{X}, d_{pro})$, assume function $\nu^i : \mathcal{X} \to \{0, 1\}$ is Lipschitz with constant L with respect to the sup-norm for each label. Suppose $\mathcal{X}$ has a finite doubling dimension: $ddim(\mathcal{X}) = \mathbb{D} < \infty$ and $diam(\mathcal{X}) = 1$. Let $D = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \cdots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}$ and $(\mathbf{x}, \mathbf{y})$ be drawn i.i.d. from the distribution $\mathcal{D}$. Then, we have*

$$E_{D\sim\mathcal{D}^n,(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\Big(\sum_{i=1}^{q} P(y_i \neq f^D_{1nn_i}(\mathbf{x}))\Big) \leq \sum_{i=1}^{q} 2P(b^*_i(\mathbf{x}) \neq y_i) + \frac{3qL(||\mathbf{U}||_F + ||\mathbf{V}||_F)}{n^{1/(\mathbb{D}+1)}} \tag{14}$$

Inspired by Theorem 19.5 in (Shalev-Shwartz and Ben-David 2014), we derive the following lemma for CoH+kNN:

Lemma 2. *Given metric space $(\mathcal{X}, d_{pro})$, assume function $\nu^i : \mathcal{X} \to \{0, 1\}$ is Lipschitz with constant L with respect to the sup-norm for each label. Suppose $\mathcal{X}$ has a finite doubling dimension: $ddim(\mathcal{X}) = \mathbb{D} < \infty$ and $diam(\mathcal{X}) = 1$. Let $D = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \cdots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}$ and $(\mathbf{x}, \mathbf{y})$ be drawn i.i.d. from the distribution $\mathcal{D}$. Then, we have*

$$E_{D\sim\mathcal{D}^n,(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\Big(\sum_{i=1}^{q} P(y_i \neq f^D_{knn_i}(\mathbf{x}))\Big) \leq \sum_{i=1}^{q}(1 + \sqrt{8/k})P(b^*_i(\mathbf{x}) \neq y_i) + \frac{q(6L(||\mathbf{U}||_F + ||\mathbf{V}||_F) + k)}{n^{1/(\mathbb{D}+1)}} \tag{15}$$

The following lemma reveals important statistical properties of CoH+1NN and CoH+kNN.

Corollary 1. *As n goes to infinity, the errors of CoH+1NN and CoH+kNN converge to the sum of twice the Bayes error and $1 + \sqrt{8/k}$ times the Bayes error over the labels, respectively.*

Experiments

In this section, we evaluate the performance of the proposed method for multi-label classification. All the computations are performed on a Red Hat Enterprise 64-Bit Linux workstation with 18-core Intel Xeon CPU E5-2680 2.80 GHz processor and 256 GB memory.

Experimental Setup

Datasets The experiments are conducted on eight multi-label real-world datasets, including six medium-sized datasets[1], i.e., CAL500, COREL5K, DELICIOUS, EUR-LEX, NUS-WIDE-V, NUS-WIDE-B, and two large-scale datasets[2], i.e., WIKI10, DELICIOUS-L. We split the training and testing set of two NUS-WIDE, WIKI10, DELICIOUS-L datasets by following publicly available experiment setting (Chua et al. 2009; Bhatia et al. 2015), while 10-fold cross-validation is applied for the other datasets. The statistics of the eight real-world datasets are summarized in Table 1.

Comparison Methods We compare the proposed method with seven state-of-the-art multi-label learning methods, i.e., BR (Tsoumakas, Katakis, and Vlahavas 2009), PLST (Tai and Lin 2012), CCA (Zhang and Schneider 2011), kNN, ML-kNN (Zhang and Zhou 2007), LM-kNN (Liu and Tsang 2015), and SLEEC (Bhatia et al. 2015). The linear classification/regression package LIBLINEAR (Fan et al. 2008) with ℓ_2-regualrized logistic regression is adopted to train the classifier for BR. Following the experimental settings (Liu and Tsang 2015), we set $\eta = 0.4$ in LM-kNN, and $C = 10$ in BR and LM-kNN. According to the original settings (Bhatia et al. 2015), we set the number of the clusters as $\lfloor n/6000 \rfloor$ and the number of learners as 15 for SLEEC. In the proposed CoH, the regularization parameter α is empirically set to 100 for the two NUS-WIDE datasets, and 1 for the other datasets. The dimension of the common subspace in SLEEC and the proposed CoH is empirically set to 50 for

[1] http://mulan.sourceforge.net

[2] http://manikvarma.org/downloads/XC/XMLRepository.html

Table 2: Predictive performance comparison on six medium-sized real-world datasets. The best results are in bold. N/A denotes not available.

