This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Deep Determinantal Point Process for Large-Scale Multi-Label Classification

Pengtao Xie[*†], Ruslan Salakhutdinov[*], Luntian Mou[§] and Eric P. Xing[†]
[*]Machine Learning Department, Carnegie Mellon University, USA
[†]Petuum Inc.
[§]Beijing Key Laboratory of Traffic Engineering, Beijing University of Technology, China
{pengtaox,rsalakhu}@cs.cmu.edu, ltmou@bjut.edu.cn, eric.xing@petuum.com

Abstract

We study large-scale multi-label classification (MLC) on two recently released datasets: Youtube-8M and Open Images that contain millions of data instances and thousands of classes. The unprecedented problem scale poses great challenges for MLC. First, finding out the correct label subset out of exponentially many choices incurs substantial ambiguity and uncertainty. Second, the large data-size and class-size entail considerable computational cost. To address the first challenge, we investigate two strategies: capturing label-correlations from the training data and incorporating label co-occurrence relations obtained from external knowledge, which effectively eliminate semantically inconsistent labels and provide contextual clues to differentiate visually ambiguous labels. Specifically, we propose a Deep Determinantal Point Process (DDPP) model which seamlessly integrates a DPP with deep neural networks (DNNs) and supports end-to-end multi-label learning and deep representation learning. The DPP is able to capture label-correlations of any order with a polynomial computational cost, while the DNNs learn hierarchical features of images/videos and capture the dependency between input data and labels. To incorporate external knowledge about label co-occurrence relations, we impose relational regularization over the kernel matrix in DDPP. To address the second challenge, we study an efficient low-rank kernel learning algorithm based on inducing point methods. Experiments on the two datasets demonstrate the efficacy and efficiency of the proposed methods.

1. Introduction

Recently two large-scale multi-label datasets have been released: YouTube-8M [1] and Open Images [32]. The YouTube-8M dataset contains about 8 million videos, each associated with multiple labels coming from 4800 classes. These videos are 0.5 million hours long and contain 1.9 billion frames. The Open Images dataset contains about 9 million images, each annotated with multiple class labels. The total number of unique labels is 6012. The scale of these two datasets is much larger than previous multi-label datasets such as NUS-WIDE [12], PASCAL VOC[1], SUN attributes [42], Mediamill[2], in terms of both the number of data instances and the number of classes, bringing in great challenges for multi-label classification (MLC). For each data instance x, a subset of labels $\mathcal{S} \subseteq \mathcal{Y} = \{1, \cdots, K\}$ are to be selected to annotate x, where $\mathcal{S}$ has exponentially many (2^K) choices. The number (K) of unique labels is 4800 in Youtube-8M and 6012 in Open Images, which result in a tremendous combinatorial search space. Finding out the correct label-subset $\mathcal{S}^*$ in this space requires not only efficient algorithms that can tackle this NP-hard problem in polynomial time, but also modeling techniques that can effectively resolve the ambiguity and uncertainty when picking up $\mathcal{S}^*$ from 2^K choices.

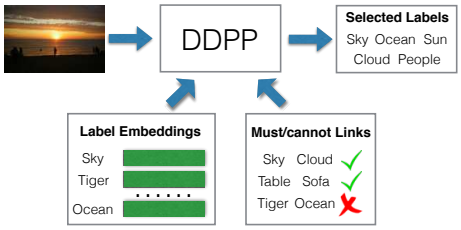

Figure 1. DDPP for multi-label classification. The inputs of DDPP include the image (or video), the embedding vectors of labels and (optional) must/cannot-links regarding label co-occurrence, and the output is a subset of selected labels. DDPP captures the correlation among labels using DPP, characterizes the dependency between the image and labels using DNN, and incorporates the must/cannot-links via relational regularization.

In this paper, we aim at addressing these challenges. To correctly hit $\mathcal{S}^*$ from 2^K candidates, we investigate two strategies: (1) capturing high-order correlations among labels from the training data; (2) incorporating external prior knowledge about label co-occurrence relations. Both strategies aim at ruling out candidate subset $\mathcal{S}$ where the labels are semantically and contextually inconsistent, thus effec-

[1]http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
[2]http://mulan.sourceforge.net/datasets-mlc.html

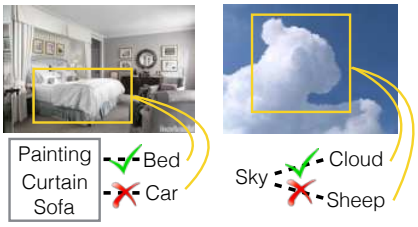

Figure 2. Labels that are visually less distinguishable can be well discriminated by leveraging label correlations learned from training data (left) and label co-occurrence relations obtained from external knowledge (right).

tively shrinking the search space and reducing the ambiguity/uncertainty of hitting $\mathcal{S}^*$. As a complement of visual features, the dependency relationship between labels provides additional clues for correct prediction. Visually hard-to-distinguish labels could be well differentiated based on label-correlations. Fig. 2 presents two examples. In the left figure, the image region marked with yellow box can be predicted either as *bed* or *car*, with similar confidence. But compared with *car*, *bed* possesses stronger correlation with other predicted labels including *painting*, *curtain*, *sofa*, which collectively form an indoor scene. Leveraging label-correlation, we can correctly assign *bed* to the region. The figure on the right shows a similar example, where the label co-occurrence relations obtained from external knowledge, such as "sky is likely to co-occur with cloud, not sheep", are leveraged to discriminate the two visually similar labels *cloud* and *sheep*.

Characterizing label correlation [18, 49, 23, 68, 52, 4, 38, 56] has been widely studied in MLC. While existing methods have demonstrated success on tens or hundreds of labels, they are less capable to deal with thousands of labels. To retain computational efficiency, many approaches [18, 23, 31, 9, 38] limit the order of label-correlations to be less than three and ignore high-order ones. High-order relations can represent semantics that is difficult to be captured in low-order relations. For instance, considering two label sets $S_1 = \{bed, desk, sofa\}$ and $S_2 = \{bed, desk, woods\}$, both of them exhibit strong second-order correlations: every two labels therein are correlated. But S_1 also possesses a third-order correlation: the three words collectively represent a *furniture* topic, while S_2 does not. Several methods [27, 22, 4, 56] are proposed to capture high-order correlations, but they either make strong assumptions that degrade classification performance [27, 22, 56], or incur substantial computational cost [4].

