A Survey on Multi-output Learning

Donna Xu, Yaxin Shi, Ivor W. Tsang, Yew-Soon Ong, Chen Gong, and Xiaobo Shen

***Abstract*—Multi-output learning aims to simultaneously predict multiple outputs given an input. It is an important learning problem due to the pressing need for sophisticated decision making in real-world applications. Inspired by big data, the 4Vs characteristics of multi-output imposes a set of challenges to multi-output learning, in terms of the *volume*, *velocity*, *variety* and *veracity* of the outputs. Increasing number of works in the literature have been devoted to the study of multi-output learning and the development of novel approaches for addressing the challenges encountered. However, it lacks a comprehensive overview on different types of challenges of multi-output learning brought by the characteristics of the multiple outputs and the techniques proposed to overcome the challenges. This paper thus attempts to fill in this gap to provide a comprehensive review on this area. We first introduce different stages of the life cycle of the output labels. Then we present the paradigm on multi-output learning, including its myriads of output structures, definitions of its different sub-problems, model evaluation metrics and popular data repositories used in the study. Subsequently, we review a number of state-of-the-art multi-output learning methods, which are categorized based on the challenges.**

***Index Terms*—Multi-output learning, structured output prediction, output label representation, crowdsourcing, label distribution, extreme classification.**

I. Introduction

TRADITIONAL supervised learning, as one of the mostly adopted machine learning paradigms in real-world smart machines and applications, helps in making fast and accurate decisions for regression and prediction tasks. The goal of traditional supervised learning is to learn a function that maps from the input instance space to the output space, where the output is either a single label (for prediction tasks) or a single value (for regression tasks). Therefore, it usually can be used to solve questions with simple answers, such as true-false or single-value answers. For example, a traditional binary classification model can be adopted to detect whether an incoming email is a spam. A traditional regression model can be used to predict daily energy consumption based on temperature, wind speed, humidity and etc.

However, with the demand of researchers and industrial players on the increasing complexity of tasks and problems, there is a pressing need for machine learning models to engage in sophisticated decision making. Many questions require complex answers, which may consist of multiple decisions. For example, in computer vision, we often need to annotate an image by multiple labels that are appear in the image, or even require a ranked list of annotations [1]. In natural language processing, we translate sentences from a specific language to another, where each sentence is a sequence of words [2]. In bioinformatics, we analyze protein sequence (primary structure) to predict the protein structures including the secondary structure elements, element arrangement and chain association [3].

Such sophisticated decision making and complex prediction problems can be typically handled by multi-output learning, which is an emerging machine learning paradigm that aims to simultaneously predict multiple outputs given an input. Multi-output learning is an important learning paradigm that subsumes many learning problems in multiple disciplines and deals complex desicion making in many real-world applications. Compared with the traditional single-output learning, it has the multivariate nature and the multiple outputs may have complex interactions which can only be handled by structured inference. The output values have diverse data types in various machine learning problems. For example, binary output values can refer to multi-label classification problem [4]; nominal output values to multi-dimensional classification problem [5]; ordinal output values to label ranking problem [6]; and real-valued outputs to multi-target regression problem [7]. The aforementioned problems and many other machine learning problems can be subsumed under the multi-output paradigm, and they have been investigated and explored by many researchers from different disciplines in the past.

A. The 4Vs Challenges of Multiple Outputs

In recent times, the characteristics of big data have been widely established and are defined as the popular 4Vs, *volume*, *velocity*, *variety* and *veracity* (usually referring to the input data). Inspired by big data, the 4Vs can be applied to describe the output labels in multi-output learning. Furthermore, the 4Vs characteristics of multi-output brings a set of new challenges to the learning process of multi-output learning. The characteristics of multi-output and the challenges imposed by each of them are introduced in the following.

1) Volume refers to the explosive growth of output labels that have been generated. Such massive amounts of labels poses many challenges to multi-output learning. First, the output label space can be extremely high, which would cause scalability issues. Second, it increases significant burden to the label annotators and results in insufficient annotations of datasets. Thus it may lead to unseen outputs during testing. Third, not all the generated labels have sufficient data instances (inputs) when constructing a dataset, and it may pose the issue of label imbalance.

D. Xu, Y. Shi and I. W. Tsang are with the Centre for Artificial Intelligence, FEIT, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: doxu2620@gmail.com, yaxin.shi@student.uts.edu.au, ivor.tsang@uts.edu.au).

Y.-S. Ong is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: asysong@ntu.edu.sg).

C. Gong and X. Shen are with the School of Computer and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chen.gong@njust.edu.cn, njust.shenxiaobo@gmail.com).

2) Velocity refers to speed of output label acquisition including the phenomenon of concept drift and update to the model. The challenge imposed by velocity could possible be the change of output distributions, where the target outputs are changing over time in unforeseen ways.
3) Variety refers to heterogeneous nature of output labels. Output labels are gathered from multiple sources and are of various data formats with different structures. Particularly, complex structures of the output labels would lead to multiple challenges to multi-output learning, such as appropriate modeling of output dependencies, multivariate loss function design and efficient algorithms to solve the problems with complex structures. Furthermore, heterogeneous outputs poses challenges to multi-outptu learning as well.
4) Veracity refers to the diverse quality of the output labels. Issues such as noise, missing values, abnormalities or incompletion of the data are considered under this category. It poses a challenge of noisy labels including missing labels and incorrect labels.

In recent years, there have been many researchers devoting themselves in dealing with the challenges brought by the 4Vs of multi-output. The goal of this paper is to provide a comprehensive overview of multi-output learning, in particular, on different output structures and underlying issues. Multi-output learning has attracted significant attentions from many machine learning related communities, such as part-of-speech sequence tagging and language translation in natural language processing, motion tracking and optical character recognition in computer vision, document categorization ranking in information retrieval and so on. We expect to deliver a whole picture of multi-output learning with subsumption of different problems across multiple communities and promote the development of multi-output learning.

B. *Organization of This Survey*

Fig. 1 illustrates the organization of the survey. Section 2 focuses on the *life cycle of the output label*. Section 3 presents the paradigm of multi-output learning by providing an overview about *myriads of output structures* and the *problem definitions* on multi-output learning and its sub-problems, *model evaluation metrics* and publicly available *datasets*. Section 4 presents the *challenges of multi-output learning* and their corresponding *representative works*.

II. Life Cycle of Output Labels

The output label plays an important role in multi-output tasks. The performance of a multi-output task relies heavily on the quality of the labels. Fig. 2 depicts various stages of *label life cycle*, which consists label annotation, label representation and label evaluation. We provide a brief overview for each stage and present several underlying issues that could potentially harm the effectiveness of the multi-output learning systems.

A. *How is Data Labeled*

Label annotation is a crucial step in training multi-output learning models. It requires human to semantically annotate the data. Depending on the multi-output tasks and applications, label annotations have various types. For example, given an image dataset, a classification task requires the images to be labeled with tags or keywords; a segmentation task requires the objects in the images to be localized and indicated by a mask; and a captioning task requires the images to be labeled with some textual descriptions.

Depending on the annotation requirement and the application area, creating large annotated dataset from scratch requires a lot of effort and it is a time-consuming activity. Nowadays, there are multiple ways to get labeled data. Social media provides a platform for researchers to search for labeled datasets. For example, social networks such as Facebook and Flickr allow users to post pictures and texts with tags, which can facilitate the classification tasks. Domain Group is a real-estate portal that provides property price report which helps in collecting labeled data for house price forecasting problem. DBLP, a computer science bibliography website that hosts a database, contains a rich source of information about publications, authorships, author networks and etc, which can be used for investigating various problems such as network analysis, classification, ranking and so on. Open-source collections such as WordNet and Wikipedia are useful sources for getting labeled datasets as well.

Apart from directly obtaining labeled datasets, crowdsourcing platforms (e.g. Amazon Mechanic Turk) help the researchers to solicit the labels for unlabeled datasets. They recruit online workers to acquire annotations given a dataset. The annotation type depends on the modeling task. Due to the efficiency of crowdsourcing, it soon become an attraction method to obtain labeled datasets. ImageNet [8] is a popular image dataset where the labels for the images are obtained through the crowdsourcing platform. It is then used in the subsequent research to help solving problems in various areas. It uses WordNet hierarchy to organize its database of images.

Furthermore, there are many annotation tools developed to annotate different types of data. LabelMe [9], a web-based tool, provides the user a convenient way for image annotation, where the user can label as many as objects in the image, as well as correct the labels annotated by other users. BRAT [10] is also a web-based annotation tool for natural language processing tasks such as named-entity recognition, POS-tagging and etc. TURKSENT [11] provides annotation specifically for sentiment analysis on social media posts.

After data getting labeled, a label set can be aggregated for further analysis.

B. *Forms of Label Representations*

There are many different types of label annotations (such as tags, captions, masks of image segmentation and etc.) for different tasks, and for each type of the annotation, there could be many representations (which are usually represented in vectors). For example, to represent the annotation of tags, there are multiple ways. The most commonly used one is the binary

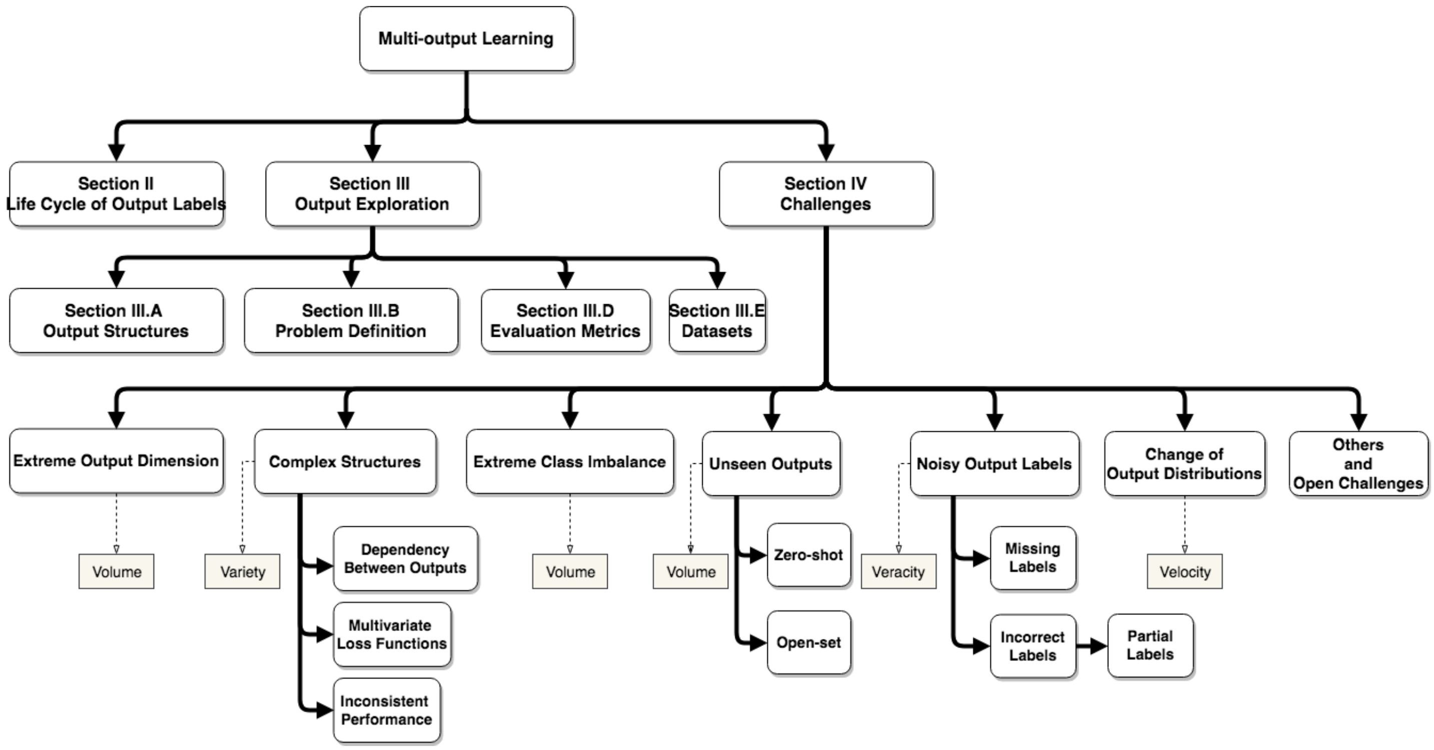

Fig. 1. The basic organization of this survey.

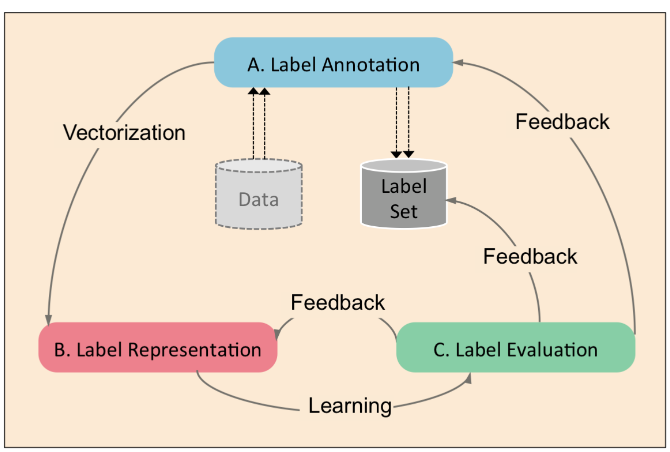

Fig. 2. The life cycle of the output label.

vector, where the size of the vector is the vocabulary size of tags, and the value 1 is assigned to the annotated tags and the value 0 to the rest of the tags. Depending on the multi-output tasks, such representation may not be well enough for the output label space as it loses some useful information such as semantics and internal structure. To tackle the aforementioned issue, many other representation methods are widely adopted. Real-valued vectors of tags [12] indicate the strength and degree of the annotated tags by real values. Binary vectors of tag-attribute association can be used to convey the characteristics for tags. Hierarchical label embedding vectors [13] can be used to capture the tag structure information. Semantic word vectors such as Word2Vec [14] incorporates the semantics of the textual descriptions of tags. Therefore, for each multi-output task, the corresponding annotated labels have various representations, then it is critical to select the one that is the most appropriate for the given task.

C. *Label Evaluation and Challenges*

Label evaluation is an essential step to guarantee the quality of labels and label representations, and thus plays a key role in the performance of multi-output tasks. Different multi-output learning models can be used to evaluate the label quality in terms of different tasks. Labels can be evaluated from three different perspectives. 1) To evaluate whether the annotation has good quality (Step A). 2) To evaluate whether the chosen label representation can well represent the labels (Step B). 3) To evaluate whether the provided label set well covers the dataset (Label Set). After the evaluation, it usually requires human expert to explore and address the underlying issues, and provide feedbacks to improve different aspects of labels accordingly.

1) Issues of Label Annotation: Various annotation ways such as crowdsourcing, annotation tools or social media help the researchers to collected annotated data with efficiency. Without the involvement of experts, these annotation methods might bring a critical issue to the table, noisy labels, including missing annotations and incorrect annotations. Noisy labels can be caused by various reasons. For example, online workers from crowdsourcing platforms or users of annotation tools lacking of domain knowledge would lead to noisy labels. Tagging activities of users from social media such as Facebook and Instagram might result in irrelevant tags to the given image or post. Some ambiguous text may also affect the label annotators on their understanding of the provided data.

