Explaining the Performance of Multi-label Classification Methods with Data Set Properties

Jasmin Bogatinovski[1,2,3✉], Ljupčo Todorovski[1,4], Sašo Džeroski[1,2], Dragi Kocev[1,2,5✉]
[1]*Jožef Stefan Institute, Ljubljana, Slovenia*
[2]*Jožef Stefan IPSchool, Ljubljana, Slovenia*
[3]*Dept. of Distributed Operating Systems, TU Berlin, Germany*
[4]*Faculty of Mathematics and Physics, University of Ljubljana, Slovenia*
[5]*Bias Variance Labs, Ljubljana, Slovenia*
{name.surname}@{tu-berlin.de} {ijs.si}

June 17, 2021

1 Datasets

ABPM [59] is a dataset with 33 features and 6 labels. The dataset has 270 records of blood pressure measurements from patients in a duration of 24 hours. The features represent general information for the patients as gender, age, weight, height, but also various statistical features obtained from the diastolic and systolic load values. The labels represent the presence and absence of validity, morning surge, blood pressure load, blood pressure variability, pulse pressure and circadian rhythm.

Foodtruck [60] is a dataset obtained from a survey conducted with 400 subjects. It represents the personal preferences of the people when ordering food from food trucks. The attributes represent objective questions about users' profile and their habits and preferences related to food trucks. To use the nominal features they are encoded as one hot vector. Some examples of the features are hygiene, taste, expenses, day period of preference, takeout option, gender, age group, etc. The labels are the 12 food types offered: Arabic, fitness, Brazilian, Japanese, gourmet, Mexican, Chinese, healthy, snacks, street, Italian and sweets desserts.

Flags [61] is a dataset that contains information about the countries' flags. Its features describe the presence or absence of different symbols appearing on flags, such as triangles, information about population, language, religion, etc. The labels represent the 7 colours: red, green, blue, yellow, white, black, orange that appear on flags.

CHD49 [62] is a dataset from the medical domain. It describes the problem of diagnosing coronary heart disease via traditional Chinese medicine approaches. The features represent the presence or absence of different symptoms accessed via feelings of cold or warm, sweating, head, body, chest, urine, etc. The labels represent the 6 commonly-used patterns, including deficiency of heart qi syndrome, deficiency of heart yang syndrome, deficiency of heart yin syndrome, qi stagnation syndrome, turbid phlegm syndrome, and blood stasis syndrome.

Water quality [63] is a dataset containing descriptions about the biological properties of a river. The features represent the different concentration of chemical components, such as biological oxygen demand, electrical conductivity, chemical oxygen demand, concentrations of different elements and compounds, water temperature and total hardness. The labels are represented by 14 taxa present at the sampling sites and their density. This dataset belongs to the domain of chemistry.

Emotions [64] is a dataset from the multimedia domain. It describes the relationship between music and emotions based on the Tellegen-Watson-Clarks model of mood. The obtained sound signals are used to calculate temporal and timber features. The labels represent 6 main emotions a music piece provides:

amazed-surprised, happy-pleased, relaxing-calm, quite-still, sad-lonely, and angry-fearful.

GO [65] datasets are from the area of bioinformatics. They describe the task of predicting sub-cellular locations of proteins in different organisms according to their protein sequences. The software "BLAST" (Basic local alignment search tool) is applied to each protein sequence from the Swiss-Prot database. The homologous proteins that have great pairwise sequence identity are collected into sets with accession numbers. Each of those accession numbers is matched with the Gene Ontology (GO) database. Then a binary feature vector is constructed such that if GO number is present, the element in the vector representing the protein has the value of 1, otherwise 0. The labels are the sub-cellular locations where a protein may appear. These numbers are different depending on the organism at interest. There are 5 datasets constructed in this fashion: HumanGO, VirusGO, PlantGO, GnegativeGO and GpositiveGO.

PseAAC [65] datasets are also from the area of bioinformatics. They describe the task of predicting sub-cellular locations of proteins in organism according to their sequences. However, compared to GO datasets it represents the protein samples using pseudo amino acid composition including 20 amino-acid, 20 pseudo-amino acid and 400 dipeptide components. The labels are the subcellular locations where a protein may appear. These numbers are different depending on the organism at interest. There are 5 datasets constructed in this fashion: HumanPseAAC, VirusPseAAC, PlantPseAAC, GnegativePseAAC and GpositivePseAAC.