To address this issue, we develop a Deep Determinantal Point Process (DDPP) model (Fig. 1) that is not only highly expressive to capture high-order label-correlations, but also computationally efficient. In DPP [34], the selection of label-subset $\mathcal{S}$ is based upon the volume of the parallelepiped formed by nonlinear feature vectors of labels in $\mathcal{S}$ [33]. The volume is collectively determined by the global relationship among all vectors, rather than their pairwise relations, hence is able to characterize high-order label dependency. The volume can be computed efficiently as the determinant of a kernel matrix between the embedding vectors of labels, with polynomial (specifically, cubic) complexity. Other than capturing label-correlation, DDPP seamlessly integrates deep neural networks (DNNs) to measure the dependency between input data and labels. The DNNs learn features from input images/videos and can be trained in an end-to-end fashion.

Besides label-correlations derived from training data, we can leverage the co-occurrence relations obtained from external knowledge to distinguish visually ambiguous labels. For simplicity, we consider binary relations: two labels can have a must-link suggesting they are very likely to co-occur, or a cannot-link indicating that they barely co-occur. Many knowledge sources provide such relations. For instance, WordNet [41] contains a lot of "A is a B" relations, such as *apple-fruit* and *tiger-animal*. In this case, if label A is assigned to the input data, so should be B. To incorporate these must/cannot links, we impose relational regularization over the kernel matrix in DDPP. According to the property of DPP [33], two labels represented with vectors $\mathbf{a}_i$ and $\mathbf{a}_j$ have a better chance to be co-selected if $k(\mathbf{a}_i, \mathbf{a}_j)$ – the output of the kernel function over them – is small. If label i and j have a must-link, the regularizer encourages $k(\mathbf{a}_i, \mathbf{a}_j)$ to be small to promote co-selection. If they share a cannot-link, $k(\mathbf{a}_i, \mathbf{a}_j)$ is favored to be large.

Lastly, to make DDPP scale to thousands of labels and millions of data instances presenting in Youtube-8M and Open Images, we study an efficient algorithm. In DDPP, for each of the N training instances, one needs to evaluate the determinant and inverse of a data-dependent $K \times K$ kernel matrix (where K is the number of classes) with a substantial $O(NK^3)$ cost. To address this problem, we investigate a scalable low-rank kernel learning algorithm based on inducing point methods [48] which seek a low-rank parameterization of the kernel matrix using auxiliary points. Thereafter, the Woodbury matrix identity can be applied to compute matrix inverse, reducing the cost from $O(K^3)$ to $O(M^3)$, where $M \ll K$ is the number of inducing points.

The major contributions of this paper are:

- We propose a deep DPP method to perform large-scale MLC. DDPP is able to capture high-order label-correlations with polynomial computational complexity and facilitates end-to-end deep feature learning.
- To incorporate external knowledge regarding label co-occurrence relations, we propose to impose relational regularization over the kernel matrix of DDPP.
- We study a low-rank kernel learning algorithm to scale DDPP to thousands of labels and millions of instances.
- Experiments on YouTube-8M and Open Images

demonstrate the effectiveness and efficiency of the proposed methods.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 introduces the DDPP methods and scalable algorithms. Section 4 presents experimental results and Section 5 concludes the paper.

2. Related Works

Multi-label classification (MLC) has been widely studied in computer vision and and machine learning [30, 8, 7, 57, 62, 16, 17, 50, 44, 61, 69, 29, 57, 64, 70]. We present a brief review from the following perspectives.

Capturing Label Correlations Many approaches have been proposed to capture label correlations, based on graphical models, latent space learning and class chaining. *Graphical model (GM) based approaches* leverage undirected [18, 49, 23, 52, 4, 38] and directed [68, 22] GMs to capture the dependency structure among labels. To retain computational efficiency, many methods limit the order of correlation to be less than three [18, 23, 31, 9, 38] or assume label-dependency is linear [22]. To capture high-order nonlinear correlation, Belanger and McCallum [4] learn a neural network which takes labels as input and produces their dependency score. This method is computationally inefficient: during training, an iterative inference procedure needs to be performed over each data instance. In [49, 68, 52], hypergraph spectral learning, Bayesian network structure learning and max-margin structure learning are studied respectively to learn high-order label-correlations. These methods lack the flexibility to perform deep visual feature learning in an end-to-end manner. *Latent space learning approaches* propose to capture label dependency in a shared hidden space. Linear subspaces based on low-rankness [27, 28] and conditional Bernoulli mixture [37] cannot capture nonlinear correlations. Nonlinear spaces induced by Restricted Boltzmann machine [39] and deep neural networks [58, 4, 65] make a strong assumption: the labels are independent conditioned on the latent space, which may not hold in practice and leads to inferior performance. *Class chaining methods* [11, 45, 56] organize the classes into a linear chain and predict them in a greedy manner: the prediction of class i depends on classes $1, \cdots, i-1$. The overall prediction relies on the class order, which is difficult to specify. To address this issue, they propose to average the predictions over a randomly chosen set of class permutations, which substantially increases computational cost. Another disadvantage is early prediction errors will be propagated to subsequent classes.

Incorporating Prior Knowledge Several approaches leverage external knowledge, such as label hierarchy [46, 55], label correlation statistics [23, 39], object bounding boxes [63], to boost MLC performance. In [23], a linear projection matrix is designed to encode prior knowledge of label-correlations. In [39], a prior distribution is defined to encourage labels with large cosine similarity (computed from external Wikipedia corpora) to be co-selected. These knowledge-incorporation methods are model-specific, and are not applicable to DPP. We propose a new approach tailored to the property of DPP.

Scaling to Large Number of Labels To solve MLC problems that have a large number of classes, approaches based on label space dimension reduction (LSDR) or section [25, 10, 3, 51, 6] and label hierarchy learning [5, 2, 43] have been studied. LSDR encodes the high-dimensional label vectors into low-dimensional coding vectors. Then predictive models are trained from instance features to codes, which are decoded to recover the original labels. These methods lack the flexibility to incorporate label co-occurrence knowledge since labels are transformed into a latent space. In [2], a hierarchy of labels is learned via recursive node-partitioning, which reduces the prediction cost to sub-linear. The limitation of this approach is its inflexibility to capture label correlations.

DPP for Computer Vision and Deep Kernel Learning DPP has been applied for several vision tasks, including video summarization [20, 67], pedestrian detection [36], among others. Our work represents the first one using DPP [33] for multi-label classification. Using deep neural networks to parameterize kernel function is studied in [60, 67]. In our method, DNN is utilized to parameterize a conditional kernel.