2) Issues of Label Representation: Labels have internal structures, which should be considered in the label representation according to the specified multi-output task. However, it is non-trivial to incorporate such information in the representation as labels usually come with large size and require domain knowledge to define the structure.

In addition, the output label space might contain ambiguity. For example, in many of the natural language processing tasks, a traditional representation for the label space is Bag-of-Words (BOW) approach. BOW approach results in word sense ambiguity as two different words may refer to the same meaning, and one word might refer to multiple meanings.

3) Issues of the Label Set: Acquiring the label set for data annotation requires human expert with domain knowledge. It is common that the provided label set does not contain sufficient labels to the data due to several possible reasons such as fast data growth and low appearance of some labels. Therefore, there could be unseen labels associated with the test data, which leads to open-set [15], zero-shot [16] or concept drift [17] problems.

III. Multi-output Learning

In contrast to traditional single-output learning, multi-output learning simultaneously predicts multiple outputs where the outputs have various types of structures in many different sub-fields with applications in a diverse range of disciplines. A summary of sub-fields of multi-output learning along with their output structures, applications and corresponding discipline is presented in Table I.

In this section, we first introduce several output structures in the multi-output learning problem. Then we provide its problem definition along with different constraints on the output space for a number of sub-fields of multi-output learning problem. Some special cases of the sub-fields are given. Subsequently, we briefly give an overview on some evaluation metrics that are specific to multi-output learning and present some new insights into the evolution of output dimensions by analyzing several commonly used datasets.

A. Myriads of Output Structures

The increasing demand of complex modeling tasks with sophisticated decisions has led to new creations of outputs, some of which are with complex structures. With the ubiquitous social media, social networks and various online services, a wide range of output labels can be stored and then collected by researchers. Output labels can be anything. For example, they could be any multimedia content such as text, image, audio or video. Given long document as input, the output can be the summarization of the input, which is in the text format. Given some text fragments, the output can be an image with the contents described by the input text. Similarly, audio such as music and videos can be generated given different types of inputs. Apart from the multimedia content, there are a number of different output structures. Here we present several typical output structures using the example in Fig. 3 to illustrate myriads of output structured given an image as an input.

1) Independent Vector: Independent vector is the vector with independent dimensions, where each dimension represents a particular label which does not necessarily depend on other labels. Binary vectors can be used to represent tags, attributes, bag-of-words, bad-of-visual-words, hash codes and etc. of a given data. Real-valued vectors provides the weighted dimensions, where the real value represents the strength of the input data to the corresponding label. Applications include annotation or classification on text, image or video for binary vectors [18]–[20] , and demand or energy prediction for real-valued vectors [22]. Independent vector can be used to represent tags of an image, as shown in Fig. 3 (1), where all the tags, *people*, *dinner*, *table* and *wine* have equal weights.

2) Distribution: Different from independent vector, distribution provides the information of probability distribution for each dimension. The tag with the largest weight, *people*, is the main content of the image. *dinner* and *table* have similar distributions. Applications include head pose estimation [24], facial age estimation [25] and text mining [26].

3) Ranking: The output can be a ranking, which shows the tags ordered from the most important to the least. The results from a distribution learning model can be converted to the ranking, but a ranking model is not restrictive to the distribution learning model only. Examples of its application are text categorization ranking [27], question answering [28] and visual object recognition [1].

4) Text: Text can be in the form of keywords, sentences, paragraphs or even documents. Fig. 3 (4) illustrate an example of text output as captioning *People are having dinner* to the image. Other applications for text outputs can be document summarization [41] and paragraph generation [42].

5) Sequence: Sequence is usually a sequence of elements selected from a label set or word set. Each element is predicted depending on the input as well as the predicted outputs before its position. An output sequence often corresponds with an input sequence. For example, in speech recognition, given a sequence of speech signal, we expect to output a sequence of text corresponds to the input audio [43]. In language translation, given a sentence in one language, output the corresponding translated sentence in target language [2]. In the example shown in Fig. 3 (5), given the input of the image caption, the output is the POS tags for each word in a sequence.

6) Tree: Tree as the output is in the form of a hierarchical labels. The outputs have the hierarchical internal structure where each output belong to a label as well as its ancestors in the tree. For example, in syntactic parsing [31] shown in Fig. 3 (6), given a sentence, each output is a POS tag and the entire output is a parsing tree. *People* belongs to the noun *N* as well as the noun phrase *NP* in the tree.

7) Image: Image is a special form of output that consists of multiple pixel values and each pixel is predicted depending on the input and the pixels around it. Fig. 3 (7) shows one popular application for image output, super-resolution construction [33], which constructs high-resolution image given a low-resolution one. Other image output applications include text-to-image synthesis [44] which generates images from natural language descriptions, and face generation [45].

TABLE I
A SUMMARY OF SUB-FIELDS OF MULTI-OUTPUT LEARNING AND THEIR CORRESPONDING OUTPUT STRUCTURES, APPLICATIONS AND DISCIPLINES.

Sub-field	Output Structure	Application	Discipline
Multi-label Learning	Independent Binary Vector	Document Categorization [18]	Natural Language Processing
		Semantic Scene Classification [19]	Computer Vision
		Automatic Video Annotation [20]	Computer Vision
Multi-target Regression	Independent Real-valued Vector	River Quality Prediction [21]	Ecology
		Natural Gas Demand Forecasting [22]	Energy Meteorology
		Drug Efficacy Prediction [23]	Medicine
Label Distribution Learning	Distribution	Head Pose Estimation [24]	Computer Vision
		Facial Age Estimation [25]	Computer Vision
		Text Mining [26]	Data Mining
Label Ranking	Ranking	Text Categorization Ranking [27]	Information Retrieval
		Question Answering [28]	Information Retrieval
		Visual Object Recognition [1]	Computer Vision
Sequence Alignment Learning	Sequence	Protein Function Prediction [3]	Bioinformatics
		Language Translation [2]	Natural Language Processing
		Named Entity Recognition [29]	Natural Language Processing
Network Analysis	Graph	Scene Graph [30]	Computer Vision
	Tree	Natural Language Parsing [31]	Natural Language Processing
	Link	Link Prediction [32]	Data Mining
Data Generation	Image	Super-resolution Image Reconstruction [33]	Computer Vision
	Text	Language Generation	Natural Language Processing
	Audio	Music Generation [34]	Signal Processing
Semantic Retrieval	Independent Real-valued Vector	Content-based Image Retrieval [35]	Computer Vision
		Microblog Retrieval [36]	Data Mining
		News Retrieval [37]	Data Mining
Time-series Prediction	Time Series	DNA Microarray Data Analysis [38]	Bioinformatics
		Energy Consumption Forecasting [39]	Energy Meteorology
		Video Surveillance [40]	Computer Vision

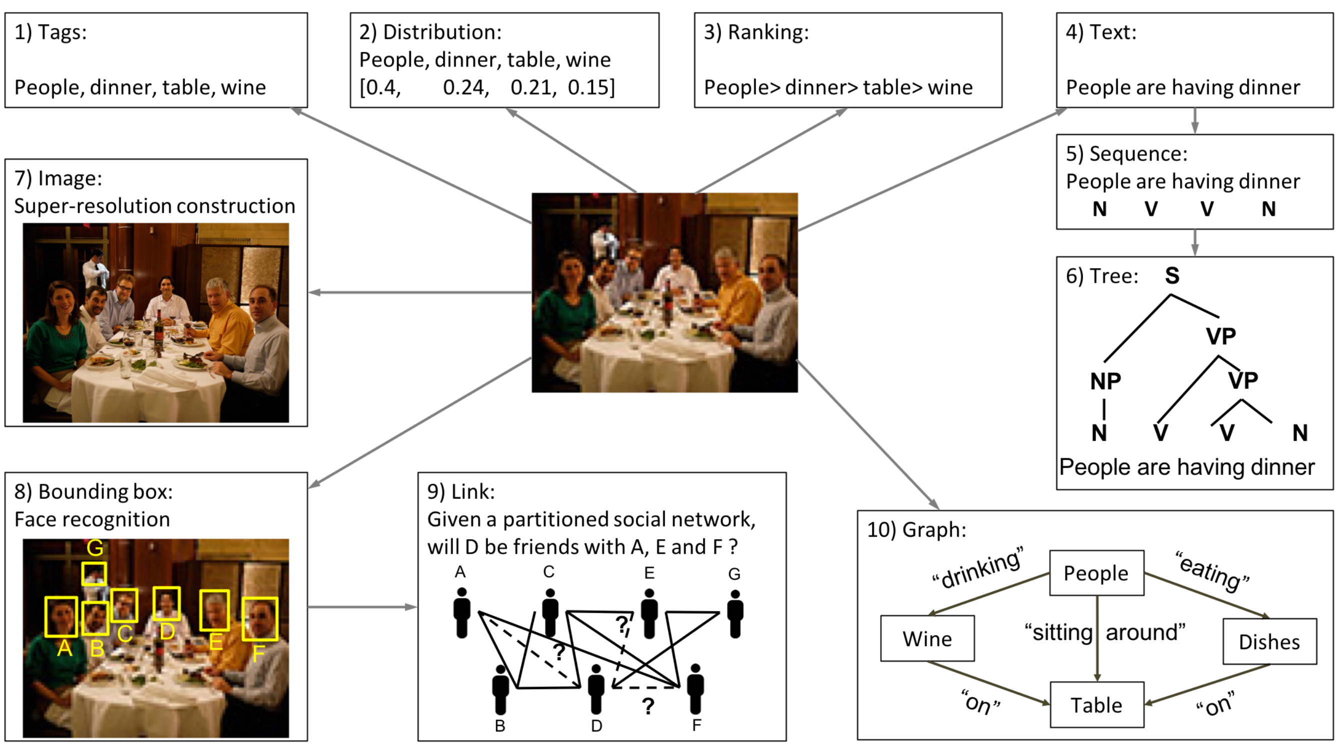

Fig. 3. An illustration on the myriads of output structures for an input image in the social network as an example.

8) Bounding Box: Bounding box is often used to find the exact locations of the objects appeared in an image and it is commonly used in object recognition and object detection [1]. In Fig. 3 (8), each of the faces is localized by a bounding box and each of them can then be identified.

9) Link: Link as the output usually represents the association between two nodes in a network [32]. As shown in Fig. 3 (9), given a partitioned social network with edges representing friendship of the users, we expect to predict whether two currently unlinked users will be friends in the future.

10) Graph: A graph is made up of a set of nodes and edges and it is used to model the relations between objects, where each object is represented by a node and the appearance of an edge represents a relationship between two connected objects. Scene graph [46] is one type of graph that can be used as output to describe image content [30]. Fig. 3 (10) shows that given an input image we expect to output a graph definition where nodes are the objects appeared in the image, *People*, *Wine*, *Dishes* and *Table*, and edges are the relations between objects. Scene graph is a very useful representation for tasks such as image generation [47] and visual question answering [48].

11) Others: There are several other output structures. For example, *contour* and *polygon* that are similar to bounding box which can be used to localize objects in an image. In information retrieval, the output can be a *list* of data objects that are similar to the given query. In image segmentation, the output is usually *segmentation masks* for different objects. In signal processing, the output can be *audio* including *speech* and *music*. In addition, some real-world applications may require more sophisticated output structures by multiple related tasks. For example, one may require to recognize the objects given multiple images and localize them at the same time such as discovering common saliency on the multiple images in co-saliency [49], simultaneously segmenting similar objects given multiple images in co-segmentation [50] and detecting and identifying objects in multiple images in object co-detection [51].

B. Problem Definition of Multi-output Learning

Multi-output learning maps each input (instance) to multiple outputs. Assume $\mathcal{X} = \mathbb{R}^d$ is the d-dimensional input space, and $\mathcal{Y} = \mathbb{R}^m$ is the m-dimensional output label space. The task of multi-output learning is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ from the training $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | 1 \leq i \leq n\}$. For each training example $(\mathbf{x}_i, \mathbf{y}_i)$, $\mathbf{x}_i \in \mathcal{X}$ is a d-dimensional feature vector, and $\mathbf{y}_i \in \mathcal{Y}$ is the corresponding output associated with $\mathbf{x}_i$. Different sub-fields of multi-output learning have a unified framework to solve the problem, which is to find a function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $F(\mathbf{x}, \mathbf{y})$ is a compatibility function that evaluates how compatible the input $\mathbf{x}$ and the output $\mathbf{y}$ is. Given an unseen instance $\mathbf{x}$, the output is predicted to be the one with the largest compatibility score $f(\mathbf{x}) = \widetilde{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y})$. Depending on the different sub-fields of multi-output learning, the output label space $\mathcal{Y}$ has various constraints. We select several popular sub-fields and present the constraints of their output space in the following sections. Note that multi-output learning is not restrictive to these settings. We simply list a number of examples for illustration.

1) Multi-label Learning: In multi-label learning [4], each instance associates with multiple labels, represented by a sparse binary label vector, with value of $+1$ for label appearance and -1 for label absence. Thus, $\mathbf{y}_i \in \mathcal{Y} = \{-1, +1\}^m$. For an unseen instance $\mathbf{x} \in \mathcal{X}$, the learned multi-label classification function $f(.)$ outputs $f(\mathbf{x}) \in \mathcal{Y}$, where the labels with value 1 in the output vector as the predicted labels for $\mathbf{x}$.

2) Multi-target Regression: In multi-target regression [7], [52], each instance associates with multiple labels, represented by a real-valued vector, with values represent the strength or degree of the instance to the label. Therefore, we have the constraint of $\mathbf{y}_i \in \mathcal{Y} = \mathbb{R}^m$. For an unseen instance $\mathbf{x} \in \mathcal{X}$, the learned multi-target regression function $f(.)$ predicts a real-valued vector $f(\mathbf{x}) \in \mathcal{Y}$ as the output.

3) Label Distribution Learning: Different from multi-label learning which learns to predict a set of labels, label distribution learning focuses on predicting multiple labels with the degree value to which each label describes the instance. In addition, each degree value is the probability of how much likely the label associates with the instance. Therefore, the sum of the degree values for each instance is 1. Thus, the output space for label distribution learning satisfies $\mathbf{y}_i = (y_i^1, y_i^2, ..., y_i^m) \in \mathcal{Y} = \mathbb{R}^m$ with the constraints $y_i^j \in [0, 1], 1 \leq j \leq m$ and $\sum_{j=1}^m y_i^j = 1$.

4) Label Ranking: In label ranking, each instance is associated with rankings of over multiple labels. Therefore, the output of each instance is a total order of all the labels. Let $\mathcal{L} = \{\lambda_1, \lambda_2, ..., \lambda_m\}$ denotes the predefined label set. A ranking can be represented as a permutation π on $\{1, 2, ..., m\}$, such that $\pi(j) = \pi(\lambda_j)$ is the position of the label λ_j in the ranking. Therefore, for an unseen instance $\mathbf{x} \in \mathcal{X}$, the learned label ranking function $f(.)$ predicts a permutation $f(\mathbf{x}) = (y_i^{\pi(1)}, y_i^{\pi(2)}, ..., y_i^{\pi(m)}) \in \mathcal{Y}$ as the output.

