Survey on Multi-Output Learning

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

***Abstract*— The aim of multi-output learning is to simultaneously predict multiple outputs given an input. It is an important learning problem for decision-making since making decisions in the real world often involves multiple complex factors and criteria. In recent times, an increasing number of research studies have focused on ways to predict multiple outputs at once. Such efforts have transpired in different forms according to the particular multi-output learning problem under study. Classic cases of multi-output learning include multi-label learning, multi-dimensional learning, multi-target regression, and others. From our survey of the topic, we were struck by a lack in studies that generalize the different forms of multi-output learning into a common framework. This article fills that gap with a comprehensive review and analysis of the multi-output learning paradigm. In particular, we characterize the four Vs of multi-output learning, i.e., volume, velocity, variety, and veracity, and the ways in which the four Vs both benefit and bring challenges to multi-output learning by taking inspiration from big data. We analyze the life cycle of output labeling, present the main mathematical definitions of multi-output learning, and examine the field's key challenges and corresponding solutions as found in the literature. Several model evaluation metrics and popular data repositories are also discussed. Last but not least, we highlight some emerging challenges with multi-output learning from the perspective of the four Vs as potential research directions worthy of further studies.**

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

Manuscript received December 31, 2018; revised August 6, 2019; accepted September 24, 2019. Date of publication November 6, 2019; date of current version July 7, 2020. This work was supported in part by the ARC under Grant LP150100671 and Grant DP180100106, in part by the CSC under Grant 201706330075, in part by the NRFS through its AI Singapore Program under Grant AISG-RP-2018-004, in part by the NSF of China under Grant 61602246 and Grant 61973162, in part by the NSF of Jiangsu Province under Grant BK20171430, in part by the FRF for the Central Universities under Grant 30918011319, in part by the Summit of the Six Top Talents Program under Grant DZXX-027, in part by the Young Elite Scientists Sponsorship Program by Jiangsu Province, in part by the Young Elite Scientists Sponsorship Program by CAST under Grant 2018QNRC001, in part by the NNSFC under Grant 61906091, in part by the NSF of Jiangsu Province, China, through the Youth Fund Project under Grant BK20190440, and in part by the FRF for the Central Universities under Grant 30919011229. *(Corresponding author: Donna Xu.)*

D. Xu, Y. Shi, and I. W. Tsang are with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, 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 Data Science & Artificial Intelligence Research Centre, 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 Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chen.gong@njust.edu.cn; njust.shenxiaobo@gmail.com).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2945133

I. Introduction

TRADITIONAL supervised learning is one of the most well established and adopted machine learning paradigms. It offers fast and accurate predictions for today's real-world smart systems and applications. The goal of traditional supervised learning is to learn a function that maps each of the given inputs to a corresponding known output. For prediction tasks, the output comes in the form of a single label. For regression tasks, it is a single value. Traditional supervised learning has been shown to be good at solving these simple single-output problems, classical examples being binary classification, such as filtering spam in an email system, or a regression problem where the daily energy consumption of a machine needs to be predicted based on temperature, wind speed, humidity levels, and so on.

However, the traditional supervised learning paradigm is not coping well with the increasing needs of today's complex decision making. As a result, there is a pressing need for new machine learning paradigms. Here, multi-output learning has emerged as a solution. The aim is to simultaneously predict multiple outputs given a single input, which means it is possible to solve far more complex decision-making problems. Compared to traditional single-output learning, multi-output learning is multi-variate nature, and the outputs may have complex interactions that can only be handled by structured inference. In addition, the potentially diverse data types of outputs have led to various categories of machine learning problems and corresponding subfields of study. For example, binary output values relate to multi-label classification problems [1], [2]; nominal output values relate to multi-dimensional classification problems [3]; ordinal output values are studied in label ranking problems [4]; and real-valued outputs are considered in multi-target regression problems [5].

Together, all these problems constitute the multi-output paradigm, and the body of literature surrounding this field has grown rapidly. Several works have been presented that provide a comprehensive review of the emerging challenges and learning algorithms in each subfield. For instance, Zhang and Zhou [1] studied the emerging area of multi-label learning; Borchani *et al.* [5] summarized the increasing problems in multi-target regression; Vembu and Gärtner [4] presented a review on multi-label ranking. However, little attention has been paid to the global picture of multi-output learning and the importance of the output labels (see Section II). In addition, although the problems in each subfield seem distinctive due to the differences in their output structures (see Section III-A), they do share common traits (see Section III-B) and encounter common challenges brought

by the characteristics of the output labels. In this article, we attempt to provide such a view.

A. Four Vs Challenges of Multiple Outputs

The popular four Vs, i.e., volume, velocity, variety, and veracity, have been well established as the main characteristics of big data. When scholars discuss the four Vs in multi-output learning scenarios, they are usually referring to input data; however, the four Vs can also be used to describe output labels. Moreover, these four Vs bring with them a set of challenges to multi-output learning processes, explained as follows.

1) Volume refers to explosive growth in output labels, which poses many challenges to multi-output learning. First, output label spaces can grow extremely large, which causes scalability issues. Second, the burden for label annotators is significantly increased, and still, there are often insufficient annotations in a data set to adequately train a model. In turn, this may lead to unseen outputs during testing. Third, the volume may pose label imbalance issues, especially if not all the generated labels in a data set have sufficient data instances (inputs).
2) Velocity refers to how rapidly output labels are acquired, which includes the phenomenon of concept drift [6]. Velocity can present challenges due to changes in output distributions, where the target outputs vary over time in unforeseen ways.
3) Variety refers to the heterogeneous nature of output labels. Output labels are gathered from multiple sources and are of various data formats with different structures. In particular, output labels with complex structures can create multiple challenges in multi-output learning, such as finding an appropriate method of modeling output dependencies, or how to design a multi-variate loss function, or how to design efficient algorithms.
4) Veracity refers to differences in the quality of the output labels. Issues such as noise, missing values, abnormalities, or incomplete data are all characteristics of veracity.

B. Purpose and Organization of This Survey

The goal of this article is to provide a comprehensive overview of the multi-output learning paradigm using the four Vs as a frame for the current and future challenges facing this field of study. Multi-output learning has attracted significant attention from many machine learning disciplines, such as part-of-speech (POS) sequence tagging, language translation and natural language processing, motion tracking and optical character recognition in computer vision, and document categorization and ranking in information retrieval. We expect this survey to deliver a complete picture of multi-output learning and a summation of the different problems being tackled across multiple communities. Ultimately, we hope to promote further development in multi-output learning and inspire researchers to pursue worthy and needed future research directions.

The remainder of this survey is structured as follows. Section II illustrates the life cycle of output labels to help understand the challenges presented by the four Vs. Section III provides an overview of the myriad output structures along with definitions for the common subproblems addressed in multi-output learning. This section also includes some brief details on the common metrics and publicly available data used when evaluating models. Section IV presents the challenges in multi-output learning presented by the four Vs and their corresponding representative works. Section V concludes the survey.

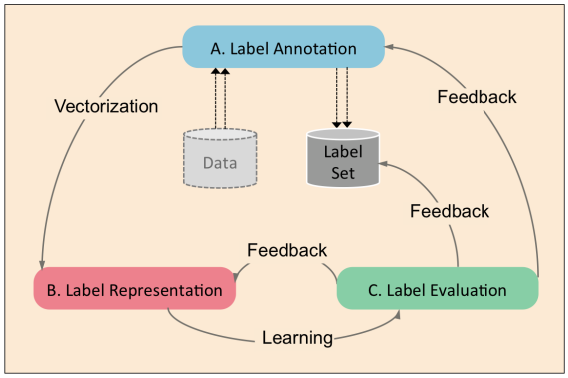


Fig. 1. Life cycle of the output label.

II. Life Cycle of Output Labels

Output labels play an important role in multi-output learning tasks in that how well a model performs a task relies heavily on the quality of those labels. Fig. 1 depicts the three stages of a label's life cycle: annotation, representation, and evaluation. A brief overview of each stage follows along with the underlying issues that could potentially harm the effectiveness of multi-output learning systems.

A. How Data Is Labeled

Label annotation requires a human to semantically annotate a piece of data and is a crucial step for training multi-output learning models. Data can be used directly with its basic annotations, or once labeled; they can be aggregated into sets for further analysis. Depending on the application and the task, label annotations come in various types. For example, the images for an image classification task should be labeled with tags or keywords, whereas a segmentation task would require each object in the images to be localized with a mask. A captioning task would require the images to be labeled with some textual descriptions, and so on.

Typically, creating large annotated data sets from scratch is time-consuming and labor-intensive no matter the annotation requirement. There are multiple ways to acquire labeled data. Social media provides a platform for researchers to search for labeled data sets, for example, Facebook and Flickr, which allow users to post pictures and comments with tags. Open-source collections, such as WordNet and Wikipedia, can also be useful sources of labeled data sets.

Beyond directly obtaining labeled data sets, crowdsourcing platforms, such as Amazon Mechanical Turk, help researchers solicit labels for unlabeled data sets by recruiting online workers. The annotation type depends on the modeling task, and due to the efficiency of crowdsourcing, this method has quickly become a popular way of obtaining labeled data sets.

ImageNet [7] is a popular data set that was labeled through a crowdsourcing platform. Its database of images is organized into a WordNet hierarchy, and it has been used to help researchers solve problems in a range of areas.

There are also many annotation tools that have been developed to annotate different types of data. LabelMe [8], a web-based tool, provides users with a convenient way to label every object in an image and also correct labels annotated by other users. BRAT [9] is also web-based but is specifically designed for natural language processing tasks, such as named-entity recognition and POS-tagging. TURKSENT [10] is an annotation tool to support sentiment analysis in social media posts.

B. Forms of Label Representations

There are many different types of label annotations for different tasks, such as tags, captions, and masks, and each type of annotation may have several representations, which are frequently represented as vectors. For example, the most common is the binary vector, whose size equals the vocabulary size of the tags. Annotated samples, e.g., samples with tags, are assigned with a value of 1 and the rest are given a 0. However, binary vectors are not optimal for more complex multi-output tasks because these representations do not preserve all useful information. Details such as the semantics or the inherent structure are lost. To tackle this issue, alternative representation methods have been developed. For instance, real-valued vectors of tags [11] indicate the strength and degree of the annotated tags using real values. Binary vectors of the associations between a tag's attributes have been used to convey the characteristics of tags. Hierarchical label embedding vectors [12] capture the structure information in tags. Semantic word vectors, such as Word2Vec [13], can be used to represent the semantics and/or context of tags and text descriptions. What is key in real-world multi-output applications is to select the label representation that is most appropriate for the given task.

C. Label Evaluation and Challenges

Label evaluation is an essential step in guaranteeing the quality of labels and label representations. Thus, label evaluation plays a key role in the performance of multi-output tasks. Different models can be used to evaluate label quality, which to choose depends on the task. Generally, labels can be evaluated in three different respects: 1) whether the annotation is of good quality (Step A); 2) whether the chosen label representation represents the labels well (Step B); and 3) whether the provided label set adequately covers the data set (Label Set). After evaluation, a human expert is generally required to explore any underlying issues and provide feedback to improve different aspects of the labels if needed.

1) Issues of Label Annotation: The aforementioned annotation methods, e.g., crowdsourcing, annotation tools, and social media, help researchers collect annotated data efficiently. However, without experts, these annotations methods are highly likely to result in the so-called noisy label problem, which includes both missing annotations and incorrect annotations. There are various reasons for noisy labels, for example, using crowdsourced workers that lack the required domain knowledge, social media users that include irrelevant tags with their image or post, or ambiguous text in a caption.

2) Issues of Label Representation: Output labels can also have internal structures, and often, this structure information is critical to the performance of the multi-output learning task at hand. Tag-based information retrieval [14] and image captioning [15] are two examples where structure is crucial. However, incorporating this information into representation as the labels is a nontrivial undertaking as the data are usually many and domain knowledge is required to define their structure. In addition, the output label space might contain ambiguity. For example, a bag-of-words (BOW) is traditionally used as a representation of a label space in natural language processing tasks, but BOW contains word sense ambiguity, as two different words may have the same meaning and one word might refer to multiple meanings.

3) Issues of the Label Set: Constructing a label set for data annotation requires a human expert with domain knowledge. Plus, it is common that the provided label set does not contain sufficient labels for the data, perhaps due to fast data growth or the low occurrence of some labels. Therefore, there are likely to be unseen labels in the test data, which leads to open-set [16], zero-shot [17], or concept drift [18] problems.