Datasets	Hamming Loss ↓							
	BR	PLST	CCA	kNN	ML-kNN	LM-kNN	SLEEC	CoH
CAL500	.1390 ± .0113	.1376 ± .0043	**.0916 ± .0022**	.1453 ± .0047	.1398 ± .0055	.1493 ± .0031	.1489 ± .0056	.1124 ± .0037
COREL5K	.0174 ± .0057	.0094 ± .0001	N/A	.0095 ± .0001	.0094 ± .0001	.0095 ± .0002	.0092 ± .0001	**.0082 ± .0002**
DELICIOUS	.0181 ± .0002	.0184 ± .0002	N/A	.0187 ± .0002	.0182 ± .0002	.0179 ± .0002	.0176 ± .0002	**.0162 ± .0003**
EUR-LEX	.0331 ± .0015	.0014 ± .0001	N/A	.0011 ± .0001	.0011 ± .0001	.0009 ± .0000	.0010 ± .0000	**.0008 ± .0000**
NUS-WIDE-V	.0209	.2140	N/A	.0213	.1590	.0213	.0203	**.0165**
NUS-WIDE-B	.0215	.0217	N/A	.0228	.0836	.0216	.0213	**.0193**
Datasets	Example-F1 ↑							
	BR	PLST	CCA	kNN	ML-kNN	LM-kNN	SLEEC	CoH
CAL500	.3446 ± .0150	.3062 ± .0108	.3520 ± .0191	.3561 ± .0131	.3216 ± .0180	.3511 ± .0161	.3099 ± .0315	**.3602 ± .0216**
COREL5K	.0781 ± .0166	.0587 ± .0054	N/A	.0223 ± .0056	.0178 ± .0069	.1295 ± .0092	.0839 ± .0061	**.1996 ± .0108**
DELICIOUS	.2093 ± .0051	.1131 ± .0039	N/A	.1878 ± .0046	.1518 ± .0044	.2553 ± .0043	.2197 ± .0044	**.2745 ± .0033**
EUR-LEX	**.4013 ± .0098**	.0725 ± .0083	N/A	.3409 ± .0057	.3005 ± .0049	.3818 ± .0045	.3705 ± .0058	.3915 ± .0076
NUS-WIDE-V	.1255	.0097	N/A	.1382	.1476	.0982	.1703	**.2838**
NUS-WIDE-B	.0981	.0825	N/A	.1263	.0815	.0877	.1528	**.2622**
Datasets	Micro-F1 ↑							
	BR	PLST	CCA	kNN	ML-kNN	LM-kNN	SLEEC	CoH
CAL500	.3448 ± .0161	.3032 ± .0095	.3537 ± .0213	.3593 ± .0127	.3184 ± .0162	.3542 ± .0157	.3077 ± .0321	**.3664 ± .0234**
COREL5K	.0956 ± .0260	.0801 ± .0081	N/A	.0321 ± .0075	.0278 ± .0117	.1670 ± .0137	.1215 ± .0093	**.2127 ± .0111**
DELICIOUS	.2447 ± .0051	.1423 ± .0058	N/A	.2154 ± .0047	.1738 ± .0047	**.3104 ± .0058**	.2532 ± .0035	.3011 ± .0039
EUR-LEX	.4266 ± .0070	.1059 ± .0115	N/A	.4011 ± .0061	.3489 ± .0055	.4344 ± .0051	.4235 ± .0056	**.4689 ± .0085**
NUS-WIDE-V	.2584	.1986	N/A	.2772	.2826	.2093	.3249	**.3282**
NUS-WIDE-B	.2162	.1910	N/A	.2458	.1768	.1993	.2822	**.3119**
Datasets	Macro-F1 ↑							
	BR	PLST	CCA	kNN	ML-kNN	LM-kNN	SLEEC	CoH
CAL500	.0781 ± .0115	.0410 ± .0019	.0917 ± .0022	.0971 ± .0060	.0534 ± .0034	.1001 ± .0067	.0462 ± .0081	**.1173 ± .0115**
COREL5K	.0276 ± .0039	.0111 ± .0017	N/A	.0052 ± .0014	.0086 ± .0036	.0266 ± .0033	.0240 ± .0022	**.0423 ± .0049**
DELICIOUS	.0933 ± .0042	.0283 ± .0008	N/A	.0550 ± .0024	.0481 ± .0018	.1383 ± .0074	.0702 ± .0014	**.0993 ± .0040**
EUR-LEX	.1039 ± .0066	.0204 ± .0031	N/A	.0877 ± .0032	.0635 ± .0019	.0919 ± .0029	.0805 ± .0030	**.1053 ± .0046**
NUS-WIDE-V	.0266	.0162	N/A	.0682	.0159	.0171	.0437	**.0498**
NUS-WIDE-B	.0202	.0139	N/A	.0373	.0206	.0143	.0302	**.0454**