5) Sequence Alignment Learning: Sequence alignment learning aims at predicting sequence of multiple labels given an input instance. Let c denotes the number of labels in total, then the output vector has the constraint $\mathbf{y}_i \in \mathcal{Y} = \{0, 1, ..., c\}^m$. In sequence alignment learning, m may vary depending on the input. For an unseen instance $\mathbf{x} \in \mathcal{X}$, the learned sequence alignment function $f(.)$ outputs $f(\mathbf{x}) \in \mathcal{Y}$, where all of the predicted labels in the output vector form the predicted sequence for $\mathbf{x}$.

6) Network Analysis: Network analysis aims at investigating the network structures including relationships and interactions between objects and entities. Let $G = (V, E)$ denotes the graph of the network. V is the set of nodes to represent objects and E is the set of edges to represent the relationships between objects. Link prediction is a typical problem in network analysis. Given a snapshot of a network, the goal of link prediction is to infer whether a connection exists between two nodes. The output vector $\mathbf{y}_i \in \mathcal{Y} = \{-1, +1\}^m$ is a binary vector with the value that represents whether there will be an edge $e = (u, v)$ for any pair of nodes $u, v \in V$ and $e \notin E$. m is the number of node pairs that are not appeared in the current graph G and each dimension in $\mathbf{y}_i$ represents a pair of nodes that are not currently connected.

7) Data Generation: Data generation is usually based on Adversarial Generative Network (GAN) to generate data such as text, image, audio and etc, and the multiple labels become the different words in the vocabulary, pixel values, audio values and etc, respectively. Take image generation for an example, the output vector has the constraint $\mathbf{y}_i \in \mathcal{Y} = \{0, 1, ..., 255\}^{m_w \times m_h \times 3}$, where m_w and m_h are the width and height of the image. For an unseen instance $\mathbf{x} \in \mathcal{X}$ which is usually a random noise or an embedding vector with some constraints, the learned GAN-based network $f(.)$ outputs $f(\mathbf{x}) \in \mathcal{Y}$, where all of the predicted pixel values in the output vector form the generated image for $\mathbf{x}$.

8) Semantic Retrieval: Here we consider the semantic retrieval as in the setting where each input instance has semantic labels which can be used to facilitate retrieval tasks [53]. Thus each instance associates with the representation of the semantic labels as the output $\mathbf{y}_i \in \mathcal{Y} = \mathbb{R}^m$. Given an unseen instance $\mathbf{x} \in \mathcal{X}$ as the query, the learned retrieval function $f(.)$ predicts a real-valued vector $f(\mathbf{x}) \in \mathcal{Y}$ as the intermediate output result. The intermediate output vector can then be used to retrieve a list of similar data instances from the database by using a proper distance-based retrieval method.

9) Time-series Prediction: In time-series prediction, the input is a series of data vectors within a period of time, and the output is the data vector at a future timestamp. Let t denotes the time index. The output vector at time t is represented as $\mathbf{y}_i^t \in \mathcal{Y} = \mathbb{R}^m$. Therefore, the outputs within a period of time from $t = 0$ to $t = T$ is $\mathbf{y}_i = (\mathbf{y}_i^0, ..., \mathbf{y}_i^t, ...\mathbf{y}_i^T)$. Given previously observed values, the learned time-series function outputs predicted consecutive future values.

C. Special Cases of Multi-output Learning

1) Multi-class Classification: Multi-class classification can be categorized as traditional single-output learning paradigm if we represent the output class by the integer encoding. It can also be concluded to multi-output learning paradigm if each output class is represented by the one-hot vector.

2) Fine-grained Classification: Though the vector representation is the same for fine-grained classification outputs to the multi-class classification outputs, their internal structures of the vectors are different. Labels under the same parent tend to have closer relationship than the ones under different parents in the label hierarchy.

3) Multi-task Learning: Multi-task learning aims at learning multiple related tasks simultaneously. Each task outputs one single label or value. It can be subsumed under multi-output learning paradigm as learning multiple tasks is similar to learning multiple outputs. It leverages the relatedness between tasks to improve the performance of learning models. One major difference between multi-task learning and multi-output learning is that different tasks might be trained on different training sets or features in multi-task learning, while output variables usually share the same training data or features in multi-output learning.

D. Model Evaluation Metrics

In this section, we introduce the evaluation metrics used to assess the learning models of multi-output learning when apply to unseen or test dataset of size. Let $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{y}_i)|1 \leq i \leq N\}$ be the test dataset. $f(.)$ is the multi-output learning model and $\hat{\mathbf{y}}_i = f(\mathbf{x}_i)$ be the predicted output by $f(.)$ for the testing example $\mathbf{x}_i$. In addition, let Y_i and $\hat{Y}_i$ denote the set of labels corresponding to $\mathbf{y}_i$ and $\hat{\mathbf{y}}_i$, respectively (which will only be used in classification-based metrics).

1) Classification-based Metrics: Classification-based metrics evaluate the performance of multi-output learning with respect to classification problems such as multi-label classification, object recognition, label ranking and etc. The outputs are usually in discrete values. There are a number of ways to average the metrics, *example-based*,*label-based* and *ranking-based*. *Example-based metrics* evaluate the performance of multi-output learning on the performance of each data instance. *Label-based metrics* evaluate the performance with respect to each label. *Ranking-based metrics* evaluate the performance in terms of the ordering of the output labels.

(a) Example-based Metrics

Exact Match Ratio computes the percentage of correct predictions. A prediction is correct only if it is an exact match of the corresponding output of the ground truth. Exact Match Ratio in multi-output learning extends the accuracy of single output learning and it is defined as follows:

$$ExactMatchRatio = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\mathbf{y}_i = \hat{\mathbf{y}}_i)$$

where $\mathbb{I}$ is the indicator function, *i.e.*,$\mathbb{I}(g) = 1$ if g is true, and 0 otherwise. Exact Match Ratio has disadvantage that it treats the partially correct predicted outputs as incorrect predictions.

Accuracy computes the percentage of correct output labels over the total number of predicted and true labels. It considers partially correct predicted outputs. Accuracy averaged over all data instances is defined as follows:

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N} \frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|}$$

It is also a good measure when the output labels in the data are balanced.

Precision is the percentage of outputs found by the learning model that are correct. It is computed as the number of true positives over the number of true positives plus the number of false positives.

$$Precision = \frac{1}{N}\sum_{i=1}^{N} \frac{|Y_i \cap \hat{Y}_i|}{|\hat{Y}_i|}$$

Recall is the percentage of outputs present in the ground truth that are found by the model. It is computed as the number of true positives over the number of true positives plus the number of false negatives.

$$Recall = \frac{1}{N}\sum_{i=1}^{N} \frac{|Y_i \cap \hat{Y}_i|}{|Y_i|}$$

F_1 **Score** is the harmonic mean of precision and recall.

$$F_1 Score = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y_i \cap \hat{Y}_i|}{|Y_i| + |\hat{Y}_i|}$$

Hamming Loss computes the average measure of difference between the predicted and the actual outputs. It considers both prediction error (an incorrect label is predicted) and omission error (a corrected label is not predicted). Hamming loss averaged over all data instances is defined as:

$$HammingLoss = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{m}|Y_i \Delta \hat{Y}_i|$$

where m is the number of labels and Δ represents the symmetric difference of two sets. The lower the hamming loss, the better the performance of the model is.

(b) Label-based Metrics. There are two averaging approaches for label-based metrics, *macro* and *micro* averaging. Specifically, macro-based approach computes the metrics for each label independently and then average over all the labels. So it gives equal weights to each label. Micro-based approach, on the other hand, give equal weights to every data samples. It aggregates the contributions of all the labels to compute the averaged metric. We denote TP_l, FP_l, TN_l and FN_l as the number of true positives, true negatives, false positives and false negatives, respectively, for each label. Let B be the binary evaluation metric (accuracy, precision, recall or F_1 score) for a particular label. Macro and micro approaches are defined in the following.

Macro-based

$$B_{macro} = \frac{1}{m}\sum_{l=1}^{m} B(TP_l, FP_l, TN_l, FN_l)$$

Micro-based

$$B_{micro} = B(\frac{1}{m}\sum_{l=1}^{m} TP_l, \frac{1}{m}\sum_{l=1}^{m} FP_l, \frac{1}{m}\sum_{l=1}^{m} TN_l, \frac{1}{m}\sum_{l=1}^{m} FN_l)$$

(c) Ranking-based Metrics

One-error is defined as the number of times the top-ranked label is not in the true label set. It only considers the most confident predicted label of the model. The averaged one-error over all data instances is computed as:

$$\textit{One-error} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\arg\min_{\lambda \in \mathcal{L}} \pi_i(\lambda) \notin Y_i)$$

where $\mathbb{I}$ is the indicator function, $\mathcal{L}$ denotes the label set and $\pi_i(\lambda)$ is the predicted rank of the label λ for a test instance $\mathbf{x_i}$. The smaller the one-error, the better the performance of the model is.

Ranking Loss reports the average proportion of incorrectly ordered label pairs.

$$RankingLoss = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i||\overline{Y_i}|}|E|, where$$
$$E = (\lambda_a, \lambda_b) : \pi_i(\lambda_a) > \pi_i(\lambda_b), (\lambda_a, \lambda_b) \in Y_i \times \overline{Y_i}$$

where $\overline{Y_i} = \mathcal{L} \setminus Y_i$. The smaller the ranking loss, the better the performance of the model.

Average Precision (AP) is defined as the proportion of the labels ranked above a particular label in the true label set, and then it averages over all the true labels. The larger the value, the better the performance of the model is. The averaged average precision over all test data instances is defined as follows:

$$AP = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{\lambda \in Y_i}\frac{\{\lambda' \in Y_i | \pi_i(\lambda') \leq \pi_i(\lambda)\}}{\pi_i(\lambda)}$$

2) Regression-based Metrics: Regression-based metrics evaluate the performance of multi-output learning with respect to regression problems such as multi-output regression, object localization, image generation and etc. The outputs are usually in real values. We list several popular regression-based metrics in the following. These metrics are averaged over all the testing data instances and all the outputs.

Mean Absolute Error (MAE) is the simplest regression metric that computes the absolute difference between the predicted and the actual outputs.

$$MAE = \frac{1}{m}\frac{1}{N}\sum_{i=1}^{N}|\mathbf{y}_i - \hat{\mathbf{y}}_i|$$

Mean Squared Error (MSE) is one of the most widely used evaluation metric in regression tasks. It computes the average squared difference between the predicted and the actual outputs. Outliers in the data using MSE will contribute much higher error than they would using MAE.

$$MSE = \frac{1}{m}\frac{1}{N}\sum_{i=1}^{N}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2$$

Average Correlation Coefficient (ACC) measures the degree of association between the actual and the predicted outputs.

$$ACC = \frac{1}{m}\sum_{l=1}^{m}\frac{\sum_{i=1}^{N}(y_i^l - \bar{\mathbf{y}}^l)(\hat{y}_i^l - \bar{\hat{\mathbf{y}}}^l)}{\sqrt{\sum_{i=1}^{N}(y_i^l - \bar{\mathbf{y}}^l)^2 \sum_{i=1}^{N}(\hat{y}_i^l - \bar{\hat{\mathbf{y}}}^l)^2}}$$

where y_i^m and $\hat{y}_i^m$ are the actual and predicted m output of $\mathbf{x_i}$, respectively, and $\bar{\mathbf{y}}^l$ and $\bar{\hat{\mathbf{y}}}^l$ are the vectors of averages of the actual and predicted outputs for label l over all the samples.

Intersection over Union threshold (IoU) is specifically designed for object localization or segmentation. It is computed as the

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

where *Area of Overlap* is computed as the area of intersection between the predicted and the actual bounding boxes or segmentation masks and *Area of Union* is computed as the union area of them.

TABLE II
CHARACTERISTICS OF THE DATASETS OF MULTI-OUTPUT LEARNING TASKS.

Multi-output Characteristic	Challenge	Application Domain	Dataset Name	Statistics	Source
Volume	Extreme Output Dimension[1]			**Output Dimension**	
		Review Text	AmazonCat-13K	13,330	[54]
		Review Text	AmazonCat-14K	14,588	[55], [56]
		Text	Wiki10-31	30,938	[57], [58]
		Social Bookmarking	Delicious-200K	205,443	[57], [59]
		Text	WikiLSHTC-325K	325,056	[60], [61]
		Text	Wikipedia-500K	501,070	Wikipedia
		Product Network	Amazon-670K	670,091	[54], [57]
		Text	Ads-1M	1,082,898	[60]
		Product Network	Amazon-3M	2,812,281	[55], [56]
	Extreme Class Imbalance			**Largest Class Imbalance Ratio**	
		Scene Image	WIDER-Attribute	1:28	[62]
		Face Image	Celeb Faces Attributes	1:43	[63]
		Clothing Image	DeepFashion	1:733	[64]
		Clothing Image	X-Domain	1:4,162	[65]
	Unseen Outputs			**Seen / Unseen Labels**	
		Image	Attribute Pascal abd Yahoo	20 / 12	[66]
		Animal Image	Animal with Attributes	40 / 10	[66]
		Scene Image	HSUN	80 / 27	[67]
		Music	MagTag5K	107 / 29	[68]
		Bird Image	Caltech-UCSD Birds 200	150 / 50	[69]
		Scene Image	SUN Attributes	645 /72	[19]
		Health	MIMIC II	3,228 / 355	[70]
		Health	MIMIC III	4,403 / 178	[71]
Velocity	Change of Output Distribution			**Time Periods**	
		Text	Reuters	365 days	[72]
		Route	ECML/PKDD 15: Taxi Trajectory Prediction	365 days	[73]
		Route	epfl/mobility	30 days	[74]
		Electricity	Portuguese Electricity Consumption	365 days	[75]
		Traffic Video	MIT Traffic Data Set	90 minutes	[40]
		Surveillance Video	VIRAT Video	8.5 hours	[76]
Variety	Complex Structures			**Output Structures**	
		Image	LabelMe	Label, Bounding Box	[9]
		Image	ImageNet	Label, Bounding Box	[8]
		Image	PASCAL VOC	Label, Bounding Box	[77]
		Image	CIFAR100	Hierarchical Label	[78]
		Lexical Database	WordNet	Hierarchy	[79]
		Wikipedia Network	Wikipedia	Graph, Link	[80]
		Blog Network	BlogCatalog	Graph, Link	[81]
		Author Collaboration Network	arXiv-AstroPh	Link	[82]
		Author Collaboration Network	arXiv-GrQc	Link	[82]
		Text	CoNLL-2000 Shared Task	Text Chunks	[83]
		Text	Wall Street Journal (WSJ) corpus	POS Tags, Parsing Tree	-
		European Languages	Europarl corpus	Sequence	[2]
Veracity	Noisy Outut Labels			**Noisy Labeled Samples**	
		Dog Image	AMT	7,354	[84]
		Food Image	Food101N	310K	[85]
		Clothing Image	Clothing1M	1M	[86]
		Web Image	WebVision	2.4M	[87]
		Image and Video	YFCC100M	100M	[88]

E. Multi-output Learning Datasets

In this subsection, we present an analysis on the development of multi-output learning in terms of different challenges organized by the 4Vs. We focus on the popular datasets where most of them are used in the last decade in terms of the year they were introduced. Table II presents the detailed characteristic of the datasets including the *multi-output characteristic*, *challenge*, *application domain*, *dataset name*, *statistics* specifically depending on the challenge, and *source*.