Yeast [2] is a dataset from the domain of biology. The data represent micro-array expressions and phylogeny profiles of genes. The labels can be multiple of the following functional groups: metabolism, energy, transcription, protein synthesis, protein destination, cell growth, transport facilitation, cell transport, cellular biogenesis, ionic homeostasis, cellular organization, transportable elements, cell death and ageing, and cell communication. So, the task is to predict the function of a gene using its micro-array expression.

Birds [66] is a dataset representing the problem of bird species classification from acoustic recordings. In one recording multiple species may appear. After obtaining the raw audio signals, the signals are filtered and segmented. From each of the segments, various features from time and frequency domain are extracted. The labels represent if a type of bird is present in the particular instance. It belongs to the multimedia domain in the subcategory audio.

Scene [67] is one of the most popular datasets from the multimedia domain, belonging to the subcategory of images. It provides a very intuitive way of depicting the aim of MLC. The dataset is about the classification of different scenes on an image. There are a total of 6 labels beach, sunset, fall foliage, field, mountain and urban. The images are described with 294 features derived from LUV space.

Cal500 [68] is a dataset from the multimedia domain, from the subcategory of audio. Each feature is calculated by analyzing a short-time series of the audio signal using various time-series generated features from the audio signal, obtained by human annotators. The targets represent various aspects of music composition such as the emotional level of the song, the music genre, the instruments present in the recording, etc.

Genbase [69] is a dataset that contains protein sequences and its functional family labels. Since a protein sequence can have multiple functions, the problem can be defined as a MLC task. Each protein sequence is mapped to an attribute vector. Since each protein sequence contains some motifs, thus it can be represented as a set of 1's and 0's depending on the presence or absence of the motif in the sequence. The labels are grouped in the 10 most common families. The labels are the classes: oxidoreductases, isomerases, cytokines and growth factors, structural proteins, receptors, DNA or RNA associated proteins, transferals, protein secretion and chaperoned, hydrolysis. GenMiner is used as a tool to prepare the data.

Yelp [70] is a dataset from the text domain. It is concerned with the classification of reviews from customers for restaurants into relevant categories. There are two groups of features, star ratings (represented by binary variables) and textual features consisting of unigrams, bigrams and trigrams. The textual features are extracted in such a way that after downcasing all the words and removing special characters the unigrams, bigrams and trigrams are extracted and their frequency among reviews is recorded. Only the ones that have their frequency above the threshold of 300 are preserved. The labels

represent the abstractions the review refers to. The meaning of the label-sets is a multiple of Food, Service, Ambiance, Deals/Discounts, Worthiness.

Medical [71] is a dataset composed of medical records, thus belongs to the group of text domain. The features are BoW representation of the datasets. The labels represent 45 possible tag diseases.

Slashdot [72] is a dataset that belongs to text domain. It consists of BoW representation of articles obtained from the website *slashdot.org*. The labels represent different subject categories such as hardware, mobile, news, interviews, games, etc.

Enron [73] is a dataset containing e-mail messages from the Enron corpus. It belongs to the group of text domain. The features are represented in BoW format. The targets represent different topics being considered. For example company strategy, legal advice, humour, etc.

Langlog [52] is a dataset from the text domain. It consists of various topics relating to predominantly English language, obtained from Language Log Forum. The dataset is given in BoW format. There are 75 labels representing different aspects for the language, for example, punctuation, humour, errors, administration, negation, etc.

Arabic200 [74] is a dataset obtained from Russia Today in Arabic news portal. It consists of news articles distributed in 40 categories. The features are numeric. There are multiple variants of the dataset available with 200, 500, 1000, 2000, 3000, 4000 features. The variant with 200 features is used in the experiments.

Stackexs [75] are a set of 6 datasets originating from 6 different forums. In this study the forums of computer science, chess and philosophy are used. The features are given in term-frequency of the words per forum post. These datasets belong to text domain. The labels represent the different topics related to the posts. The datasets are independent among different forums.

Proteins [76] datasets are a set of 3 datasets from the area of bioinformatics. They describe the problem of sub-cellular localization. Sequence descriptors of the proteins for humans [76], virus [77] and plant [86] are taken as input. The calculation of the features is done with the `propy` library [78]. The library takes as input the protein sequences and uses the default settings of the methods used to extract the features. The features describe structural and physio-chemical properties of the proteins and some of them include amino acid compositions, dipeptide compositions, transition, Moran auto-correlation, distributions, sequence-order-coupling numbers, etc.