III. Multi-Output Learning

In contrast to traditional single-output learning, multi-output learning can concurrently predict multiple outputs. The outputs can be of various types and structures, and the problems that can be solved are diverse. A summary of the subfields that use multi-output learning along with their corresponding output types, structures, and applications is presented in Table I.

We begin this section with an introduction to some of the output structures in multi-output learning problems. The different problem definitions common to various subfields are provided next, along with the different constraints placed on the output space. We also discuss some special cases of these problems and give a brief overview of some of the evaluation metrics that are specific to multi-output learning. This section concludes with some insights into the evolution of output dimensions through an analysis of several commonly used data sets.

A. Myriads of Output Structures

The increasing demand for sophisticated decision-making tasks has led to new creations of outputs, some of which have complex structures. With social media, social networks, and various online services becoming ubiquitous, a wide range of output labels can be stored and then collected by researchers. Output labels can be anything; they could be text, images, audio, or video, or a combination as multimedia. For example, given a long document as input, the output might be a summary of the input in text format. Given some text fragments, the output might be an image with its contents described by the input text. Similarly, audio, such as music and videos, can be generated given different types of inputs. In addition to the different output types, there are also a number of different possible output structures. Here, we present several typical output structures given an image as an input using the example

TABLE I
SUMMARY OF SUBFIELDS OF MULTI-OUTPUT LEARNING AND THEIR CORRESPONDING OUTPUT STRUCTURES, APPLICATIONS, AND DISCIPLINES

Subfield	Output Structure	Application	Discipline
Multi-label Learning	Independent Binary Vector	Document Categorization [19]	Natural Language Processing
		Semantic Scene Classification [20]	Computer Vision
		Automatic Video Annotation [21]	Computer Vision
Multi-target Regression	Independent Real-valued Vector	River Quality Prediction [22]	Ecology
		Natural Gas Demand Forecasting [23]	Energy Meteorology
		Drug Efficacy Prediction [24]	Medicine
Label Distribution Learning	Distribution	Head Pose Estimation [25]	Computer Vision
		Facial Age Estimation [26]	Computer Vision
		Text Mining [27]	Data Mining
Label Ranking	Ranking	Text Categorization Ranking [28]	Information Retrieval
		Question Answering [29]	Information Retrieval
		Visual Object Recognition [30]	Computer Vision
Sequence Alignment Learning	Sequence	Protein Function Prediction [31]	Bioinformatics
		Language Translation [32]	Natural Language Processing
		Named Entity Recognition [33]	Natural Language Processing
Network Analysis	Graph	Scene Graph [34]	Computer Vision
	Tree	Natural Language Parsing [35]	Natural Language Processing
	Link	Link Prediction [36]	Data Mining
Data Generation	Image	Super-resolution Image Reconstruction [37]	Computer Vision
	Text	Language Generation	Natural Language Processing
	Audio	Music Generation [38]	Signal Processing
Semantic Retrieval	Independent Real-valued Vector	Content-based Image Retrieval [39]	Computer Vision
		Microblog Retrieval [40]	Data Mining
		News Retrieval [41]	Data Mining
Time-series Prediction	Time Series	DNA Microarray Data Analysis [42]	Bioinformatics
		Energy Consumption Forecasting [43]	Energy Meteorology
		Video Surveillance [44]	Computer Vision

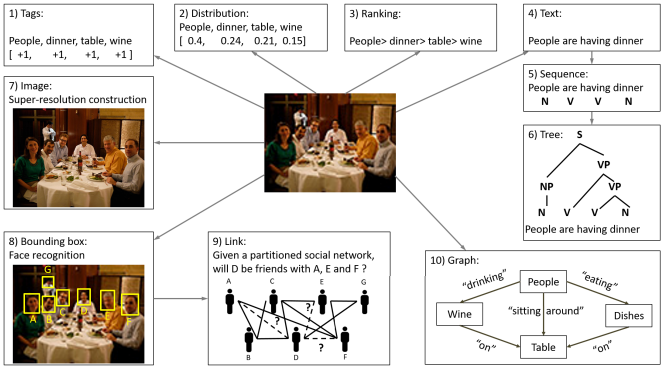


Fig. 2. Illustration of the myriads of output structures given an input image from a social network.

in Fig. 2 as a way to illustrate just how many output structures might be possible across all the different input types.

1) Independent Vector: An independent vector is a vector with separate dimensions (features), where each dimension represents a particular label that does not necessarily depend on other labels. Binary vectors can be used to represent a given piece of data as tags, attributes, BOW, bag-of-visual-words, hash codes, and so on. Real-valued vectors provide the weighted dimensions, where the real value represents the strength of the input data against the corresponding label. Applications include annotation or classification of text, images, or video with binary vectors [19]–[21] and demand or energy prediction with real-valued vectors [23]. An independent vector can be used to represent the tags of an image, as shown in Fig. 2(1), where all the tags "people," "dinner," "table," and "wine" have equal weight.

2) Distribution: Unlike independent vectors, distributions provide information about the probability that a particular dimension will be associated with a particular data sample. In Fig. 2(2), the tag with the largest weight is "people" and is the main content of the image, while "dinner" and "table" have similar distributions. Applications for distribution outputs include head pose estimation [25], facial age estimation [26], and text mining [27].

3) Ranking: Outputs might also be in the form of a ranking, which shows the tags ordered from the most to least important. The results from a distribution learning model can be converted into a ranking, but a ranking model is not restricted to only distribution learning models. Text categorization [28], question answering [29], and visual object recognition [30] are applications where rankings are often used.

4) Text: Text can be in the form of keywords, sentences, paragraphs, or even documents. Fig. 2(4) illustrates an example of text output as a caption of the image—"People are having dinner." Other applications for text outputs are document summarization [45] and paragraph generation [46].

5) Sequence: Sequence outputs refer to a series of elements selected from a label set. Each element is predicted depending on the input as well as the predicted output(s) from the preceding element. An output sequence often corresponds to an input sequence. For example, in speech recognition, we expect the output to be a sequence of text that corresponds to a given audio signal of speech [47]. In language translation, we expect the output to be a sentence transformed into the target language [32]. In the example shown in Fig. 2(5), the input is an image caption, i.e., text, and the outputs are POS tags for each word in the sequence.

6) Tree: Tree outputs are essentially the outputs in the form of a hierarchy. The outputs, usually labels, have an internal structure where each output has a label that belongs to, or is connected to, its ancestors in the tree. For example, in syntactic parsing [35], as shown in Fig. 2(6), each of the outputs for an input sentence is a POS tag and the entire output is a parsing tree; "people" is labeled as a noun N, but it is also a noun phrase NP as per the tree.

7) Image: Images are a special form of output that consists of multiple pixel values, where each pixel is predicted depending on the input and the pixels around it. Fig. 2(7) shows super-resolution construction [37] as one popular application where images are common outputs. Super-resolution construction means constructing a high-resolution image from a low-resolution image. Other image output applications include text-to-image synthesis [48], which generates images from natural language descriptions, and face generation [49].

8) Bounding Box: Bounding boxes as outputs are often used to find the exact locations of an object or objects appearing in an image. This is a common task in object recognition and object detection [30]. In Fig. 2(8), each of the faces is localized by a bounding box so that each person can be identified.

9) Link: Links as outputs usually represent the association between two nodes in a network [36]. Fig. 2(9) illustrates a task to predict whether two currently unlinked users will be friends in the future given a partitioned social network where the edges represent friendships between users.

10) Graph: Graphs are commonly used to model relationships between. They consist of a set of nodes and edges, where a node represents an object and an edge indicates a relationship between two objects. Scene graphs [50], for example, are often output as a way to describe the content of an image [34]. Fig. 2(10) shows that given an input image, the output is a graph definition where the nodes are the objects appearing in the image, i.e., "people," "dinner," "table," and "wine," and the edges are the relationships between these objects. Scene graphs are very useful as representations for tasks, such as image generation [51] and visual question answering [52].

11) Other Outputs: Beyond these few types, there are still many other types of output structures. For example, contour and polygon outputs are similar to bounding boxes and can be used as labels for object localization. In information retrieval, the output(s) could be of the list type, say, of data objects that are similar to the given query. In image segmentation, the outputs are usually segmentation masks of different objects. In signal processing, outputs might be audio of speech or music. In addition, some real-world applications may require more sophisticated output structures relating to multiple tasks. For example, one may require that objects be recognized and localized at the same time, such as in cosaliency, i.e., discovering the common saliency of multiple images [53], simultaneously segmenting similar objects given multiple images in cosegmentation [54], or detecting and identifying objects in multiple images in object codetection [55].

B. Problem Definition of Multi-Output Learning

Multi-output learning maps each input (instance) to multiple outputs. Assume that $\mathcal{X} = \mathbb{R}^d$ is a d-dimensional input space, and $\mathcal{Y} = \mathbb{R}^m$ is an m-dimensional output label space. The aim of multi-output learning is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ from the training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)|1 \leq i \leq n\}$. 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 $\mathbf{x}_i$. The general definition of multi-output learning is given as follows: finding a function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ based on the training sample of input–output pairs, where $F(\mathbf{x}, \mathbf{y})$ is a compatibility function that evaluates how compatible the input $\mathbf{x}$ and the output $\mathbf{y}$ are. Then, given an unseen instance $\mathbf{x}$ at the test state, the output is predicted to be the one with the largest compatibility score, namely, $f(\mathbf{x}) = \widetilde{\mathbf{y}} = \arg\max_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x}, \mathbf{y})$ [56].

This definition provides a general framework for multi-output learning problems. Although different multi-output learning subfields vary in their output structures, they can be defined within this framework given certain constraints on the output label space $\mathcal{Y}$.

We selected several popular subfields and present the constraints of their output space in Sections III-B1–III-B9. Note that multi-output learning is not restricted to these particular scenarios; they are just examples for illustration.

1) Multi-Label Learning: The task of multi-label learning is to learn a function $f(\cdot)$ that predicts the proper label sets for unseen instances [1]. In this task, each instance is associated with a set of class labels/tags and is represented by a sparse binary label vector. A value of $+1$ denotes that the instance is labeled and 1 means unlabeled. Thus, $\mathbf{y}_i \in \mathcal{Y} = \{-1, +1\}^m$. Given an unseen instance $\mathbf{x} \in \mathcal{X}$, the learned multi-label classification function $f(\cdot)$ outputs $f(\mathbf{x}) \in \mathcal{Y}$, where the labels in the output vector with a value of $+1$ are used as the predicted labels for $\mathbf{x}$.

2) Multi-Target Regression: The aim of multi-target regression is to simultaneously predict multiple real-valued output variables for one instance [5], [57]. Here, multiple labels are associated with each instance, represented by a real-valued vector, where the values represent how strongly the instance corresponds to a label. Therefore, we have the constraint of $\mathbf{y}_i \in \mathcal{Y} = \mathbb{R}^m$. Given an unseen instance $\mathbf{x} \in \mathcal{X}$, the learned multi-target regression function $f(\cdot)$ predicts a real-valued vector $f(\mathbf{x}) \in \mathcal{Y}$ as the output.