For challenges caused by large volume, *extreme output dimension*, *extreme class imbalance* and *unseen outputs*, the datasets are ordered according to their corresponding statistics, *output dimension*, *largest class imbalance ratios* and *seen/unseen labels*, respectively. Such extreme large and increasing statistics well illustrates the pressing need to overcome the challenges caused by large volume of output labels. Many works of *change of output distribution* adopts synthetic streaming data or static database in the experiment. We list some popular real-world dynamic databases that can be used for multi-output learning with changing output distributions. As shown in the table, the datasets are from various application domains which shows the importance of this challenge. The datasets from *complex structures* have different output structures, and even one dataset could provide multiple output

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

structures depending on the tasks. For example, the image datasets listed in the table provides labels and bounding box of the objects. Many works of noisy labels evaluate their methods on clean-label real-world datasets with artificial noise added by authors in the experiment. Here we list several popular real-world datasets with some unknown annotation error obtained during label annotation.

IV. The Challenges of Multi-output Learning and Representative Works

The pressing need for the complex prediction output and the explosive growth of output labels pose several challenges to multi-output learning and have exposed the inadequacies of many learning models that exist to date. In this section, we discuss each of these challenges and review several representative works on how they cope with the emerging phenomenons. Furthermore, with the success of Artificial Neural Network (ANN), we also present several state-of-the-art works on multi-output learning using ANN for each challenge.

A. Extreme Output Dimension

Large scale datasets are ubiquitous in the real-world applications. The dataset is defined to be in large-scale in terms of three factors, large number of data instances, high dimensionality of the input feature space, and high dimensionality of the output space. There are many research works that focus on solving the scalability issues caused by large number of data instances, such as instance selection methods [89], or high dimensionality of the feature space, such as feature selection methods [90]. The cause of high output dimensions has received much less attention.

For m-dimensional output vectors, if the label for each dimension can be selected from a label set with c different labels, then the number of output outcomes is c^m. For ultrahigh output dimensions or number of labels, the output space can be extremely large, which will result in inefficient computation. Therefore, it is crucial to design multi-output learning models to handle the immense growth of the outputs.

We provide some insights into the current state-of-the-art research on ultrahigh output dimension for multi-output learning by conducting an analysis on the output dimensionality. The analysis is based on the datasets used in the studies of multiple disciplines, such as machine learning, computer vision, natural language processing, information retrieval and data mining. Specifically, it focuses on three top journals and three top international conferences, namely, *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, *IEEE Transactions on Neural Networks and Learning Systems(TNNLS)*, *Journal of Machine Learning Research(JMLR)*, *International Conference on Machine Learning(ICML)*, *Conference on Neural Information Processing Systems(NIPS)* and *Conference on Knowledge Discovery and Data Mining (KDD)*. The analysis is summarized in Fig. 5 and 4. From these two figures, it is evident that the output dimensionality of the algorithms understudied keeps increasing over time. In addition, all of these selected conferences and journals were dealing with more than a million of output dimensions in their latest works and they are approaching billions of outputs. Compared with output dimensions of the methods from the three journals, the ones from the selected conferences are increasing exponentially over time, and some works from KDD even already reached billion outputs. In conclusion, the explosion of the output dimensionality drives the development of the current learning algorithms.

We review the works of handling high output dimension from two perspectives, qualitative and quantitative approaches. In the qualitative approach, we review the generative models on their generation of data with increasing dimensional outputs. In the quantitative models, we review the discriminative models on how they handle the increasing number of outputs while preserving expected information. The main difference between a generative and a discriminative model is that the generative model focuses on learning the joint probability $P(x, y)$ of the inputs x and the label y, while the discriminative model focuses on the posterior $P(y|x)$. Note that in generative models, $P(x, y)$ can be used to generate data x, where x is the generated output in this particular case.

1) Qualitative Approach - Generative Model: Image synthesis [44], [192] aims at synthesizing new images from textual image description. Some pioneer works have done image synthesis with the image distribution as the multiple outputs using GAN [193]. But in real life, GAN can only generate low resolution images. There are some works trying to scale up the GAN on generating high-resolution images with sensible outputs. Reed et.al [44] proposed a GAN architecture that can generate visually plausible 64 x 64 images given texts. Then their follow-up work GAWWN [192] scales the synthesized image up to 128 x 128 resolution, leveraging the additional annotations. After that, StackGAN [194] was proposed, to generate photo-realistic images with 256 x 256 resolution from text descriptions. HDGAN [195], as the state-of-the-art image synthesis method, models the high resolution image statistics in an end-to-end fashion which outperforms StackGAN and is able to generate 512 x 512 images. The resolution of the generated image is still increasing.

MaskGAN [196] adopts GAN to generate promising text, which is a sequence of words. The label set is with the vocabulary size. The output dimension is the length of word sequence that is generated, which technically can be unlimited. However, MaskGAN only handles sentence-level text generation. Document-level and book-level text generations are still challenging.

2) Quantitative Approach - Discriminative Model: Similar to instance selection methods which reduce the number of input instances and feature selection methods which reduce the input dimensionality, it is natural to design models to reduce the output dimensionality for high-dimensional outputs. Embedding methods can be used to compress a space by projecting the original space onto a lower-dimensional space, with expected information preserved, such as label correlation and neighbourhood structure. Popular methods such as random projections or canonical correlation analysis (CCA) based projections [197]–[200] can be adopted to reduce the dimensions of the output label space. As a result, these modeling tasks can be performed on a compressed output label space and then

Fig. 4. Trends of dataset output dimensions used in publications that appeared in three machine learning journals (TPAMI, TNNLS and JMLR) from year 2013 to present [91]–[145].

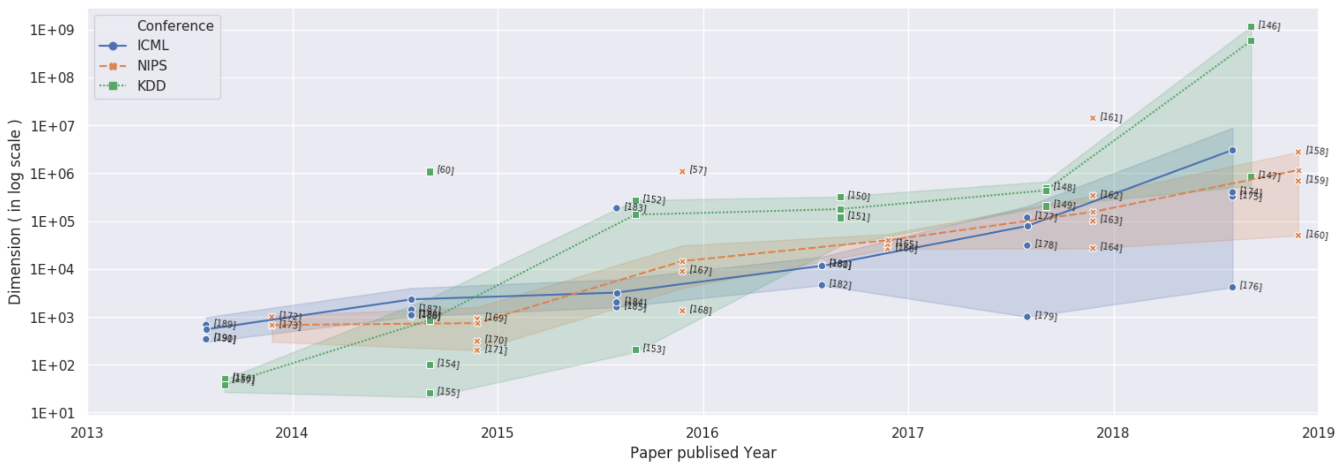

Fig. 5. Trends of dataset output dimensions used in publications that appeared in three machine learning conferences (ICML, NIPS and KDD) from year 2013 to present [57], [60], [146]–[191].

project the predicted compressed label back to the original high-dimensional label space. Recently, several embedding methods are proposed for extreme output dimension. [201] proposes a novel randomized embeddings for extremely large output space. AnnexML [149] is a novel graph embedding method that is constructed based on k-nearest neighbors of the label vectors and captures the graph structure in the embedding space. Efficient prediction is conducted by using an approximate nearest neighbor search method. Two of the popular ANN methods for handling extreme output dimension are fastText Learn Tree [202] and XML-CNN [203]. fastText Learn Tree [202] learns the data representation and the tree structure jointly. The learned tree structure can be used for efficient hierarchical prediction. XML-CNN is a CNN-based model that adopts a dynamic max pooling scheme to capture fine-grained features from regions of the input document. A hidden bottleneck layer is used to reduce the model size.

B. Complex Structures

With the increasing abundance of labels, there is a pressing need to understand their inherent structures. Complex structures could lead to multiple challenges in multi-output learning. It is common that there exists strong correlation and complex dependencies between the labels. Therefore, appropriately modeling output dependencies is critical in multi-output learning. In addition, designing the multivariate loss function specifically to the given data and task is of importance. Furthermore, proposing efficient algorithm to alleviate the high complexity caused by complex structures is also challenging.

1) Appropriate Modeling of Output Dependencies: The simplest method for multi-output learning is to decompose the multi-output learning problem into m independent single-output problems and each single-output problem corresponds to a single value in the output space. A representative work is Binary Relevance (BR) [204], which independently learns the binary classifiers for all the labels in the output space. For an unseen instance $\mathbf{x}$, BR predicts the output labels by predicting each of the binary classifiers and then aggregating the predicted labels. However, such independent models do not consider the dependencies between outputs. A set of predicted output labels might be assigned to the testing instance even though these labels never co-occur in the training set. Therefore, it is crucial to model the output dependencies appropriately to obtain better performance for multi-output tasks.

To model for multiple outputs with interdependencies, many classic learning methods are proposed, such as Label Powerset (LP) [205], Classifier Chains (CC) [206], [207], Structured SVMs (SSVM) [208], Conditional Random Fields (CRF) [209] and etc. LP models the output dependencies by considering each of different combination of labels in the output space as a single label, and thus transform the problem into learning multiple single-label classifiers. The number of single-label classifiers to be trained is the number of label combinations, which grows exponentially with the number of labels. Therefore, LP has the drawback of high computation in training with large number of output labels. Random k-Labelsets [210], an ensemble of LP classifiers, is a variant of LP to alleviate this problem by training each LP classifier on a different random subset of labels.

CC improves BR to take into account the output correlations by linking all the binary classifier from BR into a chain via modified feature space. For the jth label, the instance $\mathbf{x}_i$ is augmented with the 1st, 2nd, ... $(j-1)$th label $(\mathbf{x}_i, l_1, l_2, ..., l_{j-1})$ as the input, to train the jth classifier. Given an unseen instance, CC predicts the output using the 1st classifier, and then augment the instance with the prediction from the 1st classifier as the input to the 2nd classifier for predicting the next output. CC processes values in this way from the 1st classifier to the last one, to preserve the output correlations. However, different order of chains leads to different results. ECC [206], an ensemble of classifier chains, is proposed to solve this problem. It trains the classifiers over a set of random ordering chains and averages the results. Probabilistic Classifier Chains (PCC) [211] provides a probabilistic interpretation of CC and estimates the joint distribution of output labels to capture the output correlations. CCMC [94] is a classifier chain model that considers the order of label difficulties to tackle the problem of confusing labels in harming the performance of multi-output tasks. It is an easy-to-hard learning paradigm that identifies easy and hard labels and use the predictions of easy labels to help solve hard labels.

SSVM leverages the idea of large margin to deal with multiple outputs with interdependencies. Its compatibility function is defined as $F(\mathbf{x}, \mathbf{y}) = \mathbf{w}^T\Phi(\mathbf{x}, \mathbf{y})$ where $\mathbf{w}$ is the weight vector and $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^q$ is the joint feature map over input and output pairs. The formulation of SSVM is listed in the following:

$$\begin{aligned} &\min_{\mathbf{w}\in R^q, \{\xi_i \geq 0\}_{i=1}^n} \|\mathbf{w}\| + \frac{C}{n}\sum_{i=1}^{n}\xi_i^2 \\ &s.t.\ \mathbf{w}^T\Phi(\mathbf{x}_i, \mathbf{y}_i) - \mathbf{w}^T\Phi(\mathbf{x}_i, \mathbf{y}) \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \\ &\qquad \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i, \forall i. \end{aligned} \tag{1}$$

where $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function, C is a positive constant that controls the trade-off between the loss function and the regularizer, n is the number of training samples and ξ_i is the slack variable. In practice, SSVM is solved by Cutting-Plane algorithm [212].

Apart from the classic models to model output correlations, there are many state-of-the-art multi-output learning works proposed based on ANN. For example, convolutional neural networks (CNNs) based models focus on modeling outputs such as hierarchical multi-labels [213], ranking [214], bounding box and etc. Recurrent neural networks (RNNs) based models focus on sequence-to-sequence learning [215] and time-series prediction [216]. Generative deep neural networks can be used to generate outputs such as image, text and audio [193].

2) Multivariate Loss Functions: A loss function computes the difference between the actual output and the predicted output. Different loss functions give different errors given the same dataset, and thus they greatly affect the performance of the model.

0/1 loss is a standard loss function that is commonly considered in classification [217]:

$$L_{0/1}(\mathbf{y}, \mathbf{y}') = \mathbb{I}(\mathbf{y} \neq \mathbf{y}') \tag{2}$$

where $\mathbb{I}$ is the indicator function. In general, 0/1 loss refers to the number of misclassified training examples. However, it is very restrictive and does not consider label dependency and therefore is not suitable for large number of outputs and outputs with complex structures. In addition, it is non-convex and non-differentiable, so it is difficult to minimize the loss using the standard convex optimization methods. In practice, one typically uses a surrogate loss, which is convex upper bound of the task loss. However, a surrogate loss in multi-output learning usually loses the consistency in generalizing single-output methods to deal with multiple outputs. Consistency is defined as the convergence of the expected loss of a learned model to the Bayes loss in the limite of infinite training data. Some works on different problems of multi-output learning study the consistency of different surrogate functions and show that they are consistent under some sufficient conditions [218]–[220]. But it still remains challenging and requires exploration on the theoretical consistency of different problems for multi-output learning. Here we introduce four popular surrogate loss, *hinge loss*, *negative log loss*, *perceptron loss* and *softmax-margin loss* in the following.

Hinge loss is one of the most widely used surrogate loss. It is usually used in Structured SVMs [221]. It promotes the score of the correct outputs to be greater than the score of the predicted outputs:

$$L_{Hinge}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \max_{\mathbf{y}'\in\mathcal{Y}}[\Delta(\mathbf{y}, \mathbf{y}') + \mathbf{w}^T\Phi(\mathbf{x}, \mathbf{y}')] - \mathbf{w}^T\Phi(\mathbf{x}, \mathbf{y}) \tag{3}$$

The margin (or task loss), $\Delta(\mathbf{y}, \mathbf{y}')$, can be defined differently depending on the output structures and the tasks. Here we list several examples.

- For sequence learning or outputs with equal weights, $\Delta(\mathbf{y}, \mathbf{y}')$ can be simply defined as the Hamming loss $\sum_{j=1}^{m} \mathbb{I}(\mathbf{y}_{(j)} \neq \mathbf{y}'_{(j)})$.
- For taxonomic classification with the hierarchical output structure, $\Delta(\mathbf{y}, \mathbf{y}')$ can be defined as the tree distance between $\mathbf{y}$ and $\mathbf{y}'$ [18].
- For ranking, $\Delta(\mathbf{y}, \mathbf{y}')$ can be defined as the mean average precision of ranking $\mathbf{y}'$ compared to the optimal $\mathbf{y}$ [222].
- In syntactic parsing, $\Delta(\mathbf{y}, \mathbf{y}')$ is defined as the number of labeled spans where $\mathbf{y}$ and $\mathbf{y}'$ do not agree [31].
- Non-decomposable losses such as F_1 measure, Average Precision (AP), intersection over union (IOU) can also be defined as the margin.