3. Methods

In this section, we present the DDPP model and algorithms for parameter learning and label-subset inference.

3.1. Deep Determinantal Point Process

MLC can be formulated as a subset selection problem: given the input data x and the K classes $\mathcal{Y} = \{1, \cdots, K\}$, we aim at selecting a subset $\mathcal{S} \subseteq \mathcal{Y}$ of labels that best describes x. This is a NP-hard problem since $\mathcal{S}$ has infinitely many choices. The selection of $\mathcal{S}$ needs to consider two factors: (1) the labels in $\mathcal{S}$ should be highly relevant to x; (2) the labels should exhibit strong correlation. As stated earlier, incorporating label-correlation effectively eliminates semantically-inconsistent labels and reduces the search space of $\mathcal{S}$. However, it greatly complicates computation. One popular model that is able to simultaneously incorporate these two factors is Conditional Random Field (CRF) [35], where the dependency between input data and labels, together with correlation among labels, is characterized by potential functions. However, CRF involves a partition function which sums over exponentially many configurations and makes inference and learning extremely hard

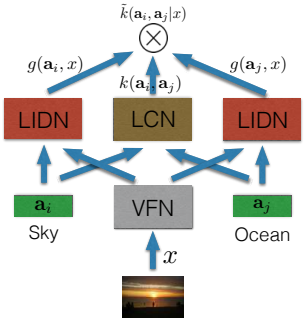

Figure 3. In DDPP, the conditional kernel function $\tilde{k}(\mathbf{a}_i, \mathbf{a}_j|x)$ is the product of a label-label kernel function $k(\mathbf{a}_i, \mathbf{a}_j)$ and two label-input score functions $g(\mathbf{a}_i, x)$ and $g(\mathbf{a}_j, x)$. $k(\mathbf{a}_i, \mathbf{a}_j)$ is characterized by a label-correlation network (LCN) and $g(\mathbf{a}_i, x)$ is represented by a visual feature network (VFN) and a label-input dependency network (LIDN).

when the potential function is of high-order. We aim at designing methods that are able to capture correlations of any-order, but also computationally tractable.

To achieve this goal, we resort to Determinantal Point Process (DPP) [34], which defines a probability distribution over subsets. Given a set of items $\{\mathbf{a}_i\}_{i=1}^K$, each represented with a vector $\mathbf{a}$, DPP computes a kernel matrix $\mathbf{L} \in \mathbb{R}^{K\times K}$, where $L_{ij} = k(\mathbf{a}_i, \mathbf{a}_j)$ and $k(\cdot,\cdot)$ is a kernel function. Then the probability over a subset of items indexed by $\mathcal{S} \subseteq \{1, \cdots, K\}$ can be defined as

$$p(\mathcal{S}) = \frac{\det(\mathbf{L}_\mathcal{S})}{\det(\mathbf{L}+\mathbf{I})} \tag{1}$$

where $\mathbf{L}_\mathcal{S} \equiv [\mathbf{L}_{ij}]_{i,j\in\mathcal{S}}$ denotes the restriction of $\mathbf{L}$ to the entries indexed by elements of $\mathcal{S}$ and $\det(\cdot)$ denotes the determinant of a matrix and $\mathbf{I}$ is an identity matrix. The determinant enables DPP to capture the high-order label-dependency. To understand this, we first present the geometry interpretation of $\det(\mathbf{L}_\mathcal{S})$. According to the kernel trick [47], $k(\mathbf{a}_i, \mathbf{a}_j)$ can be written as $\phi(\mathbf{a}_i)^\top\phi(\mathbf{a}_j)$, where $\phi(\cdot)$ is a reproducing kernel feature map [47]. Then $\det(\mathbf{L}_\mathcal{S})$ is essentially the volume of the parallelepiped formed by the vectors $\{\phi(\mathbf{a}_i)|i \in \mathcal{S}\}$ [33]. The size of the volume is collectively determined by all these vectors in a global way, which hence captures the high-order correlation among them. Another way to understand why determinant entails high-order correlation is to expand $\det(\mathbf{L}_\mathcal{S})$ as a sum of terms each involving the multiplication of $|\mathcal{S}|$ kernel function values, which hence captures label-correlations of $|\mathcal{S}|$-th order. While able to represent high-order correlation, DPP is computationally efficient. DPP's normalizer $\det(\mathbf{L}+\mathbf{I})$ can be computed in polynomial (cubic) time, as opposed to the exponential complexity in CRF.

In the context of MLC, we apply DPP to capture the correlation among labels: given the representations of K labels $\{\mathbf{a}_i\}_{i=1}^K$ (we will discuss how to learn these representations later on), we compute the kernel matrix $\mathbf{L}$ and define probability over label subset according to Eq.(1). For label-label kernel function $k(\mathbf{a}_i, \mathbf{a}_j)$, we parameterize it using a *label correlation network* (LCN) where the inputs are $\mathbf{a}_i$ and $\mathbf{a}_j$ and the output is a scalar indicating the correlation of the two labels, as shown in Fig. 3. As stated earlier, the selection of labels relies not only on the correlation among labels, but also the dependency between input data and labels. We use a deep neural network (Fig. 3) to define a score function $g(\mathbf{a}_i, x)$ to measure the dependency between input image/video x and label i. x is first fed into a *visual feature network* (VFN) to extract deep features, which then together with the representation of a label are inputted into a *label-input dependency network* (LIDN) to generate a dependency score. To enable end-to-end training of the VFN, we incorporate $g(\mathbf{a}_i, x)$ into the kernel function in DPP. On top of the kernel function $k(\mathbf{a}_i, \mathbf{a}_j)$ measuring the correlation between label i and j, we define a new kernel

$$\tilde{k}(\mathbf{a}_i, \mathbf{a}_j|x) = g(\mathbf{a}_i, x)k(\mathbf{a}_i, \mathbf{a}_j)g(\mathbf{a}_j, x) \tag{2}$$

which is conditioned on the input x. $\tilde{k}(\mathbf{a}_i, \mathbf{a}_j|x)$ simultaneously captures label-input dependency and label-label correlation. Under this conditional kernel parameterized by deep networks, we obtain a Deep DPP:

$$p(\mathcal{S}|x) = \frac{\det(\mathbf{L}_\mathcal{S}(x))}{\det(\mathbf{L}(x)+\mathbf{I})} \tag{3}$$

where $\mathbf{L}_{ij}(x) = \tilde{k}(\mathbf{a}_i, \mathbf{a}_j|x)$.