Table 3: Efficiency comparison on six medium-sized real-world datasets. The training and testing time are recorded in seconds. N/A denotes not available.

Method	CAL500		COREL5K		DELICIOUS		EUR-LEX		NUS-WIDE-V		NUS-WIDE-B	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
BR	1.33	1.78×10^{-3}	8.63	6.28×10^{-3}	120.70	0.03	2.14×10^{3}	0.49	222.21	0.07	511.83	0.02
PLST	0.02	4.00×10^{-3}	0.82	0.12	9.15	0.91	106.26	13.94	.98	0.07	2.49	0.14
CCA	895.50	1.43×10^{4}	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
kNN	N/A	0.02	N/A	3.24	N/A	30.95	N/A	393.87	N/A	3.70×10^{3}	N/A	1.24×10^{4}
ML-kNN	1.03	0.19	23.01	4.02	178.49	29.93	1.03×10^{3}	155.60	5.57×10^{3}	3.66×10^{3}	3.02×10^{4}	2.02×10^{4}
LM-kNN	13.23	0.03	1.20×10^{3}	2.35	2.40×10^{4}	56.31	4.87×10^{3}	316.75	1.64×10^{4}	4.83×10^{3}	2.08×10^{4}	3.17×10^{3}
SLEEC	27.99	0.06	736.45	3.93	3.28×10^{3}	16.62	5.23×10^{3}	41.73	9.72×10^{3}	914.56	1.14×10^{4}	1.32×10^{3}
CoH	0.12	4.00×10^{-3}	3.82	0.05	22.31	0.78	115.23	1.34	44.52	246.50	52.74	248.56

the two NUS-WIDE, WIKI10, DELICIOUS-L datasets, and 100 for the others. Following the similar settings in (Zhang and Zhou 2007; Bhatia et al. 2015), the k in kNN search is selected using 10-fold cross validation over the range $\{1, 5, 10, 20\}$ for all kNN-based methods. The running time of MMOC (Zhang and Schneider 2012) on most datasets is more than one week, thus we do not report its results. Our empirical study shows that CoH performs well in a wide range of the parameters.

Evaluation Metric Following (Guo and Schuurmans 2013; Liu and Tsang 2015; Bhatia et al. 2015), we consider the widely-used metrics to evaluate the prediction performance of all the methods, i.e., Hamming loss, Example-F1, Micro-F1, Macro-F1 for medium-sized datasets, and Precision@5, nDCG@5 for large-scale datasets.

Results on Medium-Sized Datasets

Performance We conduct the performance evaluation on six medium-sized multi-label datasets. Table 2 reports the prediction performance of all the methods, in terms of Ham-

Table 4: Performance comparison on two large-scale real-world datasets. The training and testing time are in seconds.

Dataset	WIKI10				DELICIOUS-L			
	Precision@5 ↑	nDCG@5 ↑	Training(s)	Testing(s)	Precision@5 ↑	nDCG@5 ↑	Training(s)	Testing(s)
SLEEC	.6270	.6813	1.97×10^3	49.54	.3943	.4137	5.64×10^4	2.64×10^3
CoH	.6334	.6678	2.59×10^2	6.72	.4136	.4256	2.39×10^3	478.58

ming loss, Example-F1, Micro-F1, Macro-F1. From Table 2, we observe that

- The proposed CoH generally has the best performance on the six medium-sized datasets. For example, on NUS-WIDE-B dataset, in term of Hamming loss, Example-F1, Micro-F1, Macro-F1, CoH improves the best results of the baselines by 0.2%, 11.0%, 2.9%, 1.5%, respectively. The above results demonstrate the superior performance of the proposed CoH, and corroborate our theoretical results.
- Among kNN-based baselines, LM-kNN and SLEEC achieve the better performance, verifying their effectiveness. The conventional kNN method usually outperforms ML-kNN.
- CCA can only perform on the small CAL500 dataset due to its very expensive time cost. PLST generally underperforms on all the datasets, which is consistent with the empirical results in (Tai and Lin 2012; Liu and Tsang 2015). BR is inferior than the proposed CoH, mainly because it fails to consider the correlations between labels.