Negative log loss is commonly used in CRFs [209]. Minimizing negative log loss is the same as maximizing the log probability of the data.

$$L_{NegativeLog}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \log \sum_{\mathbf{y}' \in \mathcal{Y}} \exp[\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}')] - \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) \tag{4}$$

Perceptron loss is usually adopted in Structured Perceptron [223]. It is the same as hinge loss without the margin..

$$L_{Perceptron}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \max_{\mathbf{y}' \in \mathcal{Y}} [\mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}') - \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y})] \tag{5}$$

Softmax-margin loss is one of the most popular loss used in multi-output learning models such as SSVMs and CRFs [224], [225].

$$L_{SoftmaxMargin}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \log \sum_{\mathbf{y}' \in \mathcal{Y}} \exp[\Delta(\mathbf{y}, \mathbf{y}') + \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}')] - \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) \tag{6}$$

Squared loss is a popular and convenient loss function that penalizes the difference between the ground truth and the prediction quadratically. It is commonly used in the traditional single-output learning and can be easily extended to multi-output learning by summing up the squared differences over all the outputs:

$$L_{Squared}(\mathbf{y}, \mathbf{y}') = (\mathbf{y} - \mathbf{y}')^2 \tag{7}$$

It is usually adopted in multi-output learning for continuous valued outputs or continuous intermediate results before converting to discrete valued outputs. It is also commonly used in neural networks and boosting.

3) Efficient Algorithms: Complex output structures significantly increase the burden to the algorithms of the model formulation, due to the large-scale outputs, the complex output dependencies, or the complex loss functions. There are efficient algorithms proposed to tackle the aforementioned challenge. Many of them leverage the classic machine learning models to speed up the algorithms in order to alleviate the burden. We present four most widely used classic models, k nearest neighbor (kNN) based, decision tree based, k-means based and hashing based methods.

1) kNN based methods. kNN is a simple yet powerful machine learning model. It conducts prediction based on k instances pertaining to the smallest Euclidean distance of each instance vector to the test instance vector. LMMO-kNN [226] is an SSVM-based model involves an exponential number of constraints *w.r.t.* the number of labels. It imposes a kNN constraints instantiated by the label vectors from neighbouring examples, to significantly reduce the training time and achieve a rapid prediction.
2) Decision tree based methods [227], [228] learn a tree from the training data with hierarchical output label space. They recursively partition the nodes until each leaf contains small number of labels. Each novel data point is passed down the tree until it reaches a leaf. In such a way, they usually can achieve logarithmic time prediction.
3) k-means based methods such as SLEEC [57] cluster the training data using k-means clustering. SLEEC learns a separate embedding per cluster and performs classification for a novel instance within its cluster alone. It significantly reduces the prediction time.
4) Hashing based methods such as Co-Hashing [229], [230] and DBPC [231] reduces the prediction time using hashing on the input or the intermediate embedding space. Co-Hashing learns an embedding space to preserve semantic similarity structure between inputs and outputs. It then generates compact binary representations for the learned embeddings for prediction efficiency. DBPC jointly learns the deep latent hamming space and binary prototypes while capturing the latent nonlinear structure of the data in ANN. The learned hamming space and binary prototypes allow the prediction complexity to be significantly decreased and the memory/storage cost to be saved.

C. Extreme Class Imbalance

Real-world multi-output applications rarely provide the data with equal number of training instances for all the labels/classes. The distributions of data instances between classes are often imbalanced in many applications. Therefore, traditional models learned from such data tend to favor majority classes more. For example, in image annotation, the learned models tend to annotate an image by the majority labels appeared in the training set. In face generation, the learned systems tend to generate famous faces that are dominated by the model. Though the class imbalance problem is well studied in binary classification, it still remains challenging in multi-output learning, especially when the imbalance is extreme.

Many of the existing multi-output learning works either assume balanced classes in training data or ignore the imbalanced class learning problem. To balance the class distributions, one natural way is to conduct re-sampling of the data space, such as SMOTE [232] which adopts the over-sampling technique on minority classes or [233] which down-samples the majority classes. There are other techniques of handling class imbalance in ANN. [234] adopts incremental rectification of mini-batches for deep neural network. It proposes hard sample mining strategy to minimize the dominant effect of the majority classes by discovering sparsely sampled minority class boundaries. Both [235] and [236] leverage adversarial training to mitigate the imbalanced class problem by re-weighting technique, so that majority classes tend to have similar impact to minority classes.

D. Unseen Outputs

Traditional multi-output learning assumes that the output set in testing is the same to the one in training, *i.e.*, the output labels of a testing instance have already appeared during training. However, this is not true in real-world applications. For example, a new emerging living species can not be detected by the classifier learned based on existing living animals. It is unfeasible to recognize the actions or events in the real-time video if no such actions or events with the same labels appear in the training video set before. A coarse animal classifier is unable to provide the detailed fine-grained species of a detected animal, such as whether a dog is labrador or shepherd.

Depending on the complexity of the learning task, label annotation is usually very costly. In addition, the explosive growth of labels not only lead to high dimensional output space as a result of computation inefficiency, but also make supervised learning tasks challenging due to unseen output labels during testing.

1) Zero-shot Multi-label Classification: Multi-label classification is a typical multi-output learning problem. There are various input for the multi-label classification problem, such as text, image, video and etc. depending on the application. The output for each input instance is usually a binary label vector, indicating what labels are associated with the input. Multi-label classification problem learns a mapping from the input to the output. However, as the label space increases, it is common to have unseen output labels during testing, where no such labels have appeared in the training set. In order to study such case, zero-shot multi-class classification problem is firstly proposed [16], [237] and most of them leverage the pre-defined semantic information such as attributes [12], word representations [14] and etc. It is then extended to zero-shot multi-label classification to assign multiple unseen labels to an instance. Similarly, zero-shot multi-label learning leverages the knowledge of the seen and unseen labels and models the relationships between the input features, label representations and labels. For example, [238] leverages the co-occurrence statistics of seen and unseen labels and models the label matrix and co-occurrence matrix jointly using a generative model. [239] and [240] incorporate knowledge graphs of label relationships in neural networks.

2) Zero-shot for Action Localization: Similar to zero-shot classification problem, localization of human actions in videos without any training video examples is a challenging task. Inspired by zero-shot image classification, many zero-shot action classification works predict unseen actions from disjunct training actions based on the prior knowledge of action-to-attribute mappings [241]–[243]. Such mappings are usually predefined and link the seen and unseen actions by a description of attributes. Thus they can be used to generalize to undefined actions and unable to localize actions. More recently, some works are proposed to overcome the issue. Jain *et al.* [244] proposes Objects2action without using any video data or action annotations. It leverages vast object annotations, images and textual descriptions that can be obtained from open-source collections such as WordNet and ImageNet. Mettes and Snoek [245] then enhances Objects2action by considering the relations between actors and objects.

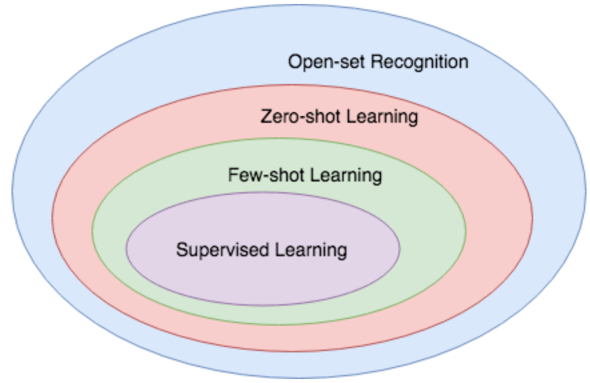

Fig. 6. Relationship among different levels of unseen outputs. All of these learning problems belong to multi-output learning.

3) Open-set Recognition: Traditional multi-output learning problem including zero-shot multi-output learning is under the closed-set assumption, where all the testing classes are known in training time, by either with training samples or pre-defined in a semantic label space. Scheirer *et al.* propose open-set recognition [15] to describe the scenario where unknown classes appear in testing. It presents 1-vs-set machine to classify the known classes as well as deal with the unknown classes. Scheirer *et al.* then extended the work to multi-class settings [246], [247] by formulating a compact abating probability model. [248] adapts ANN for open-set recognition by proposing a new model layer which estimates the probability of an input being an unknown class.

Fig. 6 illustrates the relationships among different levels of unseen outputs for multi-output learning. Open-set recognition is the most generalized problems of all. Few-shot and zero-shot learning are studied for different multi-output learning problems such as multi-label learning and event localization. However, currently open-set recognition is only studied for the setting of multi-class classification and other problems of multi-output learning are still unexplored.

E. Noisy Output Labels

Efficient label annotation methods, such as crowdsourcing, might lead to issues of noisy labels. Noisy labels are caused by various reasons such as weak associations or text ambiguity [249]. It is necessary to handle noisy data including corruptions (such as incorrect labels and partial labels) or missing labels.

1) Missing Labels: Missing labels exist in many real-world applications. For example, human annotators usually annotate an image or document by prominent labels and often miss out some labels less emphasized in the input or out of their knowledge. Objects in an image may not all be localized due to the reasons such as too many or too small objects. Social

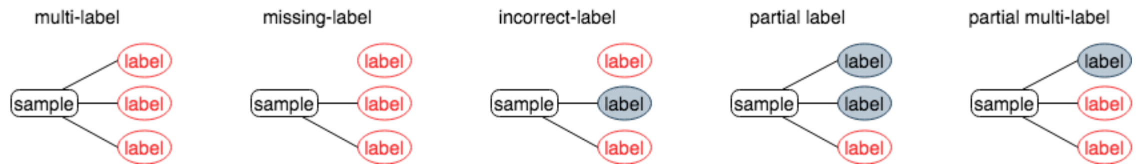

Fig. 7. Range of noisy labels in multi-label classification. Training may be: **multi-label** (sample associates with multiple labels), **missing-label** (sample has incomplete label assignment), **incorrect-label** (sample has at least one incorrect labels and possible incomplete label assignment), **partial-label** (each sample has multiple labels, only one of which is correct), **partial multi-label** (each sample has multiple labels, at least one of which is correct). A line connecting a label with the sample represents that the sample associates with the label. The label in red color represents the correct label to the sample. The label in gray box represents an incorrect label to the sample.

media such as Instagram, provides the tags as labels of an uploaded image by the user. But the tags annotated by the user could cover a wide range of aspects such as event, mood, weather and etc. and it is usually impossible to cover objects appear in the image. Directly using such labeled datasets in traditional multi-output learning models can not guarantee the performance of the given tasks. Therefore, handling missing labels is necessary in most real-world applications.

Missing labels is initially handled by treating them as negative labels [250]–[252] and then the modeling tasks can be performed based on the fully labeled dataset. This way of handling missing labels might introduce undesirable bias to the learning problem. One of the most widely used methods for missing value imputation is matrix completion, including [166], [172], [253], to fill in missing entries. Most of these works are based on low-rank assumption. More recently, label correlations can be leveraged to handle missing labels to improve the learning tasks [254], [255].

2) Incorrect Labels: Many labels in the high dimensional output space are usually non-informative or noisy obtained from the label annotation step. If the annotation step was conducted using crowdsourcing platforms which might hire non-expert workers to do the label annotation, then it is likely that the workers intentionally or unintentionally annotate incorrect labels given an input. If the labeled datasets were obtained from social media, the user provided labels such as tags, captions and etc. may not agree with the input data. The label noise might severely degrade the performance of multi-output learning tasks. A basic approach to handle incorrect labels is to simply remove the data with incorrect labels [256], [257]. But it is usually difficult to detect the harmful mislabeled data. Therefore, designing multi-output learning algorithms that learn from noisy labeled datasets is of great practical importance.

Existing multi-output learning methods handling noisy labels are mainly categorized into two groups. The first group of methods build robust loss functions [258]–[260], which modify the labels in the loss function to alleviate noise effect. The second group of methods model latent labels and learn the transition from the latent labels to noisy labels [261]–[263].

Partial Labels A special case of incorrect labels is partial labels [264], where each training instance associates with a set of candidate labels and only one of them is correct. It is a common problem in the real-world applications. For example, a photograph containing many faces with captions about who are in the photo but it does not match the name with the face. Many studies on partial label learning are developed to recover the ground-truth label from the candidate set [265], [266]. However, the assumption of one exactly ground-truth in the candidate set for each instance may not hold by some label annotation methods. With the use of multiple workers on the crowdsourcing platform to annotate a dataset, the final annotations are usually gathered from the union set of the annotations of all the workers, where each instance might associate with both multiple relevant and irrelevant labels. Recently, Xie and Huang [267] propose a new learning framework, partial multi-label learning (PML), to alleviate the assumption of partial label learning. It leverages the data structure information to optimize the confidence weighted rank loss. We summarize the learning scenarios in terms of noisy output labels in Fig. 7, including multi-label learning, missing labels, incorrect labels, partial label learning and partial multi-label learning.

F. Change of Output Distribution

Many real-world applications are required to deal with data streams, where data arrives continuously and possibly infinitely. The output distribution is thus changing over time and concept drift may also encountered. Examples of such applications include surveillance [76], driver route prediction [73] and demand forecasting [75]. Take visual tracking [268] in surveillance video as an example, the video is potentially infinite. Data streams come in high velocity as the video keeps generating consecutive frames. The goal is to detect, identify and locate events or objects in the video and the learning model is required to adapt to possible concept drift and working under limited memory.

Existing multi-output learning methods model the change of output distribution by update the learning system each time data streams arrive using ensemble-based methods [269]–[273] and ANN-based methods [268], [274]. Strategies to handle concept drift include the assumption of fading effect on the past data [272], maintaining a change detector on the predictive performance measurements and recalibrate models accordingly [271], [275], and using Stochastic Gradient Descent (SGD) method to update the network accommodating new data streams in ANN-based methods [268]. Furthermore, the k nearest neighbor (kNN) is one of the most classic frameworks in handling multi-output problems, but it cannot be successfully adapted to deal with the challenge of change of output distribution due to the inefficiency issue. Many online hashing and online quantization based methods [276],

[277] are proposed to improve the efficiency of kNN while accommodating the changing output distribution.

G. Others

Any two of the aforementioned challenges can be combined to form a more complex challenge. For example, noisy labels and unseen outputs can be combined to form open-set noisy label problem [278]. In addition, the combination of noisy labels and extreme output dimension is also worth studied and explored [186]. Change of output distribution can combine with noisy labels to result in online time-series prediction with missing values which is proposed by [279]. It combining with dynamic label set (unseen outputs) leads to open-world recognition [280], which the open-set recognition problem with incremental labels update. It adds to extreme class imbalance to become a common problem in streaming multi-output learning with concept drift and class imbalance at the same time [17], [281]. Combining complex structures of outputs with change of output distribution is also practical in real-world applications [282].