3) Label Distribution Learning: Label distribution learning determines the relative importance of each label in the multi-label learning problem [58]. This is opposed to multi-label learning, which simply learns to predict a set of labels. However, as illustrated in Fig. 2, the idea of label distribution learning is to predict multiple labels with a degree value that represents how well each label describes the instance. Therefore, the sum of the degree values for each instance is 1. Thus, the output space for label distribution learning satisfies $\mathbf{y}_i = (y_i^1, y_i^2, \ldots, y_i^m) \in \mathcal{Y} = \mathbb{R}^m$ 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: The goal of label ranking is to map instances to a total order over a finite set of predefined

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

5) Sequence Alignment Learning: Sequence alignment learning aims to identify the regions of relationships between two or more sequences. The outputs in this task are a sequence of multiple labels for the input instance. The output vector has the constraint $\mathbf{y}_i \in \mathcal{Y} = \{0, 1, \ldots, c\}^m$, where c denotes the total number of labels. In sequence alignment learning, m may vary depending on the input. Given an unseen instance $\mathbf{x} \in \mathcal{X}$, the learned sequence alignment function $f(\cdot)$ 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 explores the relationships and interactions between objects and entities in a network structure, and link prediction is a common task within this subfield. Let $G = (V, E)$ denote the graph of a network. V is the set of nodes, which represent objects, and E is the set of edges, which represents the relationships between objects. 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 whose value represents whether there will be an edge $e = (u, v)$ between any pair of nodes $u, v \in V$ and $e \notin E$. m is the number of node pairs that do not appear 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 a subfield of multi-output learning that aims to create and then output structured data of a certain distribution. Deep generative models are usually used to generate the data, which may be in the form of text, images, or audio. The multiple output labels in the problem become the different words in the vocabulary, the pixel values, the audio tones, and so on. Take image generation as an example. The output vector has the constraint $\mathbf{y}_i \in \mathcal{Y} = \{0, 1, \ldots, 255\}^{m_w \times m_h \times 3}$, where m_w and m_h are the width and height of the image. Given 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 $f(\cdot)$ 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: Semantic retrieval means finding the meanings within some given information. Here, we consider semantic retrieval in a setting where each input instance has semantic labels that can be used to help retrieval [59]. Thus, each instance representation comprises 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 $f(\cdot)$ 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: The goal in time-series prediction is to predict the future values in a series based on previous observations [60]. The inputs are a series of data vectors for a period of time, and the output is a data vector for a future timestamp. Let t denote 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 $t = T$ are $\mathbf{y}_i = (\mathbf{y}_i^0, \ldots, \mathbf{y}_i^t, \ldots \mathbf{y}_i^T)$. Given the 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 a traditional single-output learning paradigm if the output class is represented as either an integer encoding or a one-hot vector.

2) Fine-Grained Classification: Fine-grained classification is a challenging multi-classification task where the categories may only have subtle visual differences [61]. Although the output of fine-grained classification shares the same vector representation as multi-class classification, the vectors have different internal structures. Also, in its label hierarchy, labels with the same parents tend to be more closely related than labels with different parents.

3) Multi-Task Learning: The aim of multi-task learning (MTL) is the subfield that aims to improve generalization performance by learning multiple related tasks simultaneously [62], [63]. Each task in the problem outputs one single label or value. This can be thought of as part of the multi-output learning paradigm in that learning multiple tasks is similar to learning multiple outputs. MTL leverages the relatedness between tasks to improve the performance of learning models. One major difference between MTL and multi-output learning is that in MTL, different tasks might be trained on different training sets or features, while in multi-output learning, the output variables usually share the same training data or features.

D. Model Evaluation Metrics

In this section, we present the conventional evaluation metrics used to assess the multi-output learning models with a test data set. Let $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{y}_i) | 1 \leq i \leq N\}$ be the test data set, $f(\cdot)$ be the multi-output learning model, and $\hat{\mathbf{y}}_i = f(\mathbf{x}_i)$ be the predicted output of $f(\cdot)$ 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. $\mathbb{I}$ is an indicator function, where $\mathbb{I}(g) = 1$ if g is true, and 0 otherwise.

1) Classification-Based Metrics: Classification-based metrics evaluate the performance of multi-output learning with respect to classification problems, such as multi-label classification, semantic retrieval, image annotation, and label ranking. The outputs are usually in discrete values. The conventional classification metrics fall into three groups: *example-based*, *label-based*, and *ranking-based*.

a) Example-based metrics: Example-based metrics [64] evaluate the performance of multi-output learning models with respect to each data instance. Performance is first evaluated

on each test instance separately, and then, the mean of all the individual results is used to reflect the overall performance of the model. The evaluation for multi-output classification tasks works under the same mechanism as binary classification (single output) tasks: the classic metrics for binary classification can be extended to evaluate multi-output classification models [1]. The commonly used metrics are exact match ratio, accuracy, precision, recall, F_1 score and hamming loss.

b) Label-based metrics: Label-based metrics evaluate performance with respect to each output label. These metrics aggregate the contributions of all the labels to arrive at an averaged evaluation of the model. There are two techniques for obtaining label-based metrics: macroaveraging and microaveraging. Macroaveraging-based approaches compute the metrics for each label independently and then average over all the labels with equal weights. By contrast, microaveraging-based approaches give equal weight to every data sample. Let TP_l, FP_l, TN_l, and FN_l denote the number of true positives, true negatives, false positives, and false negatives, for each label, respectively. Let B be a binary evaluation metric (accuracy, precision, recall, or F_1 score) for a particular label. The macroapproach and microapproach are therefore defined as follows.

Macroaveraging:

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

Microaveraging:

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

c) Ranking-based metrics: Ranking-based metrics evaluate the performance in terms of the ordering of the output labels.

One-error is the number of times the top-ranked label is not in the true label set. This approach only considers the most confident predicted label of the model. An 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 an indicator function, $\mathcal{L}$ denotes the label set, and $\pi_i(\lambda)$ is the predicted rank of label λ for the test instance $\mathbf{x_i}$. The smaller the one-error, the better the performance.

Ranking loss indicates the average proportion of incorrectly ordered label pairs

$$\text{Ranking Loss} = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{|Y_i||\overline{Y_i}|}|E|$$

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 the proportion of the labels ranked above a particular label in the true label set as an average over all the true labels. The larger the value, the better the performance of the model. The averaged AP over all test data instances is defined as follows:

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

Discussion: The metrics listed earlier are those commonly used with classification-based multi-output learning problems, but the choice of metrics varies according to the different considerations of each application. Take image annotation for example. If the aim of the task is to annotate each image correctly, example-based metrics are optimal for evaluating performance. However, if the objective is keyword-based image retrieval, the macroaveraging metric is preferable [64]. Furthermore, some metrics are more suited to special cases of multi-output learning problems. For instance, for imbalanced learning tasks, geometric mean [65] for some classification-based metrics, e.g., the errors, accuracy, and F1-scores, are more convincing to be used for evaluation. The minimum sensitivity [66] can help determine the classes that hinder the performance in the imbalanced setting. We do not discuss these metrics in detail as they are not the focus here.

2) Regression-Based Metrics: Unsurprisingly, regression-based metrics evaluate multi-output learning performance with regression problems, e.g., object localization or image generation. The outputs are usually real values. The commonly used regression-based metrics are mean absolute error (MAE), mean squared error (MSE), average correlation coefficient (ACC) and intersection over union (IoU). Details for these metrics can be found in the Supplementary Materials.

3) New Metrics: Data generation is an emerging subfield of multi-out learning that uses generative models to output structured data with certain distributions. Based on the particulars of the task at hand, a model's performance is usually evaluated in two respects: 1) whether the generated data actually follows the desired real data distribution and 2) the quality of the generated samples. Metrics such as average log-likelihood [67], coverage metric [68], maximum mean discrepancy (MMD) [69], and geometry score [70] are frequently used to assess the veracity of the distribution. Metrics that quantify the quality of the generated data remain challenging. The commonly used are inception scores (IS) [71], mode score (MS) [72], Fréchet inception distance (FID) [73], and kernel inception distance (KID) [74]. Precision, recall, and F1 score are also employed in GANs to quantify the degree of overfitting in the model [75].

E. Multi-Output Learning Data Sets

Most of the data sets used to experiment with multi-output learning problems have either been constructed or become popular because they reflect and, therefore, test a challenge that needs to be overcome. We have presented these data sets according to the challenges reflected in the four Vs. Table II lists the data sets, including their multi-output characteristics,

TABLE II
CHARACTERISTICS OF THE DATA SETS 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	[76]
		Review Text	AmazonCat-14K	14,588	[77], [78]
		Text	Wiki10-31	30,938	[79], [80]
		Social Bookmarking	Delicious-200K	205,443	[79], [81]
		Text	WikiLSHTC-325K	325,056	[82], [83]
		Text	Wikipedia-500K	501,070	Wikipedia
		Product Network	Amazon-670K	670,091	[76], [79]
		Text	Ads-1M	1,082,898	[82]
		Product Network	Amazon-3M	2,812,281	[77], [78]
	Extreme Class Imbalance			**Largest Class Imbalance Ratio**	
		Scene Image	WIDER-Attribute	1:28	[84]
		Face Image	Celeb Faces Attributes	1:43	[85]
		Clothing Image	DeepFashion	1:733	[86]
		Clothing Image	X-Domain	1:4,162	[87]
	Unseen Outputs			**Seen / Unseen Labels**	
		Image	Attribute Pascal abd Yahoo	20 / 12	[88]
		Animal Image	Animal with Attributes	40 / 10	[88]
		Scene Image	HSUN	80 / 27	[89]
		Music	MagTag5K	107 / 29	[90]
		Bird Image	Caltech-UCSD Birds 200	150 / 50	[91]
		Scene Image	SUN Attributes	645 /72	[20]
		Health	MIMIC II	3,228 / 355	[92]
		Health	MIMIC III	4,403 / 178	[93]
Velocity	Change of Output Distribution			**Time Periods**	
		Text	Reuters	365 days	[94]
		Route	ECML/PKDD 15: Taxi Trajectory Prediction	365 days	[95]
		Route	epfl/mobility	30 days	[96]
		Electricity	Portuguese Electricity Consumption	365 days	[97]
		Traffic Video	MIT Traffic Data Set	90 minutes	[44]
		Surveillance Video	VIRAT Video	8.5 hours	[98]
Variety	Complex Structures			**Output Structures**	
		Image	LabelMe	Label, Bounding Box	[8]
		Image	ImageNet	Label, Bounding Box	[7]
		Image	PASCAL VOC	Label, Bounding Box	[99]
		Image	CIFAR100	Hierarchical Label	[100]
		Lexical Database	WordNet	Hierarchy	[101]
		Wikipedia Network	Wikipedia	Graph, Link	[102]
		Blog Network	BlogCatalog	Graph, Link	[103]
		Author Collaboration Network	arXiv-AstroPh	Link	[104]
		Author Collaboration Network	arXiv-GrQc	Link	[104]
		Text	CoNLL-2000 Shared Task	Text Chunks	[105]
		Text	Wall Street Journal (WSJ) corpus	POS Tags, Parsing Tree	-
		European Languages	Europarl corpus	Sequence	[32]
Veracity	Noisy Output Labels			**Noisy Labeled Samples**	
		Dog Image	AMT	7,354	[106]
		Food Image	Food101N	310K	[107]
		Clothing Image	Clothing1M	1M	[108]
		Web Image	WebVision	2.4M	[109]
		Image and Video	YFCC100M	100M	[110]

the challenge can be tested, the application domain, plus the data set name, source, and descriptive statistics.

The large-scale data sets, i.e., the data sets that can be used to test volume, are extremely large. The enormity of their corresponding statistics illustrates the pressing need to overcome the challenges caused by this particular V among the 4.

Many studies that have focused on change in output distribution, e.g., concept drift/velocity, rely on synthetic streaming data or static databases in their experiments. We have also included some of the more popular real-world and/or dynamic databases that are used to experiment with these tasks. As shown in the table, the data sets come from various application domains, demonstrating the importance of this challenge.

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

The data sets designed to test complex multi-output learning problems contain a mix of different output structures. For example, the image data sets listed in the table includes both labels and bounding boxes for the objects. These data sets can be used to test a variety of data.

Finally, we come to veracity. Many efforts to detail with noisy labels evaluate their methods by beginning with a clean data set to which artificial noise is then added. This helps researchers control and test different levels of noise. We have also listed several popular real-world data sets with some unknown level of errors in the annotation.

IV. 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 these emerging phenomena. Furthermore, given the success of artificial neural networks (ANNs), we also present several state-of-the-art examples of multi-output learning using an ANN for each challenge.

A. Volume—Extreme Output Dimensions

Large-scale data sets are ubiquitous in real-world applications. A data set is defined to be large-scale if it meets one of three criteria: it has a large number of data instances, the input feature space has high dimensionality, or the output space has high dimensionality. Many studies have sought to solve the scalability issues caused by a large number of data instances, e.g., the instance selection method in [212], or with high-dimensional feature spaces, such as the feature selection method in [213]. However, the issues associated with high output dimensions have received much less attention.

Consider, for example, that if the label for each dimension of an m-dimensional output vectors can be selected from a label set with c different labels, then the number of output outcomes is c^m. Hence, these ultrahigh-output dimensions/labels result in an extremely large output space and, in turn, high computation costs. Therefore, it is crucial to design multi-output learning models that can handle the immense and ongoing growth in outputs.