Given training data $\{(x_n, \mathcal{S}_n)\}_{n=1}^N$ where x_n is the input and $\mathcal{S}_n$ is the subset of labels assigned to x_n, we learn the parameters Θ of DDPP, mainly the weight and bias parameters in DNNs, by maximizing the data likelihood

$$\max_\Theta \quad \mathcal{L}(\{(x_n, \mathcal{S}_n)\}_{n=1}^N) = \prod_{n=1}^N p(\mathcal{S}_n|x_n; \Theta) \tag{4}$$

Since the DPP used in our paper has learnable weight parameters (those in the neural networks) that are adjusted during training to best fit the output labels, it is able to capture any type of relations among labels. This is different from the traditional non-learnable DPP [34] where the kernel matrix is computed on fixed feature vectors of data points and a repulsion effect among data points is favored.

3.2. Learning Label Embeddings

In DDPP, evaluating the label-label kernel function $k(\mathbf{a}_i, \mathbf{a}_j)$ and label-input score function $g(\mathbf{a}_i, x)$ both require the labels to have vector representations. Inspired from studies on word embedding [40], we propose a label embedding approach that learns an embedding vector $\mathbf{a}_i$ for each label i, by exploiting the label co-occurrence patterns in the training data. Given a training instance with labels $\mathcal{S}$, we predict the existence of each label i in $\mathcal{S}$ based on other labels $\mathcal{S} - \{i\}$. Let $\mathbf{a}_{\neg i} = \frac{1}{|\mathcal{S}|-1}\sum_{j\in(\mathcal{S}-\{i\})}\mathbf{a}_j$ be the average embedding of labels in $\mathcal{S} - \{i\}$, then the probability

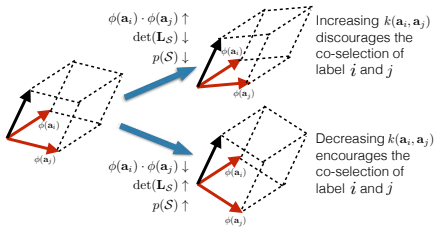

Figure 4. In DPP, the probability of a subset S is proportional to $\det(\mathbf{L}_S)$, which is the volume of the parallelepiped formed by vectors $\{\phi(\mathbf{a}_i), i \in \mathcal{S}\}$. $\phi(\cdot)$ is the reproducing kernel feature map. As we increase the inner product between $\phi(\mathbf{a}_i)$ and $\phi(\mathbf{a}_j)$ (which is essentially $k(\mathbf{a}_i, \mathbf{a}_j)$), the volume of the parallelepiped decreases and $p(S)$ decreases, which discourages the co-selection of label i and j. On the contrary, decreasing $k(\mathbf{a}_i, \mathbf{a}_j)$ encourages the two labels to be co-selected.

that $\mathcal{S}$ contains i is

$$p(i|\mathcal{S}-\{i\}) = \frac{\exp(\mathbf{a}_i^\top \mathbf{a}_{\neg i})}{\sum_{j\in((\mathcal{Y}-\mathcal{S})\cup\{i\})}\exp(\mathbf{a}_j^\top \mathbf{a}_{\neg i})} \quad (5)$$

We learn these embedding vectors by maximizing the data likelihood $\prod_{n=1}^N \prod_{i\in\mathcal{S}_n} p(i|\mathcal{S}_n - \{i\})$. Note that the label embeddings could be learned jointly with the weight parameters of VFN, LIDN and LCN. For simplicity, we performed the learning separately while leaving joint learning to future study.

3.3. Incorporating External Knowledge on Label Co-occurrence

There exists abundant prior knowledge regarding the co-occurrence relations among labels. In particular, we consider binary relations: if two labels have a must-link, they co-occur with high probability; if bearing a cannot-link, they seldom co-occur. Such relations can be obtained from various knowledge base. Other than the WordNet example given in Section 1, one can derive such relations by computing the correlation statistics on a much larger external dataset such as textual tags of Flickr images [53]: two labels are connected with a must-link if their correlation score is high and a cannot link otherwise.

We aim to incorporate these externally-obtained co-occurrence relations to boost MLC performance on Youtube-8M and Open Images. Specifically, we impose relational regularization over DDPP such that labels with must-links are encouraged to be co-selected and those with cannot-links are penalized for co-selection. This regularization approach is designed according to the property of DPP, which assigns larger probability mass $p(\mathcal{S})$ over a label-subset $\mathcal{S}$ where the labels are more mutually "different" (Fig. 4). The "difference" between two labels $\mathbf{a}_i$ and $\mathbf{a}_j$ is measured by the kernel function $k(\mathbf{a}_i, \mathbf{a}_j)$: the smaller $k(\mathbf{a}_i, \mathbf{a}_j)$ is, the more different $\mathbf{a}_i$ and $\mathbf{a}_j$ are. To encourage label i and j to be simultaneously selected into $\mathcal{S}$, we encourage $k(\mathbf{a}_i, \mathbf{a}_j)$ to be small to increase $p(\mathcal{S})$. To discourage simultaneous selection, $k(\mathbf{a}_i, \mathbf{a}_j)$ is preferred to be large to decrease $p(\mathcal{S})$. Denoting $\mathcal{M}$ and $\mathcal{C}$ the set of label pairs possessing must and cannot links respectively, we define the following relation-regularized DDPP (RDDPP) problem

$$\max_\Theta \quad \mathcal{L}(\{(x_n, \mathcal{S}_n)\}_{n=1}^N) + \lambda(-\sum_{(i,j)\in\mathcal{M}} k(\mathbf{a}_i, \mathbf{a}_j) + \sum_{(i,j)\in\mathcal{C}} k(\mathbf{a}_i, \mathbf{a}_j)) \quad (6)$$

In the second term of the objective function, we encourage label pair (i, j) with must-link to have smaller $k(\mathbf{a}_i, \mathbf{a}_j)$ and those with cannot-link to have larger $k(\mathbf{a}_i, \mathbf{a}_j)$.