H. Open Challenges

1) Output Label Interpretation: There are different ways to represent the output labels and each of them express the label information from a specific perspective. Take label tags as the output for an example, binary attributed output embedding represents what attributes the input associates with. Hierarchical label outputs embedding conveys the hierarchical structure of the inputs by the outputs. Semantic word output embedding expresses the semantic relationships between the outputs. Each of them exhibits a certain level of human interpretability. Recently, there are many label embedding works proposed to enhance the label interpretation by incorporating different label information from multiple perspectives [283]. Such output label embedding contains rich information across multiple label contexts. Thus it is a challenging task to appropriately modeling the interdependencies between outputs that well corresponds to human interpretation for multi-output learning tasks. For example, given an image of a *centaur*, it can be described as semantic labels such as *horse* and *person*. It can also be described as attributes such as *head*, *arm*, *tail* and etc. Appropriately modeling the relation between input and output with rich output label interpretation incorporated is an open challenge that can be explored in the future study.

2) Output Heterogeneity: As the demand of sophisticated decision making increasing in real-world applications, they may require outputs with more complex structures, which consist of multiple heterogeneous yet related outputs. For example, people re-identification in surveillance video usually consists of two steps in the traditional approaches, people detection step and then re-identification step given the detected person as input. However, it is demanding to learn these steps together to enhance the performance. Recently, there are several works proposed to simultaneously learn the multiple tasks with different outputs together, such as joint people detection and re-identification [284] and image segmentation with depth estimation at the same time [285]. However, multi-output learning still requires more exploration and investigation for this challenge. For example, given a social network, can we simultaneously learn the representation of a new user as well as its potential links to the existing users?

V. Conclusion

Multi-output learning has attracted significant attention in the last decade. This paper provides a comprehensive review on the study of multi-output learning. We focus on the 4Vs characteristics of the multi-output and the challenges brought by the 4Vs to the learning process of multi-output learning. We first present the life cycle of the output labels and emphasize the issues of each step in the cycle. In addition, we provide an overview on the learning paradigm of multi-output learning, including the myraids of output structures, definition of different sub-problems, model evaluation metrics and popular data repositories used in the last decade. Consequently, we review the representative works, categorized by challenges caused by 4Vs characteristics of multi-outputs.

References

[1] 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, pp. 2098–2105.

[2] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *MT summit*, vol. 5, 2005, pp. 79–86.

[3] Y. Liu, E. P. Xing, and J. G. Carbonell, "Predicting protein folds with structural repeats using a chain graph model," in *ICML*, 2005, pp. 513–520.

[4] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *TKDE*, vol. 26, no. 8, pp. 1819–1837, 2014.

[5] C. Bielza, G. Li, and P. Larrañaga, "Multi-dimensional classification with bayesian networks," *Int. J. Approx. Reasoning*, vol. 52, no. 6, pp. 705–727, 2011.

[6] S. Vembu and T. Gärtner, "Label ranking algorithms: A survey," in *Preference Learning.*, 2010, pp. 45–64.

[7] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[9] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *IJCV*, vol. 77, no. 1-3, pp. 157–173, 2008.

[10] P. Stenetorp, S. Pyysalo, G. Topic, T. Ohta, S. Ananiadou, and J. Tsujii, "brat: a web-based tool for nlp-assisted text annotation," in *EACL*, 2012.

[11] G. Eryigit, F. S. Çetin, M. Yanik, T. Temel, and I. Çiçekli, "TURKSENT: A sentiment annotation tool for social media," in *LAW@ACL*, 2013, pp. 131–134.

[12] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009, pp. 951–958.

[13] M. Rohrbach, M. Stark, and B. Schiele, "Evaluating knowledge transfer and zero-shot learning in a large-scale setting," in *CVPR*, 2011, pp. 1641–1648.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

[15] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *TPAMI*, vol. 35, no. 7, pp. 1757–1772, 2013.

[16] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *NIPS*, 2009, pp. 1410–1418.

[17] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in AI*, vol. 1, no. 1, pp. 89–101, 2012.

[18] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," in *CIKM*, 2004, pp. 78–87.

[19] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010, pp. 3485–3492.

[20] G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, and H. Zhang, "Correlative multi-label video annotation," in *ACM Multimedia*, 2007, pp. 17–26.

[21] S. Dzeroski, D. Demsar, and J. Grbovic, "Predicting chemical parameters of river water quality from bioindicator data," *Appl. Intell.*, vol. 13, no. 1, pp. 7–17, 2000.

[22] H. Aras and N. Aras, "Forecasting residential natural gas demand," *Energy Sources*, vol. 26, no. 5, pp. 463–472, 2004.

[23] H. Li, W. Zhang, Y. Chen, Y. Guo, G.-Z. Li, and X. Zhu, "A novel multi-target regression framework for time-series prediction of drug efficacy," *Scientific reports*, vol. 7, p. 40652, 2017.

[24] X. Geng and Y. Xia, "Head pose estimation based on multivariate label distribution," in *CVPR*, 2014, pp. 1837–1842.

[25] X. Geng, K. Smith-Miles, and Z. Zhou, "Facial age estimation by learning from label distributions," in *AAAI*, 2010.

[26] D. Zhou, X. Zhang, Y. Zhou, Q. Zhao, and X. Geng, "Emotion distribution learning from texts," in *EMNLP*, 2016, pp. 638–647.

[27] K. Crammer and Y. Singer, "A family of additive online algorithms for category ranking," *JMLR*, vol. 3, pp. 1025–1058, 2003.

[28] J. Ko, E. Nyberg, and L. Si, "A probabilistic graphical model for joint answer ranking in question answering," in *SIGIR*, 2007, pp. 343–350.

[29] K. Shaalan, "A survey of arabic named entity recognition and classification," *Computational Linguistics*, vol. 40, no. 2, pp. 469–510, 2014.

[30] A. Newell and J. Deng, "Pixels to graphs by associative embedding," in *NIPS*, 2017, pp. 2168–2177.

[31] B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, "Max-margin parsing," in *EMNLP*, 2004.

[32] D. Liben-Nowell and J. M. Kleinberg, "The link-prediction problem for social networks," *JASIST*, vol. 58, no. 7, pp. 1019–1031, 2007.

[33] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE signal processing magazine*, vol. 20, no. 3, pp. 21–36, 2003.

[34] L. Yang, S. Chou, and Y. Yang, "Midinet: A convolutional generative adversarial network for symbolic-domain music generation," in *ISMIR*, 2017, pp. 324–331.

[35] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. C. Jain, "Content-based image retrieval at the end of the early years," *TPAMI*, vol. 22, no. 12, pp. 1349–1380, 2000.

[36] C. H. Lau, Y. Li, and D. Tjondronegoro, "Microblog retrieval using topical features and query expansion," in *TREC*, 2011.

[37] N. Maria and M. J. Silva, "Theme-based retrieval of web news," in *SIGIR*, 2000, pp. 354–356.

[38] M. K. Choong, M. Charbit, and H. Yan, "Autoregressive-model-based missing value estimation for DNA microarray time series data," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 1, pp. 131–137, 2009.

[39] A. Azadeh, S. Ghaderi, and S. Sohrabkhani, "Annual electricity consumption forecasting by neural network in high energy consuming industrial sectors," *Energy Conversion and management*, vol. 49, no. 8, pp. 2272–2278, 2008.

[40] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models," *TPAMI*, vol. 31, no. 3, pp. 539–555, 2009.

[41] D. Shen, J. Sun, H. Li, Q. Yang, and Z. Chen, "Document summarization using conditional random fields," in *IJCAI*, 2007, pp. 2862–2867.

[42] X. Liang, Z. Hu, H. Zhang, C. Gan, and E. P. Xing, "Recurrent topic-transition GAN for visual paragraph generation," in *ICCV*, 2017, pp. 3382–3391.

[43] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013, pp. 6645–6649.

[44] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *ICML*, 2016, pp. 1060–1069.

[45] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition*, vol. 2014, no. 5, p. 2, 2014.

[46] J. Johnson, R. Krishna, M. Stark, L. Li, D. A. Shamma, M. S. Bernstein, and F. Li, "Image retrieval using scene graphs," in *CVPR*, 2015, pp. 3668–3678.

[47] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *CVPR*, 2018, pp. 1219–1228.

[48] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *IJCV*, vol. 123, no. 1, pp. 32–73, 2017.

[49] D. Zhang, H. Fu, J. Han, and F. Wu, "A review of co-saliency detection technique: Fundamentals, applications, and challenges," *CoRR*, vol. abs/1604.07090, 2016.

[50] A. Joulin, F. R. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *CVPR*, 2010, pp. 1943–1950.

[51] S. Y. Bao, Y. Xiang, and S. Savarese, "Object co-detection," in *ECCV*. Springer, 2012, pp. 86–101.

[52] H. Liu, J. Cai, and Y. Ong, "Remarks on multi-output gaussian process regression," *Knowledge-Based Systems*, vol. 144, pp. 102–121, 2018.

[53] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *TPAMI*, vol. 29, no. 3, pp. 394–410, 2007.

[54] J. J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Seventh ACM Conference on Recommender Systems*, 2013, pp. 165–172.

[55] J. J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *SIGIR*, 2015, pp. 43–52.

[56] J. J. McAuley, R. Pandey, and J. Leskovec, "Inferring networks of substitutable and complementary products," in *KDD*, 2015, pp. 785–794.

[57] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *NIPS*, 2015, pp. 730–738.

[58] A. Zubiaga, "Enhancing navigation on wikipedia with social tags," *CoRR*, vol. abs/1202.5469, 2012.

[59] R. Wetzker, C. Zimmermann, and C. Bauckhage, "Analyzing social bookmarking systems: A del.icio.us cookbook," in *Proceedings of the ECAI 2008 Mining Social Data Workshop*, 2008, pp. 26–30.

[60] Y. Prabhu and M. Varma, "Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning," in *KDD*, 2014, pp. 263–272.

[61] I. Partalas, A. Kosmopoulos, N. Baskiotis, T. Artières, G. Paliouras, É. Gaussier, I. Androutsopoulos, M. Amini, and P. Gallinari, "LSHTC: A benchmark for large-scale text classification," *CoRR*, vol. abs/1503.08581, 2015.

[62] Y. Li, C. Huang, C. C. Loy, and X. Tang, "Human attribute recognition by deep hierarchical contexts," in *ECCV*, 2016, pp. 684–700.

[63] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015, pp. 3730–3738.

[64] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016, pp. 1096–1104.

[65] Q. Chen, J. Huang, R. S. Feris, L. M. Brown, J. Dong, and S. Yan, "Deep domain adaptation for describing people based on fine-grained clothing attributes," in *CVPR*, 2015, pp. 5315–5324.

[66] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth, "Describing objects by their attributes," in *CVPR*, 2009, pp. 1778–1785.

[67] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *CVPR*, 2010, pp. 129–136.

[68] G. Marques, M. A. Domingues, T. Langlois, and F. Gouyon, "Three current issues in music autotagging," in *International Society for Music Information Retrieval Conference*, 2011, pp. 795–800.

[69] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep., 2011.

[70] V. Jouhet, G. Defossez, A. Burgun, P. Le Beux, P. Levillain, P. Ingrand, V. Claveau *et al.*, "Automated classification of free-text pathology reports for registration of incident cases of cancer," *Methods of information in medicine*, vol. 51, no. 3, p. 242, 2012.

[71] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, p. 160035, 2016.

[72] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *JMLR*, vol. 5, pp. 361–397, 2004.

[73] "Kaggle data set ecml/pkdd 15: Taxi trajectory prediction (1)."

[74] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "Crawdad data set epfl/mobility (v. 2009-02-24)," Feb. 2009. [Online]. Available: Downloadedfromhttp://crawdad.org/epfl/mobility/

[75] A. Trindade, “Uci maching learning repository - electricity-loaddiagrams20112014 data set,” 2016. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[76] S. Oh, A. Hoogs, A. G. A. Perera, N. P. Cuntoor, C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. S. Davis, E. Swears, X. Wang, Q. Ji, K. K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. K. Roy-Chowdhury, and M. Desai, “A large-scale benchmark dataset for event recognition in surveillance video,” in *CVPR*, 2011, pp. 3153–3160.

[77] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.

[78] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Citeseer, Tech. Rep., 2009.

[79] G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[80] M. Mahoney, “Large text compression benchmark,” 2011. [Online]. Available: http://www.mattmahoney.net/text/text.html

[81] R. Zafarani and H. Liu, “Social computing data repository at ASU,” 2009. [Online]. Available: http://socialcomputing.asu.edu

[82] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” http://snap.stanford.edu/data, Jun. 2014.

[83] E. F. T. K. Sang and S. Buchholz, “Introduction to the conll-2000 shared task chunking,” in *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*, 2000, pp. 127–132.

[84] D. Zhou, J. C. Platt, S. Basu, and Y. Mao, “Learning from the wisdom of crowds by minimax entropy,” in *Conference on Neural Information Processing Systems*, 2012, pp. 2204–2212.

[85] K. Lee, X. He, L. Zhang, and L. Yang, “Cleannet: Transfer learning for scalable image classifier training with label noise,” in *CVPR*, 2018, pp. 5447–5456.

[86] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification,” in *CVPR*, 2015, pp. 2691–2699.

[87] W. Li, L. Wang, W. Li, E. Agustsson, and L. V. Gool, “Webvision database: Visual learning and understanding from web data,” *CoRR*, vol. abs/1708.02862, 2017.

[88] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li, “The new data and new challenges in multimedia research,” *CoRR*, vol. abs/1503.01817, 2015.

[89] H. Brighton and C. Mellish, “Advances in instance selection for instance-based learning algorithms,” *Data mining and knowledge discovery*, vol. 6, no. 2, pp. 153–172, 2002.

[90] Y. Zhai, Y. Ong, and I. W. Tsang, “The emerging ?big dimensionality?” *IEEE Computational Intelligence Magazine*, vol. 9, no. 3, pp. 14–26, 2014.

[91] C. Kümmerle and J. Sigl, “Harmonic mean iteratively reweighted least squares for low-rank matrix recovery,” *JMLR*, vol. 19, 2018.

[92] W. Liu and I. W. Tsang, “Making decision trees feasible in ultrahigh feature and label dimensions,” *JMLR*, vol. 18, pp. 81:1–81:36, 2017.

[93] C. Dupuy and F. Bach, “Online but accurate inference for latent variable models with local gibbs sampling,” *JMLR*, vol. 18, pp. 126:1–126:45, 2017.

[94] W. Liu, I. W. Tsang, and K. Müller, “An easy-to-hard learning paradigm for multiple classes and multiple labels,” *JMLR*, vol. 18, pp. 94:1–94:38, 2017.

[95] C. Brouard, M. Szafranski, and F. d'Alché-Buc, “Input output kernel regression: Supervised and semi-supervised structured output prediction with operator-valued kernels,” *JMLR*, vol. 17, pp. 176:1–176:48, 2016.

[96] H. Shin, L. Lu, L. Kim, A. Seff, J. Yao, and R. M. Summers, “Interleaved text/image deep mining on a large-scale radiology database for automated image interpretation,” *JMLR*, vol. 17, pp. 107:1–107:31, 2016.

[97] R. Babbar, I. Partalas, É. Gaussier, M. Amini, and C. Amblard, “Learning taxonomy adaptation in large-scale classification,” *JMLR*, vol. 17, pp. 98:1–98:37, 2016.

[98] X. Li, T. Zhao, X. Yuan, and H. Liu, “The flare package for high dimensional linear regression and precision matrix estimation in R,” *JMLR*, vol. 16, pp. 553–557, 2015.

[99] F. Han, H. Lu, and H. Liu, “A direct estimation of high dimensional stationary vector autoregressions,” *JMLR*, vol. 16, pp. 3115–3150, 2015.