An analysis of the current state-of-the-art research on ultrahigh-output dimensions revealed some interesting insights. Our analysis was based on the data sets used in studies of multiple disciplines, such as machine learning, computer vision, natural language processing, information retrieval, and data mining. We specifically focused on articles in three top journals and three top international conferences: the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), the *Journal of Machine Learning Research* (JMLR), the International Conference on Machine Learning (ICML), the Conference on Neural Information Processing Systems (NIPS), and the Conference on Knowledge Discovery and Data Mining (KDD). Figs. 3 and 4 summarize our review. From Figs. 3 and 4, it is evident that the output dimensionality of the understudied algorithms has continued to increase over time. In addition, the latest articles to address this issue in all selected titles are now dealing with more than a million output dimensions and, in some cases, are approaching billions of outputs. Moreover, the statistics for the conferences with shorter time-lags to publication demonstrate just how rapidly output dimensionality is increasing. From this analysis, we conclude that the explosion in output dimensionality is driving many developments in multi-output learning algorithms.

The studies we reviewed tend to fall into two categories: qualitative and quantitative approaches. The qualitative approaches generally involve generative models, while the quantitative models generally involve discriminative models. The main difference between the two models is that generative models focus on learning the joint probability $P(x, y)$ of the inputs x and the label y, while the discriminative models focus on the posterior $P(y|x)$. Note that in a generative model, $P(x, y)$ can be used to generate some data x, where, in this case, x is the generated output in this particular case.

1) Qualitative Approaches/Generative Models: The aim of image synthesis [48], [214] is to synthesize new images from textual image descriptions of the image. Some pioneering researchers have synthesized images using a GAN with the image distribution as multiple outputs [67]. However, in real life, GANs can only generate low-resolution images. However, since the first attempts at this foray, there has been a progress in scaling up GANs to generate high-resolution images with sensible outputs. For example, Reed *et al.* [48] proposed a GAN architecture that generates visually plausible 64×64 pixel images given text descriptions. In a follow-up study, they presented GAWWN [214], which scales the synthesized image up to 128×128 resolution by leveraging additional annotations. Subsequently, StackGAN [215] was proposed, which is capable of generating photo-realistic images at a 256×256 resolution from text descriptions. HDGAN [216] is the current state of the art in image synthesis. It models high-resolution images in an end-to-end fashion at 512×512 pixels. Inevitably, the future will see further increases in resolution.

MaskGAN [217] uses GAN to generate text (i.e., meaningful word sequences). The label set size accords with the vocabulary size. The output dimension is the length of the word sequence that is generated, which, technically, can be unlimited. However, MaskGAN only handles sentence-level text generation. Document- and book-level text generations are still challenging.

2) Quantitative Approaches/Discriminative Models: Like instance and feature selection methods that reduce the number of input instances and, in turn, reduce input dimensionality, it is natural to design models that similarly reduce output dimensionality. Embedding methods can be used to compress a space by projecting the original space onto a lower dimensional space, with the expected information preserved, such as label correlations and neighborhood structure. Popular methods, such as random projections or canonical correlation analysis projections [218]–[221], 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, the predicted compressed label can be projected back onto the original high-dimensional label space. Recently, several embedding methods have been proposed for extreme output dimensions. Mineiro and Karampatziakis [222] proposed a novel randomized embedding for extremely large output spaces. AnnexML [169] is another novel embedding method for graphs that captures graph structures in the embedding space. The embeddings are constructed from the k-nearest neighbors (kNNs) of the label vectors, and the predictions are made efficiently through an approximate nearest neighbor search method. Two popular ANN methods for handling extreme output dimensions are fastText learn tree [223] and XML-CNN [224]. FastText learn tree [223] jointly learns the data representation and the tree structure, and the learned

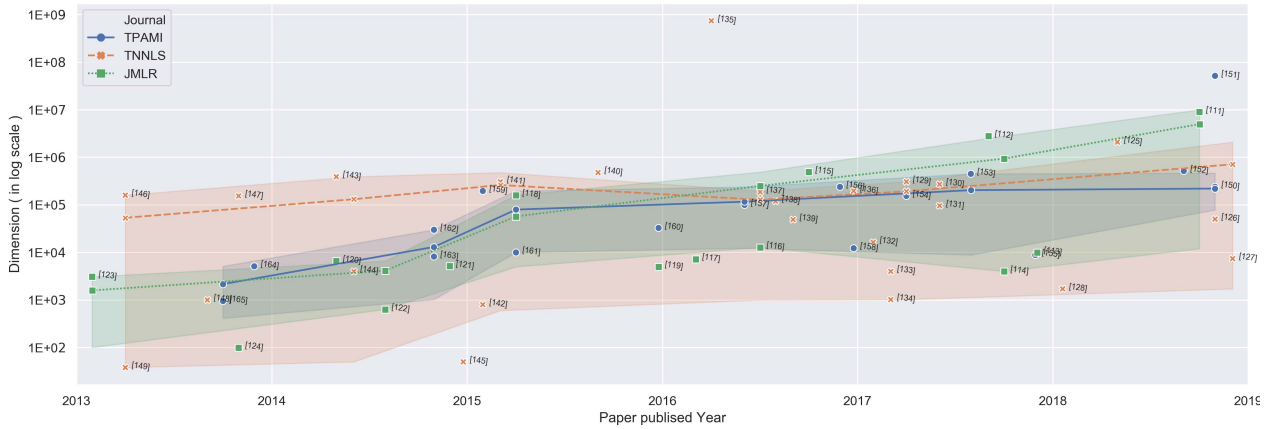


Fig. 3. Output dimension trends from articles published in the journals TPAMI, TNNLS, and JMLR since 2013 [111]–[165].

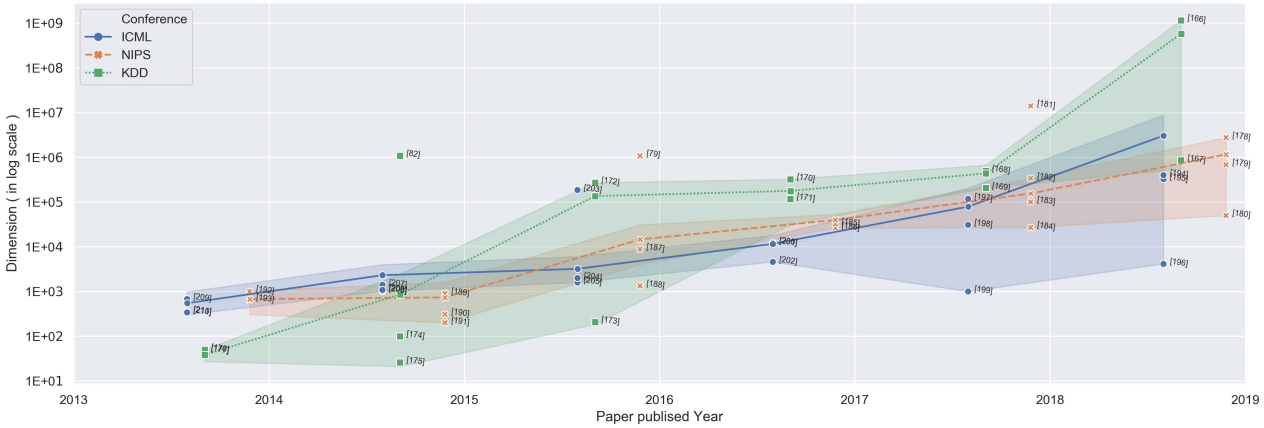


Fig. 4. Output dimension trends from articles published in the conferences ICML, NIPS, and KDD since 2013 [79], [82], [166]–[211].

tree structure is then used for efficient hierarchical prediction. XML-CNN is a CNN-based model that incorporates 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. Variety—Complex Structures

With the increasing abundance of labels, there is a pressing need to understand their inherent structures. Complex output structures can lead to multiple challenges in multi-output learning. For instance, it is common for strong correlations and complex dependencies to exist between labels. Therefore, appropriately modeling output dependencies in the label representation is critical but nontrivial in multi-output learning. In addition, designing a multi-variate loss function and proposing an efficient algorithm to alleviate the high complexity caused by complex structures are also challenging.

1) Appropriate Modeling of Output Dependencies: The simplest method of multi-output learning is to decompose the learning problem into m independent single-output problems with each corresponding to a single value in the output space. A representative approach is a binary relevance (BR) [225], which independently learns binary classifiers for all the labels in the output space. Given 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. Hence, it is crucial to model the output dependencies appropriately to obtain better performance for multi-output tasks.

Many classic learning methods have been proposed to model multiple outputs with interdependencies. These include label powersets (LPs) [226], classifier chains (CCs) [227], [228], structured support vector machine (SSVM) [56], conditional random fields (CRFs) [229], and so on. LPs model the output dependencies by treating each different combination of labels in the output space as a single label, which transforms the problem into one of 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 cost when training with a large number of output labels. Random k-labelsets [230], an ensemble of LP classifiers, is a variant of LP that alleviates the computational complexity problem by training each LP classifier on a different random subset of labels.

CC improves BR by taking the output correlations into account. It links all the binary classifiers from BR into a chain via a modified feature space. Given the jth label, the instance $\mathbf{x}_i$ is augmented with the first, second, ..., $(j-1)$th label, i.e., $(\mathbf{x}_i, l_1, l_2, \ldots, l_{j-1})$, as the input to train the jth classifier. Given an unseen instance, CC predicts the output using the first classifier and then augments the instance with the prediction from the first classifier as the input to the second classifier for predicting the next output. CC processes values in this way from the first classifier to the last and, thus, preserves the output correlations. However, a different order of chains leads to different results. ECC [227], an ensemble of CC, was proposed to solve this problem. It trains the classifiers over a set of random ordering chains and averages the results. Probabilistic CCs (PCCs) [231] provide a probabilistic interpretation of CC by estimating the joint distribution of the output labels to capture the output correlations. CCMC [114] is a CC model that considers the order of label difficulties to reduce the degradation in performance caused by ambiguous labels. It is an easy-to-hard learning paradigm that identifies easy and hard labels and uses the predictions for easy labels to help solve the harder labels.

SSVM leverages the idea of large margins to deal with multiple interdependent outputs. The 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 $\Phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^q$ is the joint feature map over input and output pairs. The SSVM aims to find the classifier $h_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle$ with the following objective:

$$\min_{\mathbf{w} \in R^q, \{\xi_i \geq 0\}_{i=1}^n} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} \underbrace{\max_{\mathbf{y} \in \mathcal{Y}} \{\Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y})\} - \mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y}_i)}_{\text{structured hinge loss}}$$

Constraining the structured hinge loss with $\Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y}) - \mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y}_i) \leq \xi_i$ for all $y \in \mathcal{Y}$, the objective can be reformulated as

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

where $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a loss function, C is a positive constant that controls the tradeoff between the training error minimization and the margin maximization [56], n is the number of training samples, and ξ_i is the slack variable. In practice, SSVM is solved with the cutting-plane algorithm [232].

Apart from the classic models that learn the correlations between outputs, some of the state-of-the-art multi-output learning models are based on ANNs. For example, models based on convolutional neural networks typically focus on hierarchical multi-labels [233] or rankings [234]. Recurrent neural networks (RNNs) model generally focus on sequence-to-sequence learning [235] and time-series prediction [236]. Generative deep neural networks are used to generate output data, such as images, text, and audio [67].

2) Multivariate Loss Functions: Various loss functions were defined to compute the difference between the groundtruth and the predicted output. Different loss functions present different errors given the same data set, and they greatly affect the performance of the model.

0/1 loss is a standard loss function that is commonly used in classification [237]

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

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 dependence. Therefore, it is not suitable for large numbers of outputs or for outputs with complex structures. In addition, it is nonconvex and nondifferentiable, so it is difficult to minimize the loss using standard convex optimization methods. In practice, one typically uses a surrogate loss, which is a convex upper bound of the task loss. However, a surrogate loss in multi-output learning usually loses consistency when generalizing single-output methods to deal with multiple outputs [238]. Several works on subfields of multi-output learning study the consistency of different surrogate functions and show that they are consistent under some sufficient conditions [239], [240]. Yet, this is still a challenging aspect of multi-output learning. More exploration of the theoretical consistency of different problems is required.

In the following, we describe four popular surrogate losses: hinge loss, negative log loss, perceptron loss, and softmaxmargin loss.

Hinge loss is one of the most widely used surrogate losses and is usually used in structured SVMs [241]. It pushes the score of the correct outputs to be greater than that of the prediction

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

The margin, $\Delta(\mathbf{y}, \mathbf{y}')$, has different definitions based on the output structures and task. For example, 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 $\mathbf{y}'$ [19]. For ranking, $\Delta(\mathbf{y}, \mathbf{y}')$ can be defined as the mean AP of a ranking $\mathbf{y}'$ compared to the optimal $\mathbf{y}$ [242]. 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 [35]. Nondecomposable losses, such as the F_1 measure, AP, or IOU, can also be defined as a margin.