3.4. Parameter Learning

To solve the problem defined in Eq.(4) and Eq.(6), we use gradient descent method to minimize the negative log-likelihood $\sum_{n=1}^N(-\log\det(\mathbf{L}_{\mathcal{S}_n}(x_n)) + \log\det(\mathbf{L}(x_n) + \mathbf{I}))$. To compute the gradient of $\log\det(\mathbf{L}(x_n) + \mathbf{I})$, one needs to invert the matrix $\mathbf{L}(x_n) + \mathbf{I}$, with a complexity of $O(K^3)$ where K is the number of classes. When K is large, the scalability of the algorithm is very prohibitive. To address this issue, we leverage a low rank kernel learning approach based on inducing points methods [48] and structure exploiting approaches [59]. The inducing points method introduces a set of auxiliary points $U = \{\mathbf{u}_m\}_{m=1}^M$ and approximates the kernel $k(\mathbf{a}_i, \mathbf{a}_j)$ as $k(\mathbf{a}_i, \mathbf{a}_j) \approx \mathbf{V}_{\mathbf{a}_i,U}^\top \mathbf{V}_{U,U}^{-1} \mathbf{V}_{\mathbf{a}_j,U}$ where $\mathbf{V}_{\mathbf{a}_i,U}$ is a M-dimensional vector whose m-th element is $k(\mathbf{a}_i, \mathbf{u}_m)$ and $\mathbf{V}_{U,U}$ is $M \times M$ matrix in which the (m, n)-th entry is $k(\mathbf{u}_m, \mathbf{u}_n)$. Inspired by [59], the vector $\mathbf{V}_{\mathbf{a}_i,U}$ can be further approximated using interpolation: first finding two inducing points $\mathbf{u}_a$ and $\mathbf{u}_b$ that closely bound $\mathbf{a}_i$, then approximating $k(\mathbf{a}_i, \mathbf{u}_m)$ as $w_i k(\mathbf{u}_a, \mathbf{u}_m) + (1 - w_i)k(\mathbf{u}_b, \mathbf{u}_m)$, where w_i is a learnable weight. Let $\mathbf{w}_i \in \mathbb{R}^M$ denote a sparse vector with only two non-zeros entries where the a-th and b-th entry are w_i and $1-w_i$ respectively, then $\mathbf{V}_{\mathbf{a}_i,U}$ can be approximated as $\mathbf{w}_i^\top \mathbf{V}_{U,U}$. To this end, the approximation of $k(\mathbf{a}_i, \mathbf{a}_j)$ is $k(\mathbf{a}_i, \mathbf{a}_j) \approx \mathbf{w}_i^\top \mathbf{V}_{U,U} \mathbf{V}_{U,U}^{-1} \mathbf{V}_{U,U} \mathbf{w}_j = \mathbf{w}_i^\top \mathbf{V}_{U,U} \mathbf{w}_j$. Under this approximated kernel, the kernel matrix $\mathbf{L}(x)$ can be written as $\mathbf{L}(x) = \mathbf{W}\mathbf{V}_{U,U}\mathbf{W}^\top$ where $\mathbf{W}$ is a $K \times M$ sparse matrix of which the i-th row is $g(\mathbf{a}_i, x)\mathbf{w}_i^\top$. According to the Woodbury matrix identity, the inverse of $\mathbf{W}\mathbf{V}_{U,U}\mathbf{W}^\top + \mathbf{I}$ can be computed as $(\mathbf{W}\mathbf{V}_{U,U}\mathbf{W}^\top + \mathbf{I})^{-1} = \mathbf{I} - \mathbf{W}(\mathbf{V}_{U,U}^{-1} + \mathbf{W}^\top\mathbf{W})^{-1}\mathbf{W}^\top$ where the dominating computation is inverting the $M \times M$ matrix $\mathbf{V}_{U,U}^{-1} + \mathbf{W}^\top\mathbf{W}$ with a complexity of $O(M^3)$. M is typically much smaller than K, hence complexity can be reduced greatly. Since $\mathbf{W}$ is very sparse where each row contains only two non-zeros, the matrix multiplication involving $\mathbf{W}$ can be performed very efficiently.

3.5. Inference

Given the learned model parameters, we perform MLC by inferring the mode of the conditional probability $p(\mathcal{S}|x)$.

Given the input data x, we compute the conditional kernel matrix $\mathbf{L}(x)$, then select the optimal subset of labels $\mathcal{S}^*$ as $\mathcal{S}^* = \text{argmax}_{\mathcal{S}}\ p(\mathcal{S}|x) = \text{argmax}_{\mathcal{S}}\ \log\det(\mathbf{L}_{\mathcal{S}}(x))$. This is a NP-hard problem since the search space $\{\mathcal{S}|\mathcal{S} \subseteq \mathcal{Y}\}$ is exponential. To address this challenge, we use an approximate inference algorithm proposed by [19], which (1) first relaxes the original 0/1 integer programming problem into a continuous one; (2) then solves the continuous optimization problem in polynomial time; (3) finally rounds the continuous solution back to the 0/1 binary solution. Please refer to the supplements for details.

4. Experiments

In this section, we present experimental results on the Youtube-8M and Open Images datasets. Due to space limit, some results are deferred to the supplements.

4.1. Datasets

The YouTube-8M [1] dataset contains ∼8 million videos, each annotated with multiple labels from 4,800 classes. The average number of annotations per video is 1.8. These videos are 0.5 million hours long and contain ∼1.9 billion frames in total. The dataset is split into a training set with ∼5.8 million videos, a validation set with ∼1.7 million videos and a hidden test set with ∼0.8 million videos. Two types of features are provided for this dataset: frame-level and video-level. To extract frame-level features, each video is decoded at 1 frame-per-second up to the first 360 seconds. The decoded frames are fed into the Inception v3 network [26] pre-trained on ImageNet [14] where the 2048-dimensional ReLu activation of the last hidden layer (layer name *pool_3_reshape*) is utilized as frame-level features. Feature dimension is reduced to 1024 using PCA (+ whitening) followed by quantization (1 byte per coefficient). To obtain video-level features, the first-, second-order and ordinal statistics of frame-level features are computed and normalized. Please refer to [1] for details.

The Open Images [32] dataset contains ∼9 million images which are annotated with labels spanning 6012 classes. The average number of labels per image is 8.85. The dataset is split into a training set (9,011,219 images) and a validation set (167,057 images).

For both datasets, testing is performed on the validation set, which is untouched during model training. The two datasets have been updated since their first release. We used the first version released in September 2016.