[100] J. R. Doppa, A. Fern, and P. Tadepalli, “Structured prediction via output space search,” *JMLR*, vol. 15, no. 1, pp. 1317–1350, 2014.

[101] D. Colombo and M. H. Maathuis, “Order-independent constraint-based causal structure learning,” *JMLR*, vol. 15, no. 1, pp. 3741–3782, 2014.

[102] C. Gentile and F. Orabona, “On multilabel classification and ranking with bandit feedback,” *JMLR*, vol. 15, no. 1, pp. 2451–2487, 2014.

[103] P. Gong, J. Ye, and C. Zhang, “Multi-stage multi-task feature learning,” *JMLR*, vol. 14, no. 1, pp. 2979–3010, 2013.

[104] A. Talwalkar, S. Kumar, M. Mohri, and H. A. Rowley, “Large-scale SVD and manifold learning,” *JMLR*, vol. 14, no. 1, pp. 3129–3152, 2013.

[105] K. Fu, J. Li, J. Jin, and C. Zhang, “Image-text surgery: Efficient concept learning in image captioning by generating pseudopairs,” *TNNLS*, vol. 29, no. 12, pp. 5910–5921, 2018.

[106] . Protas, J. D. Bratti, J. F. O. Gaya, P. Drews, and S. S. C. Botelho, “Visualization methods for image transformation convolutional neural networks,” *TNNLS*, 2018.

[107] H. Zhang, S. Wang, X. Xu, T. W. S. Chow, and Q. M. J. Wu, “Tree2vector: Learning a vectorial representation for tree-structured data,” *TNNLS*, vol. 29, no. 11, pp. 5304–5318, 2018.

[108] Z. Lin, G. Ding, J. Han, and L. Shao, “End-to-end feature-aware label space encoding for multilabel classification with many classes,” *TNNLS*, vol. 29, no. 6, pp. 2472–2487, 2018.

[109] S. Fang, J. Li, Y. Tian, T. Huang, and X. Chen, “Learning discriminative subspaces on random contrasts for image saliency analysis,” *TNNLS*, vol. 28, no. 5, pp. 1095–1108, 2017.

[110] K. Zhang, D. Tao, X. Gao, X. Li, and J. Li, “Coarse-to-fine learning for single-image super-resolution,” *TNNLS*, vol. 28, no. 5, pp. 1109–1122, 2017.

[111] M. Kim, “Mixtures of conditional random fields for improved structured output prediction,” *TNNLS*, vol. 28, no. 5, pp. 1233–1240, 2017.

[112] Y. Cheung, M. Li, Q. Peng, and C. L. P. Chen, “A cooperative learning-based clustering approach to lip segmentation without knowing segment number,” *TNNLS*, vol. 28, no. 1, pp. 80–93, 2017.

[113] L. Wang, L. Liu, and L. Zhou, “A graph-embedding approach to hierarchical visual word mergence,” *TNNLS*, vol. 28, no. 2, pp. 308–320, 2017.

[114] Z. Li, Z. Lai, Y. Xu, J. Yang, and D. Zhang, “A locality-constrained and label embedding dictionary learning algorithm for image classification,” *TNNLS*, vol. 28, no. 2, pp. 278–293, 2017.

[115] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q. Zhu, “A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method,” *TNNLS*, vol. 27, no. 3, pp. 579–592, 2016.

[116] C. Deng, J. Xu, K. Zhang, D. Tao, X. Gao, and X. Li, “Similarity constraints-based structured output regression machine: An approach to image super-resolution,” *TNNLS*, vol. 27, no. 12, pp. 2472–2485, 2016.

[117] A. Alush and J. Goldberger, “Hierarchical image segmentation using correlation clustering,” *TNNLS*, vol. 27, no. 6, pp. 1358–1367, 2016.

[118] D. Tao, J. Cheng, M. Song, and X. Lin, “Manifold ranking-based matrix factorization for saliency detection,” *TNNLS*, vol. 27, no. 6, pp. 1122–1134, 2016.

[119] F. Cao, M. Cai, Y. Tan, and J. Zhao, “Image super-resolution via adaptive $_p$ $(0<p<1)$ regularization and sparse representation,” *TNNLS*, vol. 27, no. 7, pp. 1550–1561, 2016.

[120] Q. Zhu, L. Shao, X. Li, and L. Wang, “Targeting accurate object extraction from an image: A comprehensive study of natural image matting,” *TNNLS*, vol. 26, no. 2, pp. 185–207, 2015.

[121] Y. Chen, Y. Ma, D. H. Kim, and S. Park, “Region-based object recognition by color segmentation using a simplified PCNN,” *TNNLS*, vol. 26, no. 8, pp. 1682–1697, 2015.

[122] M. Li, W. Bi, J. T. Kwok, and B. Lu, “Large-scale nyström kernel matrix approximation using randomized SVD,” *TNNLS*, vol. 26, no. 1, pp. 152–164, 2015.

[123] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, “A unified learning framework for single image super-resolution,” *TNNLS*, vol. 25, no. 4, pp. 780–792, 2014.

[124] A. Bauer, N. Görnitz, F. Biegler, K. Müller, and M. Kloft, “Efficient algorithms for exact inference in sequence labeling svms,” *TNNLS*, vol. 25, no. 5, pp. 870–881, 2014.

[125] K. Tang, R. Liu, Z. Su, and J. Zhang, “Structure-constrained low-rank representation,” *TNNLS*, vol. 25, no. 12, pp. 2167–2179, 2014.

[126] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, “Low-rank structure learning via nonconvex heuristic recovery,” *TNNLS*, vol. 24, no. 3, pp. 383–396, 2013.

[127] H. Zhang, Q. M. J. Wu, and T. M. Nguyen, “Incorporating mean template into finite mixture model for image segmentation,” *TNNLS*, vol. 24, no. 2, pp. 328–335, 2013.

[128] Y. Pang, Z. Ji, P. Jing, and X. Li, "Ranking graph embedding for learning to rerank," *TNNLS*, vol. 24, no. 8, pp. 1292–1303, 2013.

[129] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen, "Multiview vector-valued manifold regularization for multilabel image classification," *TNNLS*, vol. 24, no. 5, pp. 709–722, 2013.

[130] B. Zhang, D. Xiong, and J. Su, "Neural machine translation with deep attention," *TPAMI*, 2018.

[131] S. Jeong, J. Lee, B. Kim, Y. Kim, and J. Noh, "Object segmentation ensuring consistency across multi-viewpoint images," *TPAMI*, vol. 40, no. 10, pp. 2455–2468, 2018.

[132] C. Raposo, M. Antunes, and J. P. Barreto, "Piecewise-planar stereoscan: Sequential structure and motion using plane primitives," *TPAMI*, vol. 40, no. 8, pp. 1918–1931, 2018.

[133] M. Cordts, T. Rehfeld, M. Enzweiler, U. Franke, and S. Roth, "Tree-structured models for efficient multi-cue scene labeling," *TPAMI*, vol. 39, no. 7, pp. 1444–1454, 2017.

[134] Y. Xu, E. Carlinet, T. Géraud, and L. Najman, "Hierarchical segmentation using tree-based shape spaces," *TPAMI*, vol. 39, no. 3, pp. 457–469, 2017.

[135] K. Fu, J. Jin, R. Cui, F. Sha, and C. Zhang, "Aligning where to see and what to tell: Image captioning with region-based attention and scene-specific contexts," *TPAMI*, vol. 39, no. 12, pp. 2321–2334, 2017.

[136] M. A. Hasnat, O. Alata, and A. Trémeau, "Joint color-spatial-directional clustering and region merging (JCSD-RM) for unsupervised RGB-D image segmentation," *TPAMI*, vol. 38, no. 11, pp. 2255–2268, 2016.

[137] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *TPAMI*, vol. 38, no. 1, pp. 142–158, 2016.

[138] Z. Qin and C. R. Shelton, "Social grouping for multi-target tracking and head pose estimation in video," *TPAMI*, vol. 38, no. 10, pp. 2082–2095, 2016.

[139] Y. Kwon, K. I. Kim, J. Tompkin, J. H. Kim, and C. Theobalt, "Efficient learning of image super-resolution and compression artifact removal with semi-local gaussian processes," *TPAMI*, vol. 37, no. 9, pp. 1792–1805, 2015.

[140] A. Djelouah, J. Franco, E. Boyer, F. L. Clerc, and P. Pérez, "Sparse multi-view consistency for object segmentation," *TPAMI*, vol. 37, no. 9, pp. 1890–1903, 2015.

[141] S. Wang, Y. Wei, K. Long, X. Zeng, and M. Zheng, "Image super-resolution via self-similarity learning and conformal sparse representation," *TPAMI*.

[142] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *TPAMI*, vol. 36, no. 8, pp. 1614–1627, 2014.

[143] N. Zhou and J. Fan, "Jointly learning visually correlated dictionaries for large-scale visual recognition applications," *TPAMI*, vol. 36, no. 4, pp. 715–730, 2014.

[144] P. Perakis, G. Passalis, T. Theoharis, and I. A. Kakadiaris, "3d facial landmark detection under large yaw and expression variations," *TPAMI*, vol. 35, no. 7, pp. 1552–1564, 2013.

[145] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *TPAMI*, vol. 35, no. 12, pp. 2916–2929, 2013.

[146] K. G. Dizaji, X. Wang, and H. Huang, "Semi-supervised generative adversarial network for gene expression inference," in *KDD*, 2018, pp. 1435–1444.

[147] M. Lee, B. Gao, and R. Zhang, "Rare query expansion through generative adversarial networks in search advertising," in *KDD*, 2018, pp. 500–508.

[148] I. E. Yen, X. Huang, W. Dai, P. Ravikumar, I. S. Dhillon, and E. P. Xing, "Ppdsparse: A parallel primal-dual sparse method for extreme classification," in *KDD*, 2017, pp. 545–553.

[149] Y. Tagami, "Annexml: Approximate nearest neighbor search for extreme multi-label classification," in *KDD*, 2017, pp. 455–464.

[150] H. Jain, Y. Prabhu, and M. Varma, "Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications," in *KDD*, 2016, pp. 935–944.

[151] C. Xu, D. Tao, and C. Xu, "Robust extreme multi-label learning," in *KDD*, 2016, pp. 1275–1284.

[152] C. Kuo, X. Wang, P. B. Walker, O. T. Carmichael, J. Ye, and I. Davidson, "Unified and contrasting cuts in multiple graphs: Application to medical imaging segmentation," in *KDD*, 2015, pp. 617–626.

[153] C. Papagiannopoulou, G. Tsoumakas, and I. Tsamardinos, "Discovering and exploiting deterministic label relationships in multi-label learning," in *KDD*, 2015, pp. 915–924.

[154] B. Wu, E. Zhong, B. Tan, A. Horner, and Q. Yang, "Crowdsourced time-sync video tagging using temporal and personalized topic modeling," in *KDD*, 2014, pp. 721–730.

[155] S. Zhai, T. Xia, and S. Wang, "A multi-class boosting method with direct optimization," in *KDD*, 2014, pp. 273–282.

[156] X. Kong, B. Cao, and P. S. Yu, "Multi-label classification by mining label and instance correlations from heterogeneous information networks," in *KDD*, 2013, pp. 614–622.

[157] X. Wang and G. Sukthankar, "Multi-label relational neighbor classification using social context features," in *KDD*, 2013, pp. 464–472.

[158] S. Hong, X. Yan, T. S. Huang, and H. Lee, "Learning hierarchical semantic image manipulation through structured representations," in *NIPS*, 2018, pp. 2713–2723.

[159] M. Wydmuch, K. Jasinska, M. Kuznetsov, R. Busa-Fekete, and K. Dembczynski, "A no-regret generalization of hierarchical softmax to extreme multi-label classification," in *NIPS*, 2018, pp. 6358–6368.

[160] B. Pan, Y. Yang, H. Li, Z. Zhao, Y. Zhuang, D. Cai, and X. He, "Macnet: Transferring knowledge from machine comprehension to sequence-to-sequence models," in *NIPS*, 2018, pp. 6095–6105.

[161] E. Racah, C. Beckham, T. Maharaj, S. E. Kahou, Prabhat, and C. Pal, "Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events," in *NIPS*, 2017, pp. 3405–3416.

[162] Y. Hu, J. Huang, and A. G. Schwing, "Maskrnn: Instance level video object segmentation," in *NIPS*, 2017, pp. 324–333.

[163] B. Joshi, M. Amini, I. Partalas, F. Iutzeler, and Y. Maximov, "Aggressive sampling for multi-class to binary reduction with applications to text classification," in *NIPS*, 2017, pp. 4162–4171.

[164] J. Nam, E. Loza Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *NIPS*, 2017, pp. 5419–5429.

[165] N. Rosenfeld and A. Globerson, "Optimal tagging with markov chain optimization," in *NIPS*, 2016, pp. 1307–1315.

[166] H. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *NIPS*, 2016, pp. 847–855.

[167] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *NIPS*, 2015, pp. 1171–1179.

[168] P. Rai, C. Hu, R. Henao, and L. Carin, "Large-scale bayesian multi-label learning via topic-based label embeddings," in *NIPS*, 2015, pp. 3222–3230.

[169] A. Wu, M. Park, O. Koyejo, and J. W. Pillow, "Sparse bayesian structure learning with dependent relevance determination priors," in *NIPS*, 2014, pp. 1628–1636.

[170] V. Nguyen, J. L. Boyd-Graber, P. Resnik, and J. Chang, "Learning a concept hierarchy from multi-labeled documents," in *NIPS*, 2014, pp. 3671–3679.

[171] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. B. Girshick, T. Darrell, and K. Saenko, "LSDA: large scale detection through adaptation," in *NIPS*, 2014, pp. 3536–3544.

[172] M. Xu, R. Jin, and Z. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *NIPS*, 2013, pp. 2301–2309.

[173] M. Cissé, N. Usunier, T. Artières, and P. Gallinari, "Robust bloom filters for large multilabel classification tasks," in *NIPS*, 2013, pp. 1851–1859.

[174] W. Siblini, F. Meyer, and P. Kuntz, "Craftml, an efficient clustering-based random forest for extreme multi-label learning," in *ICML*, 2018, pp. 4671–4680.

[175] I. E. Yen, S. Kale, F. X. Yu, D. N. Holtmann-Rice, S. Kumar, and P. Ravikumar, "Loss decomposition for fast learning in large output spaces," in *ICML*, 2018, pp. 5626–5635.

[176] J. Wehrmann, R. Cerri, and R. C. Barros, "Hierarchical multi-label classification networks," in *ICML*, 2018, pp. 5225–5234.

[177] S. Si, H. Zhang, S. S. Keerthi, D. Mahajan, I. S. Dhillon, and C. Hsieh, "Gradient boosted decision trees for high dimensional sparse output," in *ICML*, 2017, pp. 3182–3190.

[178] V. Jain, N. Modhe, and P. Rai, "Scalable generative models for multi-label learning with missing labels," in *ICML*, 2017, pp. 1636–1644.

[179] T. Zhang and Z. Zhou, "Multi-class optimal margin distribution machine," in *ICML*, 2017, pp. 4063–4071.

[180] C. Li, B. Wang, V. Pavlu, and J. A. Aslam, "Conditional bernoulli mixtures for multi-label classification," in *ICML*, 2016, pp. 2482–2491.