Negative log loss is commonly used in CRFs [229]. Note that minimizing negative log loss is the same as maximizing the log probability of the data

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

Perceptron loss is usually adopted in structured perceptron tasks [243] and is the same as hinge loss without the margin

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

Softmax-margin loss is one of the most popular loss functions in multi-output learning models, such as SSVMs [244] and CRFs [245]

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

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

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

In multi-output learning, it is usually used with continuous-valued outputs or continuous intermediate results before converting them into discrete-valued outputs. It is also commonly used in neural networks and boosting.

3) Efficient Algorithms: Complex output structures significantly increase the burden on algorithms to formulate a model. Large-scale outputs, complex output dependencies, and/or complex loss functions can all be problematic. Therefore, several algorithms have been proposed specifically to tackle these challenges efficiently. Many leverage classic machine learning models so as to speed up the algorithms and alleviate the burden of complexity. The four most widely used classic models are based on kNN, decision trees, k-means, and hashing.

1) kNN-based methods are simple yet powerful machine learning models. Predictions are made based on the closest k instances to the test instance vector in terms of the Euclidean distance. LMMO-kNN [246] is an SSVM-based model involving an exponential number of constraints with respect to the number of labels. This model imposes kNN constraints instantiated by the label vectors from neighboring examples to significantly reduce the training time and make rapid predictions.
2) Decision tree-based methods [247], [248] learn a tree from the training data with a hierarchical output label space. They recursively partition the nodes until each leaf contains a small number of labels. Each novel data point is passed down the tree until it reaches a leaf. This method usually achieves a logarithmic time prediction.
3) k-means based methods, such as SLEEC [79], 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. This significantly reduces the prediction time.
4) Hashing methods, such as cohashing [249], [250] and DBPC [251], reduce the prediction time by using hashing on the input or the intermediate embedding space. Cohashing learns an embedding space to preserve semantic similarity structures between inputs and outputs. Compact binary representations are then generated for the learned embeddings for prediction efficiency. DBPC jointly learns a deep latent Hamming space and binary prototypes while capturing the latent nonlinear structures of the data with an ANN. The learned Hamming space and binary prototypes significantly decrease the prediction complexity and reduce memory/storage costs.

C. Volume—Extreme Class Imbalances

Real-world multi-output applications rarely provide data with an equal number of training instances for all labels/classes. Too many instances in one class over another mean the data is imbalanced, this is, common in many applications. Therefore, traditional models learned from such data tend to favor majority classes more. For example, in face generation, a trained model tends to generate the faces of famous people because there are so many more images of celebrities than other people. Though class imbalance problems have been studied extensively in the context of binary classification, this issue still remains a challenge in multi-output learning, especially with extreme imbalances.

Many studies on multi-output learning either create a balanced data set or ignore the problems introduced by imbalanced data. A natural way to balance class distributions is to resample the data set. There are two main resampling techniques: undersampling and oversampling [252]. Undersampling methods downsize the majority classes. The NearMiss family of methods [253] are representative works of this category. The oversampling methods, such as SMOTE and its variants [254], adopt oversampling technique on minority classes to handle the imbalanced class learning problem. However, all these resampling methods are mainly designed for single output learning problems. There are other techniques to handle class imbalance in multi-output learning tasks with ANN.

For example, Dong *et al.* [255] combined incremental rectification of mini-batches with a deep neural network. Then, a hard sample mining strategy minimizes the dominant effect of the majority classes by discovering the boundaries of sparsely sampled minority classes. Both of the methods in [256] and [257] leveraged adversarial training to mitigate imbalance by using a reweighting technique so that majority classes tend to have a similar impact as minority classes.

D. Volume—Unseen Outputs

Traditional multi-output learning assumes that the output set in testing is the same as the one in training, i.e., the output labels of a testing instance have already appeared during training. However, this may not be true in real-world applications. For example, a new emerging living species can not be detected using a learned classifier based on existing living animals. Similarly, it is infeasible to recognize the actions or events in a real-time video if no such actions or events with the same labels appeared in the training video set, nor could a coarse animal classifier provide details of the species of a detected animal, such as whether a dog is a labrador or a shepherd.

Depending on the complexity of the learning task, label annotation is usually very costly. In addition, the enormous growth in the number labels not only leads to high-dimensional output space as a result of computation inefficiency but also makes 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. Multi-label classification problems can have various inputs, such as text, images, and video, 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 problems learn a mapping from the input to the output. However, as the label space increases, it is common to find unseen output labels during testing, where no such labels have appeared in the training set. To study such cases, the zero-shot multi-class classification problem was first proposed in [17] and [260] and most leverage the predefined semantic information, such as attributes [11] and word representations [13]. This technique was 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, Gaure *et al.* [259] leverage the co-occurrence statistics of seen and unseen labels and model the label matrix and co-occurrence matrix jointly using a generative model. Rios and Kavuluru [260] and Lee *et al.* [261] incorporate knowledge graphs of the label relationships with neural networks.

2) Zero-Shot Action Localization: Similar to zero-shot classification problems, localizing human actions in videos without any training video examples is a challenging task. Inspired by zero-shot image classification, many studies into zero-shot action classification predict unseen actions from disjunct training actions based on the prior knowledge of action-to-attribute mappings [262]–[264]. Such mappings are usually predefined, and the seen and unseen actions are linked through a description of the attributes. Thus, they can be used to generalize undefined actions but are unable to localize actions. More recently, some works are proposed to overcome the issue. Jain *et al.* [265] propose Objects2action without using any video data or action annotations. It leverages vast object annotations, images, and text descriptions that can be obtained from open-source collections, such as WordNet and ImageNet. Mettes and Snoek [266] have subsequently enhanced Objects2action by considering the relationships between actors and objects.

3) Open-Set Recognition: Traditional multi-output learning problems, including zero-shot multi-output learning, operate under a closed-set assumption, i.e., where all the testing classes are known at the time of training time either through the training samples or because they are predefined in a semantic label space. However, Scheirer *et al.* [16] proposed a concept called open-set recognition to describe a scenario where unknown classes appear in testing. Open-set recognition presents one-vs-set machine to classify the known classes as well as deal with the unknown classes. In later studies [267], [268], they extended this idea into multi-class settings by formulating a compact abating probability model. Bendale and Boult [269] adapted ANNs for open-set recognition by proposing a new model layer that estimates the probability of an input being an unknown class.

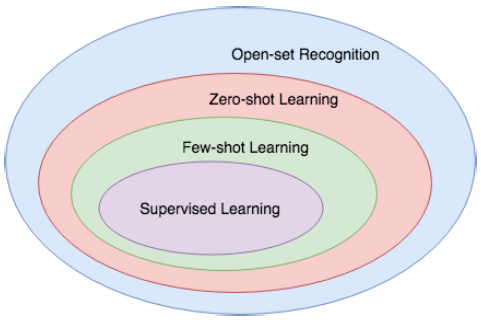


Fig. 5. Relationship among different levels of unseen outputs. All of these learning problems belong to multi-output learning.

Fig. 5 illustrates the relationships between different levels of unseen outputs in multi-output learning. Open-set recognition is the most generalized problem of all. Few-shot and zero-shot learning have studied with different multi-output learning problems, such as multi-label learning and event localization. However, open-set recognition has only been studied in conjunctions with multi-class classification. Other problems in the context of multi-output learning are still unexplored.

E. Veracity—Noisy Output Labels

Almost all methods of label annotation lead to some amount of noise for various reasons. Associations may be weak, the text may be ambiguous, and crowdsourced workers may not be domain experts, so labels may be incorrect [270]. Therefore, it is usually necessary to handle noisy outputs, such as missing, corrupt, incorrect, and/or partial labels, in real-world tasks.

1) Missing Labels: Often human annotators annotate an image or document with prominent labels but miss some of the less emphasized labels. In addition, all the objects in an image may not be localized because there are, say, too many objects or the objects are too small. Social media, such as Instagram, allow users to tag uploaded images. However, the tags could relate to anything: the type of event, the person's mood, and the weather. Plus, no user is likely to tag every object or every aspect of an image. Directly using such labeled data sets in traditional multi-output learning models cannot guarantee the performance of the given tasks. Therefore, handling missing labels is necessary in real-world applications.

In early studies, missing labels were handled by treating them as negative labels [271]–[273]. Then, modeling tasks are performed based on a fully labeled data set. However, this approach can introduce undesirable bias into the learning problem. Therefore, a more widely used method now is missing value imputation through matrix completion [186], [192], [274]. Most of these approaches are based on a low-rank assumption and, more recently, on label correlations, which improves learning performance [275], [276].

2) Incorrect Labels: Many labels in high-dimensional output space are noninformative or simply wrong [277]. This

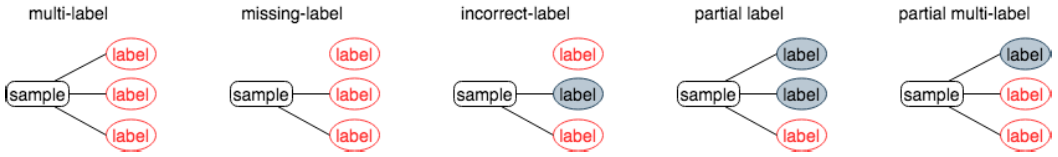


Fig. 6. Range of noisy labels in multi-label classification. Training may be *multi-label* (sample associates with multiple labels), *missing-label* (sample has incomplete label assignment), *incorrect-label* (sample has at least one incorrect labels and possible incomplete label assignment), *partial-label* (each sample has multiple labels, only one of which is correct), and *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.

is especially common with annotations from crowdsourcing platforms that hire nonexpert workers. Labeled data sets from social media networks are also often less than useful. A basic approach for handling incorrect labels is to simply remove those samples [278], [279]. That said, it is frequently difficult to detect that samples have been mislabeled. Therefore, designing multi-output learning algorithms that learn from noisy data sets is of great practical importance.

Existing multi-output learning methods handling noisy labels generally fall into two groups. The first group is based on building robust loss functions [280]–[282], which modify the labels in the loss function to alleviate the effect of noise. The second group models latent labels and learns the transition from the latent to the noisy labels [283]–[285].

Partial Labels: A special case of incorrect labels is partial labels [286]–[288], where each training instance is associated with a set of candidate labels but only one of them is correct. This is a common problem in real-world applications. For example, a photograph might contain many faces with captions listing who is in the photograph, but the names are not matched to the face. Many methods for learning partial labels have been developed to recover the ground-truth labels from a candidate set [289], [290]. However, most are based on the assumption of exactly one ground truth for each instance, which may not always hold true by different label annotation methods. With the use of multiple workers on the crowdsourcing platform to annotate a data set, 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. Hence, Xie and Huang [291] developed a new learning framework, partial multi-label learning (PML), that relaxes this assumption by leveraging the data structure information to optimize the confidence weighted rank loss. Fig. 6 summarizes all the scenarios with noisy output labels, including multi-label learning, missing labels, incorrect labels, partial label learning, and PML.

F. Velocity—Changes in Output Distribution

Many real-world applications must deal with data streams, where data arrive continuously and possibly endlessly. In these cases, the output distributions can change over time or concept drift can occur. Streaming data are common in surveillance [98], driver route prediction [95], demand forecasting [97], and many other applications. Take visual tracking [292] in the surveillance video as an example, where the video stream is potentially endless. 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. Therefore, the learning model must adapt to possible concept drift while working with limited memory.

Existing multi-output learning methods model changes in output distribution by updating the learning system each time data streams arrive. The update method might be ensemble-based [293]–[297] or ANN-based methods [292], [298]. Other strategies to handle concept drift include the assumption of a fading effect on past data [296]; maintaining a change detector on predictive performance measurements, and recalibrating models accordingly [295], [299]; using stochastic gradient descent to update the network and accommodate new data streams with an ANN [292]. Notably, the 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 [300], [301] are proposed to improve the efficiency of kNN while accommodating the changing output distribution.

G. Other Challenges

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 an open-set noisy label problem [302]. In addition, the combination of noisy labels and extreme output dimensions are also worthy of study and further exploration [206]. Changes in output distribution together with noisy labels result in online time-series prediction problems with missing values [303], while changes in distributions combined with dynamic label sets (unseen outputs) lead to open-world recognition problems with incremental labels [304]. Changing output distribution with extreme class imbalances creates the common problem of streaming data with concept drift and class imbalances at the same time [18], [305]. Moreover, the combination of complex output structures with changing output distribution is also frequent in real-world applications [306].