Time Table 3 illustrates both the training and testing time of all the methods. The conclusions that can be drawn from the table are as follows:

- On the aspect of the training time, we can see that PLST is the fastest in training, but performs poorly. The proposed CoH generally takes second place for training efficiency, that is much faster than the other kNN-based multi-label methods. For instance, on the NUS-WIDE-B dataset, CoH is significantly more efficient, nearly 400 times faster than LM-kNN and 200 times faster than SLEEC. CCA is the most time consuming for its expensive encoding procedure.
- On the aspect of the testing time, the proposed CoH is the most efficient among the kNN-based methods, due to the efficient computation in the Hamming space. SLEEC is also fast on the two NUS-WIDE datasets, as it uses the clustering technique to accelerate the prediction. CoH is nearly 4 times faster than SLEEC, and at least 10 times faster than the other kNN-based methods on the two NUS-WIDE datasets. In addition, BR and PLST are also efficient because of their simple decoding scheme, which, however, fails to obtain good performance. CCA is significantly slower than the other methods.

Results on Large-Scale Datasets

We conduct the performance evaluation on two large-scale datasets, i.e., WIKI10, DELICIOUS-L. The label vectors in large-scale datasets are highly sparse, thus we can further enforce the sparsity constraint on the transformation matrices $\mathbf{U}$ and $\mathbf{V}$ in CoH, which helps to select the most meaningful labels and reduce the computation cost. The efficient feature generating paradigm (Tan, Tsang, and Wang 2014; Liu and Tsang 2017) is used to select labels, and it can easily handle millions of labels. The performances of SLEEC and CoH are reported in Table 4. From Table 4, we can see that CoH can achieve comparable performance, and have less training and testing time.

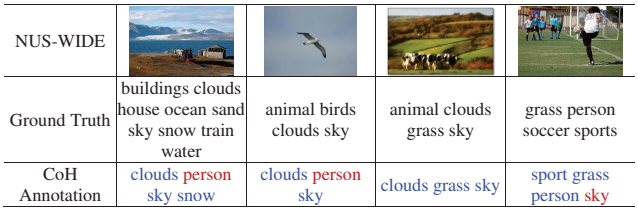

Figure 1: Several multi-label image annotation examples on NUS-WIDE-V dataset. For each image, we show the ground truth annotations, and the labels predicted by the proposed CoH.

Case Study

We present a case study where the proposed CoH is applied to a multi-label image annotation application. We apply CoH on the NUS-WIDE-V dataset, and illustrate the annotation results of several randomly selected images, as shown in Figure 1. It can be seen that CoH correctly predicts most labels for these images. This case study suggests that CoH works well in real-world image annotation applications.

Conclusion

This paper provides a cross-view insight into the multi-label prediction to fully discover the correlations between the input and output. The proposed CoH learns a latent discrete code space shared by both input and output. CoH performs fast multi-label prediction via the efficient cross-view search in Hamming space. We empirically show that, on the NUS-WIDE datasets, CoH improves the prediction efficiency of LM-kNN and SLEEC by around 10 and 4 times, respectively. The proposed CoH is theoretically shown to diminish the generalization error bounds of multi-label prediction. The empirical studies verify our theoretical results, and the proposed CoH achieves the superior performance than the state-of-the-art methods with less computation cost.

Acknowledgments

This research is supported by the National Science Foundation of China (Grant No. 61273251, 61673220), the ARC Future Fellowship FT130100746, ARC grant LP150100671,

DP180100106, and partially supported under the Data Science and Artificial Intelligence Center (DSAIR) at the Nanyang Technological University.

References

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR* 7:2399–2434.

Bhatia, K.; Jain, H.; Kar, P.; Varma, M.; and Jain, P. 2015. Sparse local embeddings for extreme multi-label classification. In *NIPS*, 730–738.