4.2. Experimental Setup

Hyperparameters DDPP performs visual representation learning on Youtube-8M frame-level features and Open Image raw pixels. For Youtube-8M, the visual feature network (VFN) in DDPP is configured as a 2-layer 1024-units LSTM network [66]. On Open Images, the VFN is chosen to be the Inception v3 network [26]. For Youtube-8M video-level features, they are directly fed into the label-input dependency network (LIDN) in DDPP without further representation learning. For both datasets, the LIDN is configured to be a fully-connected network with 2 hidden layers where the number of units in the first and second layer is 1024 and 512 respectively and the activation function is ReLU. It takes the concatenation of label embedding and visual representation as inputs and produces a dependency score. The label-correlation network (LCN) has 2 hidden layers where the number of units in the first and second layer is 200 and 100 respectively and the activation function is ReLU. It takes each label-embedding vector as input and produces a 100-dimensional latent representation. Then a linear kernel is applied to the latent representations of two labels. The dimension of label embeddings is set to 300. The regularization parameter in RDDPP is set to 0.1 for Youtube-8M and 0.01 for Open Images. We use AdaGrad [15] with a learning rate of 0.1 and batch size of 32 to learn model parameters. In LSTM training, the network is unrolled for 60 iterations. In low-rank kernel learning, the number of inducing points is set to 200. The hyperparameters of baseline methods are deferred to the supplements.

Baselines For Youtube-8M experiments, we compare with the following baselines on frame-level features:

- Logistic regression with average pooling (LR-Avg) [1]: train 4800 one-vs-rest LR classifiers for each class based on frame-level features; at test time, prediction scores on individual frames are averaged into a video-level score.
- Deep bag of frames [1] with independent labels (DBoF-IL), and long short-term memory network [24, 66, 1] with independent labels (LSTM-IL): use DBoF and LSTM to encode frame features; the output labels are treated as independent, each associated with a sigmoid unit and a binary cross-entropy loss.
- Structured prediction energy networks (SPEN) [4].

and the following baselines on video-level features:

- LR, Support Vector Machine (SVM), Mixture of Experts (MoE) [1]: learn one-vs-rest LR, SVM and MoE classifiers on video-level features.
- Multi-label learning by exploiting label dependency (LEAD) [68], principle label space transformation (PLST) [51], clique generating machine (CGM) [52].

For Open Images, we compare with:

- LR and SVM: use the Inception v3 network pre-trained on ImageNet to extract image features, then learn one-vs-rest LR or SVM classifiers for each class.

Feature	Methods	MAP	Hit@1	PERR
Frame level	LR-Avg [1]	11.0	50.8	42.2
	DBoF-IL [1]	26.9	62.7	55.1
	LSTM-IL [1]	26.6	64.5	57.3
	SPEN [4]	27.4	65.7	57.9
	DDPP-Sep	29.1	66.3	59.1
	DDPP	**30.3**	**67.9**	**59.8**
	RDDPP	**31.8**	**69.1**	**60.6**
Video level	LR [1]	28.1	60.5	53.0
	SVM [1]	17.0	56.3	47.9
	MoE [1]	29.6	62.3	54.9
	LEAD [68]	30.3	63.5	55.6
	PLST [51]	30.7	62.7	56.2
	CGM [38]	31.5	65.2	55.3
	DDPP	**33.8**	**67.1**	**57.5**
	RDDPP	**34.9**	**68.7**	**58.4**

Table 1. MLC Performance (%) on the Youtube-8M Validation Set

- CNN with independent labels (CNN-IL): replace the softmax output layer in Inception v3 network with 6012-way sigmoid layer.
- Deep convolutional ranking (DCR) [21], CNN-RNN [56], conditional graphical Lasso (CGL) [38], multi-task label cleaning (MTLC) [54].

Evaluation Metrics Following [1, 54], we use mean average precision (MAP), Hit@k and precision at equal recall rate (PERR) to evaluate the MLC performance. The AP for each class c is defined as $AP(c) = 1/M \sum_{n=1}^{N} p(c,n)\mathbb{I}(c,n)$ where $p(c,n)$ is the precision for class c when retrieving n annotations, $\mathbb{I}(c,n)$ is an indicator function that is 1 if the ground truth for class c and the image at rank n is positive. N is the size of the validation set and M is the number of positives. The MAP of C classes is $1/C \sum_{c=1}^{C} AP(c)$. Hit@k is the percentage of test instances that contain at least one of the groundtruth labels in the top k predictions. For the definition of PERR, please refer to [1]. These metrics require DDPP to assign a confidence score to each predicted label. We use the continuous values obtained in step 2 of the inference algorithm (Section 3.5) to represent the confidence.

External Knowledge on Label Co-occurrence Relations We harvest label co-occurrence relations from the YFCC100M dataset [53], which contains ∼99.2 million photos and ∼0.8 million videos from Flickr, each annotated with multiple textual tags. On YFCC100M tags, we compute the pointwise mutual information (PMI) [13] between each pair of classes in Youtube-8M or Open Images and assign must-links to 5000 class-pairs that have the largest PMI and cannot-links to 5000 pairs with the smallest PMI.

4.3. Results

Table 1 shows the MLC performance on the Youtube-8M dataset. As can be seen, our method DDPP outperforms the baselines with a large margin, on both the frame-level features and video-level features. At frame level, LSTM-IL and DDPP both leverage LSTM networks to learn visual representations. Their major difference is: LSTM-IL treats the output labels as independent while DDPP aims at capturing high-order label-correlations. As a result, DDPP achieves much better performance than LSTM-IL. Similar to LSTM-IL, DBof-IL ignores label-correlations, which loses the semantic and contextual clues among labels. SPEN uses deep networks to capture nonlinear label-dependency. While outperforming other baselines, it is inferior to DDPP. One possible reason is: during parameter learning of SPEN, an approximated inference procedure is conducted over each training instance, which may incur large approximation errors. On the contrary, no inference is needed when learning DDPP parameters. LR-Avg reduces video-level MLC into frame-level MLC, which fails to consider the inter-frame relations, hence leading to inferior performance.