H. Open Challenges

1) Output Label Interpretation: There are different ways to represent output labels, and each expresses label information from a specific perspective. Taking label tags as an output, for example, binary attributed output embeddings represent what attributes the input relates to. Hierarchical label output embedding conveys the hierarchical structure of the inputs. Semantic word output embeddings reflect the semantic relationships between the outputs. As one can see, each exhibits

a certain level of human interpretability. Hence, an emerging approach to label embedding is to incorporate different label information from multiple perspectives and rich contexts to enhance interpretability [307]. This is a challenging undertaking because it is quite difficult to appropriately model the interdependencies between outputs in a way that humans can easily interpret and understand. For example, an image of a centaur is expected to be described with semantic labels, such as horse and person. Moreover, the image is expected to be described with attributes, such as head, arm, and tail. As such, appropriately modeling the relationships between input and outputs with rich interpretations of the labels is an open challenge that should be explored in the future studies.

2) Output Heterogeneity: As the demand for sophisticated decision-making increases, so does demand for outputs with more complex structures. Returning to the example of surveillance, people reidentification in traditional approaches usually consists of two steps: people detection and then reidentifying that person if they are input. These steps are essentially two separate tasks that need to be learned together if performance is to be enhanced. Several researchers have recently attempted this demanding challenge, i.e., building a model that can simultaneously learn multiple tasks with different outputs. Mousavian *et al.* [308] undertook joint people detection in tandem with reidentification, while Van Ranst *et al.* [309] tackled image segmentation with depth estimation. However, more exploration and investigation to overcome this challenge are needed. As an example, one worthy undertaking would be to answer the question: Can we simultaneously learn the representation of a new user in a social network as well as their potential links to existing users?

V. Conclusion

Multi-output learning has attracted significant attention over the last decade. This article provides a comprehensive review of the study of multi-output learning using the four Vs as a frame. We explore the characteristics of the multi-output learning paradigm beginning with the life cycle of the output labels. We emphasize the issues associated with each step of the learning process. In addition, we provide an overview of the types of outputs, the structures, selected problem definitions, common model evaluation metrics, and the popular data repositories used in experiments, with representative works referenced throughout. The article concludes with a discussion on the challenges caused by four Vs and some future research directions that are worthy of further study.

References

[1] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[2] C. Gong, D. Tao, J. Yang, and W. Liu, "Teaching-to-learn and learning-to-teach for multi-label propagation," in *Proc. AAAI*, 2016, pp. 1610–1616.

[3] 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.

[4] S. Vembu and T. Gärtner, "Label ranking algorithms: A survey," in *Preference Learning*. Berlin, Germany: Springer, 2010, pp. 45–64, doi: 10.1007/978-3-642-14125-6_3.

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

[6] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, 1996.

[7] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[8] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and Web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, 2008.

[9] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "BRAT: A Web-based tool for NLP-assisted text annotation," in *Proc. EACL*, 2012, pp. 102–107.

[10] G. Eryiğit, F. S. Çetin, M. Yanik, T. Temel, and I. Çiçekli, "TURKSENT: A sentiment annotation tool for social media," in *Proc. 7th Linguistic Annotation Workshop Interoperability Discourse*, 2013, pp. 131–134.

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

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

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

[14] S. C. Deerwester *et al.*, "Computer information retrieval using latent semantic structure," U.S. Patent 4 839 853 A, Jun. 13, 1989.

[15] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[16] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1757–1772, Jul. 2013.

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

[18] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: An overview," *Prog. Artif. Intell.*, vol. 1, no. 1, pp. 89–101, 2012.

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

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

[21] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *Proc. 15th ACM Int. Conf. Multimedia*, 2007, pp. 17–26.

[22] S. Džeroski, D. Demšar, and J. Grbović, "Predicting chemical parameters of river water quality from bioindicator data," *Appl. Intell.*, vol. 13, no. 1, pp. 7–17, 2000.

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

[24] 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," *Sci. Rep.*, vol. 7, Jan. 2017, Art. no. 40652.

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

[26] X. Geng, K. Smith-Miles, and Z. Zhou, "Facial age estimation by learning from label distributions," in *Proc. AAAI*, 2010, pp. 451–456.

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

[28] K. Crammer and Y. Singer, "A family of additive online algorithms for category ranking," *J. Mach. Learn. Res.*, vol. 3, pp. 1025–1058, Mar. 2003.

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

[30] 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 *Proc. ICCV*, 2009, pp. 2098–2105.

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

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

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

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

[35] B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, "Max-margin parsing," in *Proc. EMNLP*, 2004, pp. 1–8.

[36] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.

[37] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21–36, May 2003.

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

[39] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000.

[40] C. H. Lau, Y. Li, and D. Tjondronegoro, "Microblog retrieval using topical features and query expansion," in *Proc. TREC*, 2011, pp. 1–6.

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

[42] M. K. Choong, M. Charbit, and H. Yan, "Autoregressive-model-based missing value estimation for DNA microarray time series data," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 1, pp. 131–137, Jan. 2009.

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

[44] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

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

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

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

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

[49] J. Gauthier, "Conditional generative adversarial nets for convolutional face generation," *Class Project Stanford CS231N, Convolutional Neural Netw. Vis. Recognit.*, vol. 2014, no. 5, p. 2, 2014.

[50] J. Johnson *et al.*, "Image retrieval using scene graphs," in *Proc. CVPR*, 2015, pp. 3668–3678.

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

[52] R. Krishna *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, 2017.

[53] D. Zhang, H. Fu, J. Han, A. Borji, and X. Li, "A review of co-saliency detection technique: Fundamentals, applications, and challenges," Apr. 2016, *arXiv:1604.07090*. [Online]. Available: https://arxiv.org/abs/1604.07090

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

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

[56] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, Sep. 2005.

[57] H. Liu, J. Cai, and Y.-S. Ong, "Remarks on multi-output Gaussian process regression," *Knowl.-Based Syst.*, vol. 144, pp. 102–121, Mar. 2018.

[58] X. Geng, "Label distribution learning," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1734–1748, Jul. 2016.

[59] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, "Supervised learning of semantic classes for image annotation and retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 394–410, Mar. 2007.

[60] A. S. Weigend, *Time Series Prediction: Forecasting The Future And Understanding The Past*. Evanston, IL, USA: Routledge, 2018.

[61] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2927–2936.

[62] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[63] S. Thrun and J. O'Sullivan, "Clustering learning tasks and the selective cross-task transfer of knowledge," in *Learning to Learn*. Boston, MA, USA: Springer, 1998, pp. 235–257.

[64] Q. Mao, I. W.-H. Tsang, and S. Gao, "Objective-guided image annotation," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1585–1597, Apr. 2013.

[65] M. A. Tahir, J. Kittler, and F. Yan, "Inverse random under sampling for class imbalance problem and its application to multi-label classification," *Pattern Recognit.*, vol. 45, no. 10, pp. 3738–3750, 2012.

[66] R. Alejo, V. García, and J. H. Pacheco-Sánchez, "An efficient over-sampling approach based on mean square error back-propagation for dealing with the multi-class imbalance problem," *Neural Process. Lett.*, vol. 42, no. 3, pp. 603–617, 2015, doi: 10.1007/s11063-014-9376-3.

[67] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[68] I. O. Tolstikhin, S. Gelly, O. Bousquet, C.-J. Simon-Gabriel, and B. Schölkopf, "AdaGAN: Boosting generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5424–5433.

[69] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.

[70] V. Khrulkov and I. Oseledets, "Geometry score: A method for comparing generative adversarial networks," Feb. 2018, *arXiv: 1802.02664*. [Online]. Available: https://arxiv.org/abs/1802.02664

[71] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[72] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li, "Mode regularized generative adversarial networks," Dec. 2016, *arXiv:1612.02136*. [Online]. Available: https://arxiv.org/abs/1612.02136

[73] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.

[74] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," Jan. 2018, *arXiv:1801.01401*. [Online]. Available: https://arxiv.org/abs/1801.01401

[75] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are GANs created equal? A large-scale study," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 698–707.

[76] J. J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 9th ACM Conf. Recommender Syst.*, 2013, pp. 165–172.

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

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

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

[80] A. Zubiaga, "Enhancing navigation on wikipedia with social tags," Feb. 2012, *arXiv:1202.5469*. [Online]. Available: https://arxiv.org/abs/1202.5469

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

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

[83] I. Partalas *et al.*, "LSHTC: A benchmark for large-scale text classification," Mar. 2015, *arXiv:1503.08581*. [Online]. Available: https://arxiv.org/abs/1503.08581

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

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

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

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

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

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

[90] G. Marques, M. A. Domingues, T. Langlois, and F. Gouyon, "Three current issues in music autotagging," in *Proc. Int. Soc. Music Inf. Retr. Conf.*, 2011, pp. 795–800.

[91] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-UCSD birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[92] V. Jouhet *et al.*, "Automated classification of free-text pathology reports for registration of incident cases of cancer," *Methods Inf. Med.*, vol. 51, no. 3, pp. 242–251, 2012.

[93] A. E. W. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," *Sci. Data*, vol. 3, May 2016, Art. no. 160035.

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

[95] *Kaggle Data Set ECML/PKDD 15: Taxi Trajectory Prediction (1)*, ECML/PKDD, Porto, Portugal, 2015.

[96] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. (Feb. 2009). *CRAWDAD Data Set EPFL/Mobility (v. 2009-02-24)*. [Online]. Available: http://crawdad.org/epfl/mobility/

[97] A. Trindade. (2016). *UCI Maching Learning Repository-Electricityloaddiagrams20112014 Data Set*. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[98] S. Oh *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. CVPR*, 2011, pp. 3153–3160.

[99] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[100] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009, vol. 1, no. 4, p. 7.

[101] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[102] M. Mahoney. (2011). *Large Text Compression Benchmark*. [Online]. Available: http://www.mattmahoney.net/text/text.html

[103] R. Zafarani and H. Liu. (2009). *Social Computing Data Repository at ASU*. [Online]. Available: http://socialcomputing.asu.edu

[104] J. Leskovec and A. Krevl. (Jun. 2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. [Online]. Available: http://snap.stanford.edu/data

[105] E. F. T. K. Sang and S. Buchholz, "Introduction to the CoNLL-2000 shared task: Chunking," in *Proc. 2nd Workshop Learn. Lang. Log. 4th Conf. Comput. Natural Lang. Learn.*, 2000, pp. 127–132.

[106] D. Zhou, S. Basu, Y. Mao, and J. C. Platt, "Learning from the wisdom of crowds by minimax entropy," in *Proc. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2204–2212.

[107] K. Lee, X. He, L. Zhang, and L. Yang, "CleanNet: Transfer learning for scalable image classifier training with label noise," in *Proc. CVPR*, 2018, pp. 5447–5456.

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

[109] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, "Webvision database: Visual learning and understanding from Web data," Aug. 2017, *arXiv:1708.02862*. [Online]. Available: https://arxiv.org/abs/1708.02862

[110] B. Thomee *et al.*, "YFCC100M: The new data in multimedia research," Mar. 2015, *arXiv:1503.01817*. [Online]. Available: https://arxiv.org/abs/1503.01817

[111] C. Kümmerle and J. Sigl, "Harmonic mean iteratively reweighted least squares for low-rank matrix recovery," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 1815–1863, 2018.

[112] W. Liu and I. W. Tsang, "Making decision trees feasible in ultra-high feature and label dimensions," *J. Mach. Learn. Res.*, vol. 18, pp. 81:1–81:36, Jan. 2017.

[113] C. Dupuy and F. Bach, "Online but accurate inference for latent variable models with local gibbs sampling," *J. Mach. Learn. Res.*, vol. 18, no. 126, pp. 126:1–126:45, 2017.

[114] W. Liu, I. W. Tsang, and K. Müller, "An easy-to-hard learning paradigm for multiple classes and multiple labels," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 3300–3337, 2017.

[115] C. Brouard, M. Szafranski, and F. D'Alché-Buc, "Input output kernel regression: Supervised and semi-supervised structured output prediction with operator-valued kernels," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 6105–6152, 2016.

[116] 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," *J. Mach. Learn. Res.*, vol. 17, pp. 107:1–107:31, Jan. 2016.