[181] I. E. Yen, X. Huang, P. Ravikumar, K. Zhong, and I. S. Dhillon, "Pd-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification," in *ICML*, 2016, pp. 3069–3077.

[182] M. Cissé, M. Al-Shedivat, and S. Bengio, "ADIOS: architectures deep in output space," in *ICML*, 2016, pp. 2770–2779.
[183] D. Park, J. Neeman, J. Zhang, S. Sanghavi, and I. S. Dhillon, "Preference completion: Large-scale collaborative ranking from pairwise comparisons," in *ICML*, 2015, pp. 1907–1916.
[184] D. Hernández-Lobato, J. M. Hernández-Lobato, and Z. Ghahramani, "A probabilistic model for dirty multi-task feature selection," in *ICML*, 2015, pp. 1073–1082.
[185] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *ICML*, 2015, pp. 720–729.
[186] H. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale multi-label learning with missing labels," in *ICML*, 2014, pp. 593–601.
[187] Z. Lin, G. Ding, M. Hu, and J. Wang, "Multi-label classification via feature-aware implicit label space encoding," in *ICML*, 2014, pp. 325–333.
[188] Y. Li and R. S. Zemel, "High order regularization for semi-supervised learning of structured output problems," in *ICML*, 2014, pp. 1368–1376.
[189] W. Bi and J. T. Kwok, "Efficient multi-label classification with many labels," in *ICML*, 2013, pp. 405–413.
[190] R. Takhanov and V. Kolmogorov, "Inference algorithms for pattern-based crfs on sequence data," in *ICML*, 2013, pp. 145–153.
[191] M. Xiao and Y. Guo, "Domain adaptation for sequence labeling tasks with a probabilistic language adaptation model," in *ICML*, 2013, pp. 293–301.
[192] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *NIPS*, 2016, pp. 217–225.
[193] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, pp. 2672–2680.
[194] H. Zhang, T. Xu, and H. Li, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *ICCV*, 2017, pp. 5908–5916.
[195] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," *CoRR*, vol. abs/1802.09178, 2018.
[196] W. Fedus, I. J. Goodfellow, and A. M. Dai, "Maskgan: Better text generation via filling in the ______," *CoRR*, vol. abs/1801.07736, 2018.
[197] Y. Chen and H. Lin, "Feature-aware label space dimension reduction for multi-label classification," in *NIPS*, 2012, pp. 1538–1546.
[198] D. J. Hsu, S. Kakade, J. Langford, and T. Zhang, "Multi-label prediction via compressed sensing," in *NIPS*, 2009, pp. 772–780.
[199] F. Tai and H. Lin, "Multilabel classification with principal label space transformation," *Neural Computation*, vol. 24, no. 9, pp. 2508–2542, 2012.
[200] A. Kapoor, R. Viswanathan, and P. Jain, "Multilabel classification using bayesian compressed sensing," in *NIPS*, 2012, pp. 2654–2662.
[201] P. Mineiro and N. Karampatziakis, "Fast label embeddings via randomized linear algebra," in *Joint European conference on machine learning and knowledge discovery in databases*, 2015, pp. 37–51.
[202] Y. Jernite, A. Choromanska, and D. Sontag, "Simultaneous learning of trees and representations for extreme classification and density estimation," in *ICML*, 2017, pp. 1665–1674.
[203] J. Liu, W. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *SIGIR*, 2017, pp. 115–124.
[204] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.
[205] O. Maimon and L. Rokach, Eds., *Data Mining and Knowledge Discovery Handbook, 2nd ed*. Springer, 2010.
[206] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *ECML PKDD*, 2009, pp. 254–269.
[207] ——, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, no. 3, pp. 333–359, 2011.
[208] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *JMLR*, vol. 6, pp. 1453–1484, 2005.
[209] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001, pp. 282–289.
[210] G. Tsoumakas and I. P. Vlahavas, "Random k -labelsets: An ensemble method for multilabel classification," in *ECML*, 2007, pp. 406–417.
[211] K. Dembczynski, W. Cheng, and E. Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," in *ICML*, 2010, pp. 279–286.
[212] T. Joachims, T. Finley, and C. J. Yu, "Cutting-plane training of structural svms," *Machine Learning*, vol. 77, no. 1, pp. 27–59, 2009.
[213] S. Baker and A. Korhonen, "Initializing neural networks for hierarchical multi-label text classification," in *BioNLP*, 2017, pp. 307–315.
[214] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, "Using ranking-cnn for age estimation," in *CVPR*, 2017, pp. 742–751.
[215] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 3104–3112.
[216] C. Smith and Y. Jin, "Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction," *Neurocomputing*, vol. 143, pp. 302–311, 2014.
[217] F. J. Och, "Minimum error rate training in statistical machine translation," in *Association for Computational Linguistics*, 2003, pp. 160–167.
[218] A. Tewari and P. L. Bartlett, "On the consistency of multiclass classification methods," *JMLR*, vol. 8, pp. 1007–1025, 2007.
[219] W. Gao and Z.-H. Zhou, "On the consistency of multi-label learning," in *Proceedings of the 24th annual conference on learning theory*, 2011, pp. 341–358.
[220] D. A. McAllester and J. Keshet, "Generalization bounds and consistency for latent structural probit and ramp loss," in *NIPS*, 2011, pp. 2205–2212.
[221] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *NIPS*, 2003, pp. 25–32.
[222] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *SIGIR*, 2007, pp. 271–278.
[223] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Empirical Methods in Natural Language Processing*, 2002.
[224] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *ICASSP*, 2008, pp. 4057–4060.
[225] K. Gimpel and N. A. Smith, "Softmax-margin crfs: Training log-linear models with cost functions," in *HLT-NAACL*, 2010, pp. 733–736.
[226] W. Liu, D. Xu, I. Tsang, and W. Zhang, "Metric learning for multi-output tasks," *TPAMI*, 2018, doi:10.1109/TPAMI.2018.2794976.
[227] J. Deng, S. Satheesh, A. C. Berg, and F. Li, "Fast and balanced: Efficient label tree learning for large scale object recognition," in *NIPS*, 2011, pp. 567–575.
[228] T. Gao and D. Koller, "Discriminative learning of relaxed hierarchy for large-scale visual recognition," in *ICCV*, 2011, pp. 2072–2079.
[229] X. Shen, W. Liu, I. W. Tsang, Q. Sun, and Y. Ong, "Compact multi-label learning," in *AAAI*, 2018, pp. 4066–4073.
[230] ——, "Multilabel prediction via cross-view search," *TNNLS*, vol. 29, no. 9, pp. 4324–4338, 2018.
[231] X. Shen, W. Liu, Y. Luo, Y. Ong, and I. W. Tsang, "Deep discrete prototype multilabel learning," in *IJCAI*, 2018, pp. 2675–2681.
[232] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *JAIR*, vol. 16, pp. 321–357, 2002.
[233] J. Jang, T.-j. Eo, M. Kim, N. Choi, D. Han, D. Kim, and D. Hwang, "Medical image matching using variable randomized undersampling probability pattern in data acquisition," in *International Conference on Electronics, Information and Communications*. IEEE, 2014, pp. 1–2.
[234] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *CoRR*, vol. abs/1804.10851, 2018.
[235] M. Rezaei, H. Yang, and C. Meinel, "Multi-task generative adversarial network for handling imbalanced clinical data," *CoRR*, vol. abs/1811.10419, 2018.
[236] E. Montahaei, M. Ghorbani, M. S. Baghshah, and H. R. Rabiee, "Adversarial classifier for imbalanced problems," *CoRR*, vol. abs/1811.08812, 2018.
[237] B. Romera-Paredes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," in *ICML*, 2015, pp. 2152–2161.
[238] A. Gaure, A. Gupta, V. K. Verma, and P. Rai, "A probabilistic framework for zero-shot multi-label learning," in *The Conference on Uncertainty in Artificial Intelligence (UAI)*, vol. 1, 2017, p. 3.
[239] A. Rios and R. Kavuluru, "Few-shot and zero-shot multi-label learning for structured label spaces," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3132–3142.
[240] C. Lee, W. Fang, C. Yeh, and Y. F. Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *CVPR*, 2018, pp. 1576–1585.

[241] C. Gan, T. Yang, and B. Gong, “Learning attributes equals multi-source domain generalization,” in *CVPR*, 2016, pp. 87–97.

[242] J. Liu, B. Kuipers, and S. Savarese, “Recognizing human actions by attributes,” in *CVPR*, 2011, pp. 3337–3344.

[243] Z. Zhang, C. Wang, B. Xiao, W. Zhou, and S. Liu, “Robust relative attributes for human action recognition,” *Pattern Analysis and Applications*, vol. 18, no. 1, pp. 157–171, 2015.

[244] M. Jain, J. C. van Gemert, T. Mensink, and C. G. M. Snoek, “Objects2action: Classifying and localizing actions without any video example,” in *ICCV*, 2015, pp. 4588–4596.

[245] P. Mettes and C. G. M. Snoek, “Spatial-aware object embeddings for zero-shot localization and classification of actions,” in *ICCV*, 2017, pp. 4453–4462.

[246] W. J. Scheirer, L. P. Jain, and T. E. Boult, “Probability models for open set recognition,” *TPAMI*, vol. 36, no. 11, pp. 2317–2324, 2014.

[247] L. P. Jain, W. J. Scheirer, and T. E. Boult, “Multi-class open set recognition using probability of inclusion,” in *ECCV*, 2014, pp. 393–409.

[248] A. Bendale and T. E. Boult, “Towards open set deep networks,” in *CVPR*, 2016, pp. 1563–1572.

[249] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L. Li, “Learning from noisy labels with distillation,” in *ICCV*, 2017, pp. 1928–1936.

[250] G. Chen, Y. Song, F. Wang, and C. Zhang, “Semi-supervised multi-label learning by solving a sylvester equation,” in *ICDM*, 2008, pp. 410–419.

[251] Y. Sun, Y. Zhang, and Z. Zhou, “Multi-label learning with weak label,” in *AAAI*, 2010.

[252] S. S. Bucak, R. Jin, and A. K. Jain, “Multi-label learning with incomplete class assignments,” in *CVPR*, 2011, pp. 2801–2808.

[253] R. S. Cabral, F. D. la Torre, J. P. Costeira, and A. Bernardino, “Matrix completion for multi-label image classification,” in *NIPS*, 2011, pp. 190–198.

[254] W. Bi and J. T. Kwok, “Multilabel classification with label correlations and missing labels,” in *AAAI*, 2014, pp. 1680–1686.

[255] H. Yang, J. T. Zhou, and J. Cai, “Improving multi-label learning with missing labels by structured semantic correlations,” in *ECCV*, 2016, pp. 835–851.

[256] R. Barandela and E. Gasca, “Decontamination of training samples for supervised pattern recognition methods,” in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2000, pp. 621–630.

[257] C. E. Brodley and M. A. Friedl, “Identifying mislabeled training data,” *JAIR*, vol. 11, pp. 131–167, 1999.

[258] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache, “Learning visual features from large weakly supervised data,” in *ECCV*, 2016, pp. 67–84.

[259] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, “Training deep neural networks on noisy labels with bootstrapping,” *CoRR*, vol. abs/1412.6596, 2014.

[260] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. J. Belongie, “Learning from noisy large-scale datasets with minimal supervision,” in *CVPR*, 2017, pp. 6575–6583.

[261] V. Mnih and G. E. Hinton, “Learning to label aerial images from noisy data,” in *ICML*, 2012.

[262] I. Jindal, M. S. Nokleby, and X. Chen, “Learning deep networks from noisy labels with dropout regularization,” in *ICDM*, 2016, pp. 967–972.

[263] J. Yao, J. Wang, I. W. Tsang, Y. Zhang, J. Sun, C. Zhang, and R. Zhang, “Deep learning from noisy image labels with quality embedding,” *CoRR*, vol. abs/1711.00583, 2017.

[264] Y. Mao, G. Cheung, C. Lin, and Y. Ji, “Joint learning of similarity graph and image classifier from partial labels,” in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2016, pp. 1–4.

[265] F. Yu and M. Zhang, “Maximum margin partial label learning,” *Machine Learning*, vol. 106, no. 4, pp. 573–593, 2017.

[266] M. Zhang, F. Yu, and C. Tang, “Disambiguation-free partial label learning,” *TKDE*, vol. 29, no. 10, pp. 2155–2167, 2017.

[267] M. Xie and S. Huang, “Partial multi-label learning,” in *AAAI*, 2018, pp. 4302–4309.

[268] H. Nam and B. Han, “Learning multi-domain convolutional neural networks for visual tracking,” in *CVPR*, 2016, pp. 4293–4302.

[269] S. Avidan, “Ensemble tracking,” *TPAMI*, vol. 29, no. 2, 2007.

[270] W. Qu, Y. Zhang, J. Zhu, and Q. Qiu, “Mining multi-label concept-drifting data streams using dynamic classifier ensemble,” in *ACML*, 2009, pp. 308–321.

[271] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, “New ensemble methods for evolving data streams,” in *KDD*, 2009, pp. 139–148.

[272] X. Kong and P. S. Yu, “An ensemble-based approach to fast classification of multi-label data streams,” in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 2011, pp. 95–104.

[273] A. Büyükçakir, H. R. Bonab, and F. Can, “A novel online stacked ensemble for multi-label stream classification,” in *CIKM*, 2018, pp. 1063–1072.

[274] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler, “Online multi-target tracking using recurrent neural networks,” in *AAAI*, 2017, pp. 4225–4232.

[275] J. Read, A. Bifet, G. Holmes, and B. Pfahringer, “Scalable and efficient multi-label classification for evolving data streams,” *Machine Learning*, vol. 88, no. 1-2, pp. 243–272, 2012.

[276] L. Huang, Q. Yang, and W. Zheng, “Online hashing,” *TNNLS*, vol. 29, no. 6, pp. 2309–2322, 2018.

[277] D. Xu, I. W. Tsang, and Y. Zhang, “Online product quantization,” *TKDE*, vol. 30, no. 11, pp. 2185 – 2198, 2018.

[278] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S. Xia, “Iterative learning with open-set noisy labels,” in *Computer Vision and Pattern Recognition*, 2018, pp. 8688–8696.

[279] O. Anava, E. Hazan, and A. Zeevi, “Online time series prediction with missing data,” in *ICML*, 2015, pp. 2191–2199.

[280] A. Bendale and T. E. Boult, “Towards open world recognition,” in *CVPR*, 2015, pp. 1893–1902.

[281] E. S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. P. Vlahavas, “Dealing with concept drift and class imbalance in multi-label stream classification,” in *IJCAI*, 2011, pp. 1583–1588.

[282] Z. Ren, M. Peetz, S. Liang, W. van Dolen, and M. de Rijke, “Hierarchical multi-label classification of social text streams,” in *SIGIR*, 2014, pp. 213–222.

[283] Y. Shi, D. Xu, Y. Pan, and I. W. Tsang, “Multi-context label embedding,” *CoRR*, vol. abs/1805.01199, 2018.

[284] A. Mousavian, H. Pirsiavash, and J. Kosecka, “Joint semantic segmentation and depth estimation with deep convolutional networks,” in *3DV*, 2016, pp. 611–619.

[285] W. Van Ranst, F. De Smedt, J. Berte, T. Goedemé, and T.-Z. RoboVision, “Fast simultaneous people detection and re-identification in a single shot network,” in *AVSS*, 2018.