Methods	MAP
LR	51.9
SVM	53.5
CNN-IL	62.4
DCR [21]	63.6
CNN-RNN [56]	64.1
CGL [38]	63.5
MTLC [54]	63.3
DDPP-Sep	64.6
DDPP	**65.1**
RDDPP	**67.2**

Table 2. MLC Performance on the Open Images Validation Set

The importance of capturing high-order label-correlations is reflected on the video-level features as well. LR, SVM and MoE ignore label-correlations, hence performing less well. LEAD, PLST and CGM aim at exploiting high-order label dependency based on Bayesian network structure learning, label space dimension reduction and clique generation. The experiments show that they are less effective than DDPP. The possible reasons are: (1) LEAD performs structure learning and parameter learning separately while DDPP learns label-correlations and label-input dependency in a joint manner; (2) PLST projects the labels into a linear subspace, hence can only capture linear dependency while DDPP uses kernel methods to exploit nonlinear dependency; (3) Similar to SPEN, CGM's parameter learning involves approximate inference on training data, which may incur considerable approximation errors, while DDPP can avoid this. Besides, these methods lack the flexibility to perform end-to-end feature learning and hence cannot be applied to frame-level features.

In DDPP, capturing label-correlations and learning visual features are performed jointly. To evaluate whether joint learning is better than performing the two tasks separately, we compare DDPP with a model variant which first learns visual features without considering label-correlation, then feeds the learned features to DPP to perform correlation-aware MLC. The first step is performed on the frame-level features using LSTM-IL. We refer to this method as *DDPP Separation* (DDPP-Sep). As shown in

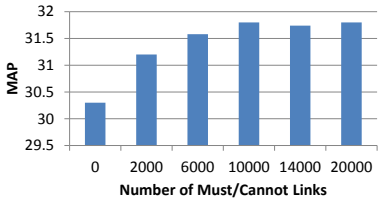

Figure 5. MAP versus the total number of must/cannot links, on the Youtube-8M frame-level features.

Table 1, DDPP-Sep performs worse than DDPP, which corroborates the merit of joint learning.

By incorporating label co-occurrence knowledge, RDDPP improves DDPP greatly. To further study the effect of knowledge-incorporation, we measure how the MAP varies as we gradually add must/cannot links. Class-pairs are ranked according to their PMI in descending order. Must and cannot links are added by visiting the ranked list from top to bottom and from bottom to top respectively. The number of must-links is equal to that of cannot links. Fig. 5 shows the results on Youtube-8M frame-level features. MAP consistently increases when the link number is increased from 2,000 to 10,000, which demonstrates the benefits of incorporating prior knowledge and the efficacy of relational regularization in RDDPP to realize that. Further increasing the link number does not improve MAP, possibly because the links added later contain more noise.

Table 2 shows the MAP on Open Images, where DDPP and RDDPP outperform baselines. LR, SVM, CNN-IL and MTLC learn independent classifiers for each class and ignore label-correlations. DCR uses a ranking-loss to differentiate relevant and irrelevant labels, but is less capable of capturing fine-grained correlations among relevant labels. CNN-RNN characterizes label-dependency using class chaining, where a proper class-order is difficult to specify. CGL captures pairwise correlations using conditional graphical Lasso, but is unable to exploit high-order label-dependency. DDPP performs better than DDPP-Sep by jointly capturing label-correlation and learning visual features. RDDPP outperforms DDPP by incorporating label-correlation knowledge. More results on Open Images are in the supplements.

4.4. Computational Time

The experiments were conducted on two clusters: a 40-machines GPU (Titan X) cluster which ran deep learning (DL) experiments and a 34-machines CPU cluster for non-DL experiments. We compare the training time of DDPP with DL methods. To verify the efficiency-gain brought by low rank kernel learning (Section 3.4), we also compare with the case where the kernel matrix in DDPP is of full rank (DDPP-FullRank). Each model is trained using a distributed system that adopts a data-parallel strategy. Table 3 shows the convergence time of different models on the GPU/CPU cluster, from which we observe: (1) DDPP, which by default adopts low-rank kernel learning, is much more efficient than DDPP-FullRank; the computational complexity of DDPP and DDPP-FullRank is $O(M^3)$ and $O(K^3)$ respectively, where the number of inducing points M is much smaller than the number of classes K. (2) RDDPP, though adding an additional regularizer to DDPP, does not incur substantial extra cost. (3) While being able to capture high-order label-correlations, DDPP's convergence time is comparable with that of other deep learning methods which assume label-independence, such as DBoF-IL, LSTM-IL and CNN-IL. (4) DDPP is much more efficient than SPEN which involves a costly inference procedure in parameter learning; no inference is needed in learning DDPP parameters.

Methods	Youtube-8M	Open Images
DBoF-IL [1]	6.3	–
LSTM-IL [1]	6.7	–
SPEN [4]	23.6	–
CNN-IL	–	9.4
DCR [21]	–	9.5
CNN-RNN [56]	–	9.8
MTLC [56]	–	10.6
DDPP-FullRank	19.3	38.2
DDPP	7.4	10.4
RDDPP	7.8	11.3

Table 3. Convergence Time (Hours)

5. Conclusions and Future Works

We study the large-scale multi-label classification on two recently released datasets: Youtube-8M and Open Images. To capture the high-order correlation among labels while retaining computational efficiency, we propose Deep Determinantal Point Process (DDPP) that seamlessly integrates DPP and deep neural networks (DNNs) and supports end-to-end learning. DPP is able to capture label-correlation of arbitrary order within polynomial computational time while DNNs play the role of representation learning of images and videos. To incorporate prior knowledge regarding label co-occurrence relations, we impose relational regularization over DDPP's kernel matrix. A low-rank kernel learning algorithm is investigated to scale DDPP to millions of instances and thousands of labels. Experiments on the two datasets demonstrate the efficacy and efficiency of our methods. For future works, we plan to investigate the noisy and missing label problem [54] presenting in Open Images and leverage label hierarchy to improve MLC performance.