[117] R. Babbar, I. Partalas, E. Gaussier, M.-R. Amini, and C. Amblard, "Learning taxonomy adaptation in large-scale classification," *J. Mach. Learn. Res.*, vol. 17, pp. 98:1–98:37, Feb. 2016.

[118] X. Li, T. Zhao, X. Yuan, and H. Liu, "The flare package for high dimensional linear regression and precision matrix estimation in R," *J. Mach. Learn. Res.*, vol. 16, pp. 553–557, Mar. 2015.

[119] F. Han, H. Lu, and H. Liu, "A direct estimation of high dimensional stationary vector autoregressions," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 3115–3150, Jan. 2015.

[120] J. R. Doppa, A. Fern, and P. Tadepalli, "Structured prediction via output space search," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1317–1350, 2014.

[121] D. Colombo and M. H. Maathuis, "Order-independent constraint-based causal structure learning," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3741–3782, 2014.

[122] C. Gentile and F. Orabona, "On multilabel classification and ranking with bandit feedback," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2451–2487, 2014.

[123] P. Gong, J. Ye, and C. Zhang, "Multi-stage multi-task feature learning," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2979–3010, 2013.

[124] A. Talwalkar, S. Kumar, M. Mohri, and H. Rowley, "Large-scale SVD and manifold learning," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 3129–3152, 2013.

[125] K. Fu, J. Li, J. Jin, and C. Zhang, "Image-text surgery: Efficient concept learning in image captioning by generating pseudopairs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 5910–5921, Apr. 2018.

[126] É. Protas, J. D. Bratti, J. F. O. Gaya, P. Drews, and S. S. C. Botelho, "Visualization methods for image transformation convolutional neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2231–2243, Jul. 2019.

[127] H. Zhang, S. Wang, X. Xu, T. W. S. Chow, and Q. M. J. Wu, "Tree2Vector: Learning a vectorial representation for tree-structured data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5304–5318, Nov. 2018.

[128] Z. Lin, G. Ding, J. Han, and L. Shao, "End-to-end feature-aware label space encoding for multilabel classification with many classes," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2472–2487, Jun. 2018.

[129] S. Fang, J. Li, Y. Tian, T. Huang, and X. Chen, "Learning discriminative subspaces on random contrasts for image saliency analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1095–1108, May 2017.

[130] K. Zhang, D. Tao, X. Gao, X. Li, and J. Li, "Coarse-to-fine learning for single-image super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1109–1122, May 2017.

[131] M. Kim, "Mixtures of conditional random fields for improved structured output prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1233–1240, May 2017.

[132] Y. Cheung, M. Li, Q. Peng, and C. L. P. Chen, "A cooperative learning-based clustering approach to lip segmentation without knowing segment number," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 1, pp. 80–93, Jan. 2017.

[133] L. Wang, L. Liu, and L. Zhou, "A graph-embedding approach to hierarchical visual word mergence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 308–320, Feb. 2017.

[134] Z. Li, Z. Lai, Y. Xu, J. Yang, and D. Zhang, "A locality-constrained and label embedding dictionary learning algorithm for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 278–293, Feb. 2015.

[135] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q.-S. Zhu, "A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 579–592, Mar. 2016.

[136] 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," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2472–2485, Dec. 2016.

[137] A. Alush and J. Goldberger, "Hierarchical image segmentation using correlation clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1358–1367, Jun. 2016.

[138] D. Tao, J. Cheng, M. Song, and X. Lin, "Manifold ranking-based matrix factorization for saliency detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1122–1134, Jun. 2016.

[139] F. Cao, M. Cai, Y. Tan, and J. Zhao, "Image super-resolution via adaptive $\ell_p(0<p<1)$ regularization and sparse representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1550–1561, Jul. 2016.

[140] Q. Zhu, L. Shao, X. Li, and L. Wang, "Targeting accurate object extraction from an image: A comprehensive study of natural image matting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 185–207, Feb. 2015.

[141] Y. Chen, Y. Ma, D. H. Kim, and S.-K. Park, "Region-based object recognition by color segmentation using a simplified PCNN," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1682–1697, Aug. 2015.

[142] M. Li, W. Bi, J. T. Kwok, and B.-L. Lu, "Large-scale Nyström kernel matrix approximation using randomized SVD," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 152–164, Jan. 2015.

[143] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, "A unified learning framework for single image super-resolution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 4, pp. 780–792, Apr. 2014.

[144] A. Bauer, N. Görnitz, F. Biegler, K.-R. Müller, and M. Kloft, "Efficient algorithms for exact inference in sequence labeling SVMs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 870–881, May 2014.

[145] K. Tang, R. Liu, Z. Su, and J. Zhang, "Structure-constrained low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2167–2179, Dec. 2014.

[146] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, "Low-rank structure learning via nonconvex heuristic recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 383–396, Mar. 2013.

[147] H. Zhang, Q. M. J. Wu, and T. M. Nguyen, "Incorporating mean template into finite mixture model for image segmentation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 2, pp. 328–335, Feb. 2013.

[148] Y. Pang, Z. Ji, P. Jing, and X. Li, "Ranking graph embedding for learning to rerank," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 8, pp. 1292–1303, Aug. 2013.

[149] Y. Luo, D. Tao, C. Xu, C. Xu, H. Liu, and Y. Wen, "Multiview vector-valued manifold regularization for multilabel image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 709–722, May 2013.

[150] B. Zhang, D. Xiong, and J. Su, "Neural machine translation with deep attention," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[151] S. Jeong, J. Lee, B. Kim, Y. Kim, and J. Noh, "Object segmentation ensuring consistency across multi-viewpoint images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2455–2468, Oct. 2018.

[152] C. Raposo, M. Antunes, and J. P. Barreto, "Piecewise-planar stereoscan: Sequential structure and motion using plane primitives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1918–1931, Aug. 2018.

[153] M. Cordts, T. Rehfeld, M. Enzweiler, U. Franke, and S. Roth, "Tree-structured models for efficient multi-cue scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1444–1454, Jul. 2017.

[154] Y. Xu, E. Carlinet, T. Géraud, and L. Najman, "Hierarchical segmentation using tree-based shape spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 457–469, Mar. 2017.

[155] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2321–2334, Dec. 2017.

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

[157] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[158] Z. Qin and C. R. Shelton, "Social grouping for multi-target tracking and head pose estimation in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2082–2095, Oct. 2016.

[159] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1792–1805, Sep. 2015.

[160] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. Pérez, "Sparse multi-view consistency for object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1890–1903, Sep. 2015.

[161] S. Wang, Y. Wei, K. Long, X. Zeng, and M. Zheng, "Image super-resolution via self-similarity learning and conformal sparse representation," *IEEE Access*, vol. 6, pp. 68277–68287, 2018.

[162] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1614–1627, Aug. 2014.

[163] N. Zhou and J. Fan, "Jointly learning visually correlated dictionaries for large-scale visual recognition applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 715–730, Apr. 2014.

[164] P. Perakis, G. Passalis, T. Theoharis, and I. A. Kakadiaris, "3D facial landmark detection under large yaw and expression variations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1552–1564, Jul. 2013.

[165] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

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

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

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

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

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

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

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

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

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

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

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

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

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

[179] 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 *Proc. NIPS*, 2018, pp. 6358–6368.

[180] B. Pan *et al.*, "MacNet: Transferring knowledge from machine comprehension to sequence-to-sequence models," in *Proc. NIPS*, 2018, pp. 6095–6105.

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

[182] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, "MaskRNN: Instance level video object segmentation," in *Proc. NIPS*, 2017, pp. 324–333.

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

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

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

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

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

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

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

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

[191] J. Hoffman *et al.*, "LSDA: Large scale detection through adaptation," in *Proc. NIPS*, 2014, pp. 3536–3544.

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

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

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

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

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

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

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

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

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

[201] 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 *Proc. ICML*, 2016, pp. 3069–3077.

[202] M. Cissé, M. Al-Shedivat, and S. Bengio, "ADIOS: Architectures deep in output space," in *Proc. ICML*, 2016, pp. 2770–2779.

[203] D. Park, J. Neeman, J. Zhang, S. Sanghavi, and I. S. Dhillon, "Preference completion: Large-scale collaborative ranking from pairwise comparisons," in *Proc. ICML*, 2015, pp. 1907–1916.

[204] D. Hernández-Lobato, J. M. Hernández-Lobato, and Z. Ghahramani, "A probabilistic model for dirty multi-task feature selection," in *Proc. ICML*, 2015, pp. 1073–1082.

[205] 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 *Proc. ICML*, 2015, pp. 720–729.

[206] H. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale multi-label learning with missing labels," in *Proc. ICML*, 2014, pp. 593–601.

[207] Z. Lin, G. Ding, M. Hu, and J. Wang, "Multi-label classification via feature-aware implicit label space encoding," in *Proc. ICML*, 2014, pp. 325–333.

[208] Y. Li and R. Zemel, "High order regularization for semi-supervised learning of structured output problems," in *Proc. ICML*, 2014, pp. 1368–1376.

[209] W. Bi and J. Kwok, "Efficient multi-label classification with many labels," in *Proc. ICML*, 2013, pp. 405–413.

[210] R. Takhanov and V. Kolmogorov, "Inference algorithms for pattern-based CRFs on sequence data," in *Proc. ICML*, 2013, pp. 145–153.

[211] M. Xiao and Y. Guo, "Domain adaptation for sequence labeling tasks with a probabilistic language adaptation model," in *Proc. ICML*, 2013, pp. 293–301.

[212] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," *Data Mining Knowl. Discovery*, vol. 6, no. 2, pp. 153–172, 2002.

[213] Y. Zhai, Y.-S. Ong, and I. W. Tsang, "The emerging 'big dimensionality,"' *IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 14–26, Aug. 2014.

[214] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proc. NIPS*, 2016, pp. 217–225.

[215] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. ICCV*, 2017, pp. 5908–5916.

[216] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," Feb. 2018, *arXiv: 1802.09178*. [Online]. Available: https://arxiv.org/abs/1802.09178

[217] W. Fedus, I. J. Goodfellow, and A. M. Dai, "MaskGAN: Better text generation via filling in the _____," Jan. 2018, *arXiv:1801.07736*. [Online]. Available: https://arxiv.org/abs/1801.07736

[218] Y.-N. Chen and H.-T. Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Proc. NIPS*, 2012, pp. 1538–1546.

[219] D. J. Hsu, S. M. Kakade, J. Langford, and T. Zhang, "Multi-label prediction via compressed sensing," in *Proc. NIPS*, 2009, pp. 772–780.

[220] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Comput.*, vol. 24, no. 9, pp. 2508–2542, 2012.

[221] A. Kapoor, R. Viswanathan, and P. Jain, "Multilabel classification using Bayesian compressed sensing," in *Proc. NIPS*, 2012, pp. 2654–2662.

[222] P. Mineiro and N. Karampatziakis, "Fast label embeddings via randomized linear algebra," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2015, pp. 37–51.

[223] Y. Jernite, A. Choromanska, and D. Sontag, "Simultaneous learning of trees and representations for extreme classification and density estimation," in *Proc. ICML*, 2017, pp. 1665–1674.

[224] J. Liu, W. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proc. SIGIR*, 2017, pp. 115–124.

[225] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

[226] O. Maimon and L. Rokach, Eds., *Data Mining and Knowledge Discovery Handbook*, 2nd ed. Springer, 2010.

[227] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. ECML PKDD*, 2009, pp. 254–269.

[228] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *J. Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.

[229] J. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, pp. 282–289.

[230] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Proc. ECML*, 2007, pp. 406–417.

[231] K. Dembczynski, W. Cheng, and E. Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Proc. ICML*, 2010, pp. 279–286.

[232] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, Oct. 2009.

[233] S. Baker and A. Korhonen, "Initializing neural networks for hierarchical multi-label text classification," in *Proc. BioNLP*, 2017, pp. 307–315.

[234] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, "Using ranking-CNN for age estimation," in *Proc. CVPR*, 2017, pp. 742–751.

[235] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, vol. 27, Dec. 2014, pp. 3104–3112.

[236] C. Smith and Y. Jin, "Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction," *Neurocomputing*, vol. 143, pp. 302–311, Nov. 2014.

[237] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. 41st Annu. Meeting Assoc. Comput. Linguistics-Assoc. Comput. Linguistics*, vol. 1, 2003, pp. 160–167.