Chen, Y.-N., and Lin, H.-T. 2012. Feature-aware label space dimension reduction for multi-label classification. In *NIPS*, 1529–1537.

Chua, T.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y. 2009. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*.

Dhillon, P.; Foster, D. P.; and Ungar, L. H. 2011. Multi-view learning of word embeddings via cca. In *NIPS*, 199–207.

Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *JMLR* 9:1871–1874.

Gong, Y.; Lazebnik, S.; Gordo, A.; and Perronnin, F. 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *TPAMI* 35(12):2916–2929.

Guo, Y., and Schuurmans, D. 2013. Multi-label classification with output kernels. In *ECML/PKDD*, 417–432.

Guo, Y., and Xiao, M. 2012. Cross language text classification via subspace co-regularized multi-view learning. In *ICML*, 1615–1622.

Hsu, D.; Kakade, S.; Langford, J.; and Zhang, T. 2009. Multi-label prediction via compressed sensing. In *NIPS*, 772–780.

Kontorovich, A., and Weiss, R. 2014. Maximum margin multiclass nearest neighbors. In *ICML*, 892–900.

Krauthgamer, R., and Lee, J. R. 2004. Navigating nets: Simple algorithms for proximity search. In *SODA*, 798–807.

Lai, H.; Yan, P.; Shu, X.; Wei, Y.; and Yan, S. 2016. Instance-aware hashing for multi-label image retrieval. *TIP* 25(6):2469–2479.

Liu, W., and Tsang, I. W. 2015. Large margin metric learning for multi-label prediction. In *AAAI*, 2800–2806.

Liu, W., and Tsang, I. W. 2017. Making decision trees feasible in ultrahigh feature and label dimensions. *JMLR* 18(81):1–36.

Liu, W.; Tsang, I. W.; and Müller, K.-R. 2017. An easy-to-hard learning paradigm for multiple classes and multiple labels. *JMLR* 18(94):1–38.

Quadrianto, N., and Lampert, C. H. 2011. Learning multi-view neighborhood preserving projections. In *ICML*, 425–432.

Schönemann, P. H. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika* 31(1):1–10.

Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding Machine Learning: From Theory to Algorithms*. New York, USA: Cambridge University Press.

Shawe-Taylor, J.; Bartlett, P. L.; Williamson, R. C.; and Anthony, M. 1998. Structural risk minimization over data-dependent hierarchies. *TIT* 44(5):1926–1940.

Shen, F.; Shen, C.; Liu, W.; and Tao Shen, H. 2015. Supervised discrete hashing. In *CVPR*, 37–45.

Shen, X.; Liu, W.; Tsang, I. W.; Shen, F.; and Sun, Q. 2017a. Compressed k-means for large-scale clustering. In *AAAI*, 2527–2533.

Shen, X.; Shen, F.; Sun, Q.-S.; Yang, Y.; Yuan, Y.-H.; and Shen, H. T. 2017b. Semi-paired discrete hashing: Learning latent hash codes for semi-paired cross-view retrieval. *TCYB* 47(12):4275–4288.

Tai, F., and Lin, H.-T. 2012. Multilabel classification with principal label space transformation. *Neural Computation* 24(9):2508–2542.

Tan, M.; Tsang, I. W.; and Wang, L. 2014. Towards ultrahigh dimensional feature selection for big data. *JMLR* 15(1):1371–1429.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2009. Mining multi-label data. In *Data mining and knowledge discovery handbook*. 667–685.

Wang, J.; Liu, W.; Kumar, S.; and Chang, S.-F. 2016. Learning to hash for indexing big data-a survey. *Proceedings of the IEEE* 104(1):34–57.

Weiss, Y.; Torralba, A.; and Fergus, R. 2009. Spectral hashing. In *NIPS*, 1753–1760.

Zhang, Y., and Schneider, J. 2011. Multi-label output codes using canonical correlation analysis. In *AISTATS*, 873–882.

Zhang, Y., and Schneider, J. 2012. Maximum margin output coding. In *ICML*, 1575–1582.

Zhang, M.-L., and Zhou, Z.-H. 2007. ML-KNN: A lazy learning approach to multi-label learning. *PR* 40(7):2038–2048.

Zhang, M.-L., and Zhou, Z.-H. 2014. A review on multi-label learning algorithms. *TKDE* 26(8):1819–1837.

Zhao, F.; Huang, Y.; Wang, L.; and Tan, T. 2015. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*, 1556–1564.