Acknowledgements

P.X and E.X are supported by National Institutes of Health P30DA035778, R01GM114311, National Science Foundation IIS1617583, DARPA FA872105C0003. L.M is supported by National Natural Science Foundation of China No.61672068.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 1, 6, 7, 8

[2] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*, 2013. 3

[3] K. Balasubramanian and G. Lebanon. The landmark selection method for multiple output prediction. *ICML*, 2012. 3

[4] D. Belanger and A. McCallum. Structured prediction energy networks. In *ICML*, 2016. 2, 3, 6, 7, 8

[5] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, 2010. 3

[6] W. Bi and J. T.-Y. Kwok. Efficient multi-label classification with many labels. In *ICML*, 2013. 3

[7] S. S. Bucak, R. Jin, and A. K. Jain. Multi-label learning with incomplete class assignments. In *CVPR*, 2011. 3

[8] S. S. Bucak, P. K. Mallapragada, R. Jin, and A. K. Jain. Efficient multi-label ranking for multi-class learning: application to object recognition. In *ICCV*, 2009. 3

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2014. 2, 3

[10] Y. Chen and H. Lin. Feature-aware label space dimension reduction for multi-label classification. In *NIPS*, 2012. 3

[11] W. Cheng, E. Hüllermeier, and K. J. Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML*, 2010. 3

[12] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, 2009. 1

[13] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990. 7

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[15] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011. 6

[16] X. Geng and L. Luo. Multilabel ranking with inconsistent rankers. In *CVPR*, 2014. 3

[17] M. George and C. Floerkemeier. Recognizing products: A per-exemplar multi-label image classification approach. In *ECCV*, 2014. 3

[18] N. Ghamrawi and A. McCallum. Collective multi-label classification. In *CIKM*, 2005. 2, 3

[19] J. Gillenwater, A. Kulesza, and B. Taskar. Near-optimal map inference in determinantal point process. In *NIPS*, 2012. 6

[20] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*, 2014. 3

[21] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013. 7, 8

[22] Y. Guo and S. Gu. Multi-label classification using conditional dependency networks. In *IJCAI*, 2011. 2, 3

[23] B. Hariharan, L. Zelnik-Manor, M. Varma, and S. Vishwanathan. Large scale max-margin multi-label classification with priors. In *ICML*, 2010. 2, 3

[24] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 6

[25] D. J. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, 2009. 3

[26] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6

[27] S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *KDD*, 2008. 2, 3

[28] L. Jing, L. Yang, J. Yu, and M. K. Ng. Semi-supervised low-rank mapping learning for multi-label classification. In *CVPR*, 2015. 3

[29] A. Kanehira and T. Harada. Multi-label ranking from positive and unlabeled data. In *CVPR*, 2016. 3

[30] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *CVPR*, 2006. 3

[31] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011. 2, 3

[32] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, S. Belongie, D. Cai, Z. Feng, V. Ferrari, V. Gomes, A. Gupta, D. Narayanan, C. Sun, G. Chechik, and K. Murphy. Open-images: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2016. 1, 6

[33] A. Kulesza and B. Taskar. Learning determinantal point processes. *UAI*, 2012. 2, 3, 4

[34] A. Kulesza, B. Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 2012. 2, 4

[35] J. Lafferty, A. McCallum, F. Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001. 3

[36] D. Lee, G. Cha, M.-H. Yang, and S. Oh. Individualness and determinantal point processes for pedestrian detection. In *ECCV*, 2016. 3

[37] C. Li, B. Wang, V. Pavlu, and J. Aslam. Conditional bernoulli mixtures for multi-label classification. In *ICML*, 2016. 3

[38] Q. Li, M. Qiao, W. Bian, and D. Tao. Conditional graphical lasso for multi-label image classification. In *CVPR*, 2016. 2, 3, 7

[39] X. Li, F. Zhao, and Y. Guo. Conditional restricted boltzmann machines for multi-label learning with incomplete labels. In *AISTATS*, 2015. 3

[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 4

[41] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 2

[42] G. Patterson, C. Xu, H. Su, and J. Hays. The sun attribute database: Beyond categories for deeper scene understanding. *IJCV*, 108(1-2):59–81, 2014. 1

[43] Y. Prabhu and M. Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, 2014. 3

[44] V. Ranjan, N. Rasiwasia, and C. Jawahar. Multi-label cross-modal retrieval. In *ICCV*, 2015. 3

[45] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine learning*, 2011. 3

[46] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *JMLR*, 2006. 3

[47] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002. 4

[48] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–52, 1985. 2, 5

[49] L. Sun, S. Ji, and J. Ye. Hypergraph spectral learning for multi-label classification. In *KDD*, 2008. 2, 3

[50] G. Sundaramoorthi and B.-W. Hong. Fast label: Easy and efficient solution of joint multi-label and estimation problems. In *CVPR*, 2014. 3

[51] F. Tai and H. Lin. Multilabel classification with principal label space transform. *Neural Computation*, 2012. 3, 6, 7

[52] M. Tan, Q. Shi, A. van den Hengel, C. Shen, J. Gao, F. Hu, and Z. Zhang. Learning graph structure for multi-label image classification via clique generation. In *CVPR*, 2015. 2, 3, 6

[53] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 5, 7

[54] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie. Learning from noisy large-scale datasets with minimal supervision. *CVPR*, 2017. 7, 8

[55] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. Decision trees for hierarchical multi-label classification. *Machine Learning*, 2008. 3

[56] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu. Cnn-rnn: A unified framework for multi-label image classification. In *CVPR*, 2016. 2, 3, 7, 8

[57] Z. Wang, B. Du, L. Zhang, L. Zhang, M. Fang, and D. Tao. Multi-label active learning based on maximum correntropy criterion: Towards robust and discriminative labeling. In *ECCV*, 2016. 3

[58] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. Cnn: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*, 2014. 3

[59] A. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *ICML*, 2015. 5

[60] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. *AISTATS*, 2015. 3

[61] B. Wu, S. Lyu, and B. Ghanem. Ml-mg: multi-label learning with missing labels using a mixed graph. In *ICCV*, 2015. 3

[62] X. Xue, W. Zhang, J. Zhang, B. Wu, J. Fan, and Y. Lu. Correlative multi-label multi-instance image annotation. In *ICCV*, 2011. 3

[63] H. Yang, J. Tianyi Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai. Exploit bounding box annotations for multi-label object recognition. In *CVPR*, 2016. 3

[64] H. Yang, J. T. Zhou, and J. Cai. Improving multi-label learning with missing labels by structured semantic correlations. In *ECCV*, 2016. 3

[65] C.-K. Yeh, W.-C. Wu, W.-J. Ko, and Y.-C. F. Wang. Learning deep latent spaces for multi-label classification. 2017. 3

[66] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 6

[67] K. Zhang, W. Chao, F. Sha, and K. Grauman. Video summarization with lstm. In *ECCV*, 2016. 3

[68] M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *KDD*, 2010. 2, 3, 6, 7

[69] F. Zhao, Y. Huang, L. Wang, and T. Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*, 2015. 3

[70] K. Zhao, W.-S. Chu, and H. Zhang. Deep region and multi-label learning for facial action unit detection. In *CVPR*, 2016. 3