[238] W. Gao and Z.-H. Zhou, "On the consistency of multi-label learning," in *Proc. 24th Annu. Conf. Learn. Theory*, 2011, pp. 341–358.

[239] A. Tewari and P. L. Bartlett, "On the consistency of multiclass classification methods," *J. Mach. Learn. Res.*, vol. 8, pp. 1007–1025, May 2007.

[240] J. Keshet and D. A. McAllester, "Generalization bounds and consistency for latent structural probit and ramp loss," in *Proc. NIPS*, 2011, pp. 2205–2212.

[241] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," in *Proc. NIPS*, 2003, pp. 25–32.

[242] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *Proc. SIGIR*, 2007, pp. 271–278.

[243] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," in *Proc. ACL-Conference Empirical Methods Natural Lang. Process.-Assoc. Comput. Linguistics*, vol. 10, 2002, pp. 1–8.

[244] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. ICASSP*, 2008, pp. 4057–4060.

[245] K. Gimpel and N. A. Smith, "Softmax-margin CRFs: Training log-linear models with cost functions," in *Proc. HLT-NAACL*, 2010, pp. 733–736.

[246] W. Liu, D. Xu, I. W. Tsang, and W. Zhang, "Metric learning for multi-output tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 408–422, Feb. 2019, doi: 10.1109/TPAMI.2018.2794976.

[247] J. Deng, S. Satheesh, A. C. Berg, and F. Li, "Fast and balanced: Efficient label tree learning for large scale object recognition," in *Proc. NIPS*, 2011, pp. 567–575.

[248] T. Gao and D. Koller, "Discriminative learning of relaxed hierarchy for large-scale visual recognition," in *Proc. ICCV*, 2011, pp. 2072–2079.

[249] X. Shen, W. Liu, I. W. Tsang, Q. S. Sun, and Y. S. Ong, "Compact multi-label learning," in *Proc. AAAI*, 2018, pp. 4066–4073.

[250] X. Shen, W. Liu, I. W. Tsang, Q.-S. Sun, and Y.-S. Ong, "Multilabel prediction via cross-view search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4324–4338, Sep. 2017.

[251] X. Shen, W. Liu, Y. Luo, Y.-S. Ong, and I. W. Tsang, "Deep discrete prototype multilabel learning," in *Proc. IJCAI*, 2018, pp. 2675–2681.

[252] A. More, "Survey of resampling techniques for improving classification performance in unbalanced datasets," Aug. 2016, *arXiv: 1608.06048*. [Online]. Available: https://arxiv.org/abs/1608.06048

[253] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. Workshop Learn. From Imbalanced Datasets*, vol. 126, 2003, pp. 1–7.

[254] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.

[255] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," Apr. 2018, *arXiv:1804.10851*. [Online]. Available: https://arxiv.org/abs/1804.10851

[256] M. Rezaei, H. Yang, and C. Meinel, "Multi-task generative adversarial network for handling imbalanced clinical data," Nov. 2018, *arXiv: 1811.10419*. [Online]. Available: https://arxiv.org/abs/1811.10419

[257] E. Montahaei, M. Ghorbani, M. S. Baghshah, and H. R. Rabiee, "Adversarial classifier for imbalanced problems," Nov. 2018, *arXiv: 1811.08812*. [Online]. Available: https://arxiv.org/abs/1811.08812

[258] B. Romera-Paredes and P. H. S. Torr, "An embarrassingly simple approach to zero-shot learning," in *Proc. ICML*, 2015, pp. 2152–2161.

[259] A. Gaure, A. Gupta, V. K. Verma, and P. Rai, "A probabilistic framework for zero-shot multi-label learning," in *Proc. Conf. Uncertainty Artif. Intell. (UAI)*, vol. 1, 2017, p. 3.

[260] A. Rios and R. Kavuluru, "Few-shot and zero-shot multi-label learning for structured label spaces," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3132–3142.

[261] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. F. Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *Proc. CVPR*, 2018, pp. 1576–1585.

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

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

[264] Z. Zhang, C. Wang, B. Xiao, W. Zhou, and S. Liu, "Robust relative attributes for human action recognition," *Pattern Anal. Appl.*, vol. 18, no. 1, pp. 157–171, 2015.

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

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

[267] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2317–2324, Nov. 2014.

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

[269] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. CVPR*, 2016, pp. 1563–1572.

[270] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *Proc. ICCV*, 2017, pp. 1928–1936.

[271] G. Chen, Y. Song, F. Wang, and C. Zhang, "Semi-supervised multi-label learning by solving a Sylvester equation," in *Proc. ICDM*, 2008, pp. 410–419.

[272] Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou, "Multi-label learning with weak label," in *Proc. AAAI*, 2010, pp. 1–6.

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

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

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

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

[277] C. Gong, H. Zhang, J. Yang, and D. Tao, "Learning with inadequate and incorrect supervision," in *Proc. ICDM*, 2017, pp. 889–894.

[278] R. Barandela and E. Gasca, "Decontamination of training samples for supervised pattern recognition methods," in *Proc. Joint IAPR Int. Workshops (SPR SSPR)*, 2000, pp. 621–630.

[279] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *J. Artif. Intell. Res.*, vol. 11, no. 1, pp. 131–167, 1999.

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

[281] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," Dec. 2014, *arXiv:1412.6596*. [Online]. Available: https://arxiv.org/abs/1412.6596

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

[283] V. Mnih and G. E. Hinton, "Learning to label aerial images from noisy data," in *Proc. ICML*, 2012, pp. 1–8.

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

[285] J. Yao *et al.*, "Deep learning from noisy image labels with quality embedding," Nov. 2017, *arXiv:1711.00583*. [Online]. Available: https://arxiv.org/abs/1711.00583

[286] Y. Mao, G. Cheung, C.-W. Lin, and Y. Ji, "Joint learning of similarity graph and image classifier from partial labels," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2016, pp. 1–4.

[287] J. Chai, I. W. Tsang, and W. Chen, "Large margin partial label machine," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[288] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, and D. Tao, "A regularization approach for instance-based superset label learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 967–978, Mar. 2018.

[289] F. Yu and M.-L. Zhang, "Maximum margin partial label learning," *Mach. Learn.*, vol. 106, no. 4, pp. 573–593, Apr. 2017.

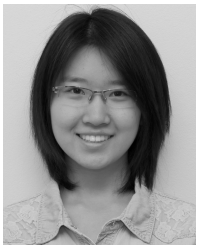
[290] M.-L. Zhang, F. Yu, and C.-Z. Tang, "Disambiguation-free partial label learning," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2155–2167, Oct. 2017.

[291] M.-K. Xie and S.-J. Huang, "Partial multi-label learning," in *Proc. AAAI*, 2018, pp. 4302–4309.

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

[293] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[294] W. Qu, Y. Zhang, J. Zhu, and Q. Qiu, "Mining multi-label concept-drifting data streams using dynamic classifier ensemble," in *Proc. ACML*, 2009, pp. 308–321.
[295] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," in *Proc. KDD*, 2009, pp. 139–148.
[296] X. Kong and P. S. Yu, "An ensemble-based approach to fast classification of multi-label data streams," in *Proc. 7th Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*, 2011, pp. 95–104.
[297] A. Büyükçakir, H. Bonab, and F. Can, "A novel online stacked ensemble for multi-label stream classification," in *Proc. CIKM*, 2018, pp. 1063–1072.
[298] A. Milan, S. H. Rezatofighi, A. R. Dick, I. D. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. AAAI*, 2017, pp. 4225–4232.
[299] J. Read, A. Bifet, G. Holmes, and B. Pfahringer, "Scalable and efficient multi-label classification for evolving data streams," *Mach. Learn.*, vol. 88, nos. 1–2, pp. 243–272, 2012.
[300] L. Huang, Q. Yang, and W. Zheng, "Online hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2309–2322, Jun. 2018.
[301] D. Xu, I. W. Tsang, and Y. Zhang, "Online product quantization," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 11, pp. 2185–2198, Nov. 2018.
[302] Y. Wang *et al.*, "Iterative learning with open-set noisy labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8688–8696.
[303] O. Anava, E. Hazan, and A. Zeevi, "Online time series prediction with missing data," in *Proc. ICML*, 2015, pp. 2191–2199.
[304] A. Bendale and T. E. Boult, "Towards open world recognition," in *Proc. CVPR*, 2015, pp. 1893–1902.
[305] E. S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas, "Dealing with concept drift and class imbalance in multi-label stream classification," in *Proc. IJCAI*, 2011, pp. 1583–1588.
[306] Z. Ren, M. Peetz, S. Liang, W. van Dolen, and M. de Rijke, "Hierarchical multi-label classification of social text streams," in *Proc. SIGIR*, 2014, pp. 213–222.
[307] Y. Shi, D. Xu, Y. Pan, I. W. Tsang, and S. Pan, "Label embedding with partial heterogeneous contexts," in *Proc. AAAI*, 2019, pp. 4926–4933.
[308] A. Mousavian, H. Pirsiavash, and J. Košecká, "Joint semantic segmentation and depth estimation with deep convolutional networks," in *Proc. 3DV*, 2016, pp. 611–619.
[309] W. Van Ranst, F. De Smedt, J. Berte, and T. Goedemé, "Fast simultaneous people detection and re-identification in a single shot network," in *Proc. AVSS*, 2018, pp. 1–6.

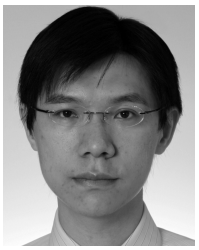


Donna Xu received the B.C.S.T. degree (Hons.) in computer science from The University of Sydney, Sydney, NSW, Australia, in 2014, and the Ph.D. degree from the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia.

Her current research interests include multi-class classification, online hashing, and information retrieval.

Yaxin Shi received the M.E. degree in computer science from the Ocean University of China, Qingdao, China, in 2017. She is currently pursuing the Ph.D. degree under the supervision of Prof. I. W. Tsang with the Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW, Australia.

Her current research interests include including multi-view learning, structure learning, and deep generative networks.



Ivor W. Tsang received the Ph.D. degree from The Hong Kong University of Science and Technology, Hong Kong, in 2007.

He is currently a Professor with the University of Technology Sydney, Ultimo, NSW, Australia, where he is also the Research Director of the UTS Priority Research Centre for Artificial Intelligence.

Dr. Tsang was conferred the 2008 Natural Science Award (Class II), the Australian Research Council Future Fellowship in 2013, the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding 2004 Paper Award in 2007, the 2014 IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award, and the Best Student Paper Award at CVPR 2010. He serves as an Associate Editor for the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, the IEEE TRANSACTIONS ON BIG DATA, and *Neurocomputing*. He also serves as an Area Chair/SPC for NeurIPS, AAAI, and IJCAI.

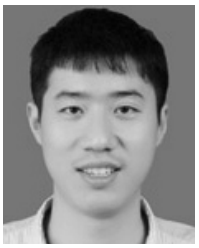
Yew-Soon Ong received the Ph.D. degree from the University of Southampton, Southampton, U.K., in 2003, for his work on artificial intelligence in complex design.

He is currently a President Chair Professor of computer science with Nanyang Technological University (NTU), Singapore, where he also serves as the Director of the Singtel-NTU Cognitive & Artificial Intelligence Joint Laboratory. He is also the Chief Artificial Intelligence Scientist with the Agency for Science, Technology and Research, Singapore. His current research interests include artificial and computational intelligence.

Dr. Ong is also the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE and an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, and so on.

Chen Gong (M'16) is currently a Professor with the Nanjing University of Science and Technology, Nanjing, China. He has published more than 70 technical articles at prominent journals and conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), NeurIPS, CVPR, AAAI, IJCAI, and ICDM. His current research interests include machine learning, data mining, and learning-based vision problems.

He received the "Excellent Doctorial Dissertation" awarded by SJTU and the Chinese Association for Artificial Intelligence. He was also enrolled by the "Young Elite Scientists Sponsorship Program" of Jiangsu Province and the China Association for Science and Technology.



Xiaobo Shen received the B.Sc. and Ph.D. degrees from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2011 and 2017, respectively.

He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. He has authored over 30 technical articles in prominent journals and conferences, such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CYBERNETICS, the Conference on Neural Information Processing Systems, the International Conference on Machine Learning, ACM MM, AAAI, and IJCAI. His current research interests include multi-view learning, multi-label learning, network embedding, and hashing.