Reutersk500 [52] is a dataset originating from Reuters RCV1 corpus [87]. Since the RCV1 corpus possesses around 46000 features, [52] applies a feature selection technique to reduce the number of features to 500. The features are given in `tf-idf` format as applied in [79]. Since the labels in the corpus have a hierarchical structure in order to be used in the MLC setting, the hierarchy is flattened.

Tmc2007 [80] is a dataset containing Aviation Safety Reporting textual data. The texts are free text reports, obtained by crew members about various events during a flight. The features are given in a BoW form. The labels represent the various events that may occur during the flight.

Ohsumed [81] is a dataset from text domain. It is a subset from the MEDLINE database, which is a bibliographic database of peer-reviews of medical literature. The features are BoW representation of the words appearing in the reports. The labels represent 23 medical categories of cardiovascular disease.

Ng20 [82] is a dataset containing news data. The features are given in a BoW representation. The labels represent different topics such as politics, cars, religion, space, etc.

Corel5 [83] is a dataset from the multimedia domain. The samples represent Corel images. Each image is segmented using the Normalized Cuts method. The segments are then clustered into regions and described with 33 features each. For each image, there are 5-10 regions describing them. The features represent whether the region is present or not in a particular image. The labels are word description of the region.

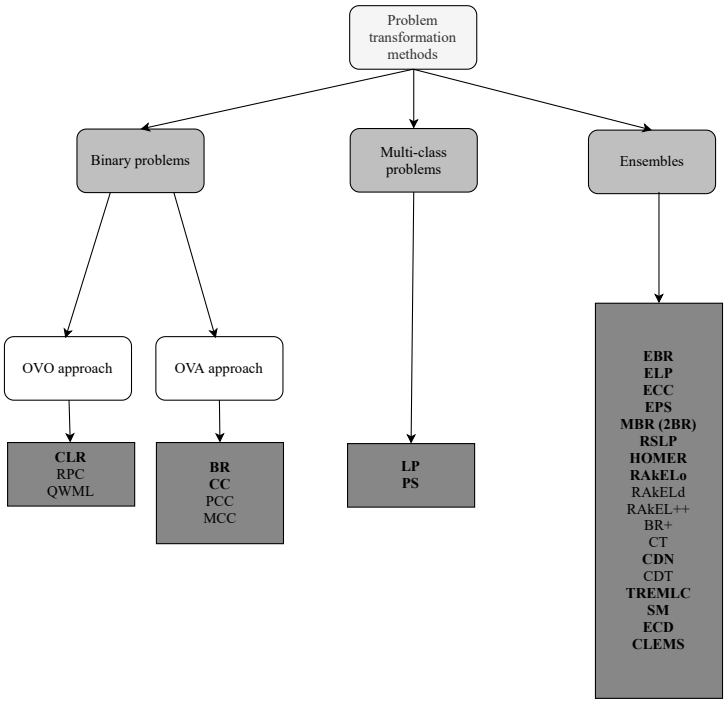

Figure 1: Problem transformation methods for MLC.

Bibtex [84] is a dataset from the text domain. It emerges from the social bookmarking and publication-sharing system Bibsonomy. The data is stored and organized in BibTeX entries. The labels represent the different tags a user can assign to their BibTeX submission to the system.

Delicious [85] is a dataset from the text domain. The data is extracted from *del.icio.us* social bookmarking site on the 1st of April 2007. It contains textual data of web pages alongside their tags. The words appearing on the pages are given in a BoW representation. The labels represent the different tags that can appear on the bookmarking site.

2 Methods

The methods in the area of MLC are separated into two groups: problem transformation and algorithm adaptation [53] (see Figs. 1 and 2).

The group of problem transformation methods approach the problem of MLC with transforming the multi-label dataset into one or multiple binary datasets. This allows for the application of well established single target machine learning methods and building one or multiple single target models. At prediction time, it is required that all built models are invoked to generate the prediction for the test sample.

Algorithm adaptation methods include some adaptation of the training and prediction phases of the single target methods towards handling multiple labels simultaneously. For example, trees change the heuristic used when creating the splits, Support Vector Machines (SVMs) employ additional threshold technique etc. The adaptations provide a mechanism to handle the dependency between the labels directly. Their separation can be made based on the underlying paradigm being adapted. The literature recognizes 5 defined groups of algorithm adaptation methods according to the performed adaptation: trees, neural networks, support vector machines, instance-based and probabilistic [36]. There is another unspecified group of methods that lists approaches from genetic programming and other optimization technique.

2.1 Binary relevance

The Binary Relevance method **(BR)**, introduced in [53], transforms the MLC problem into $|L|$ binary classification problems that share the same feature (descriptive) space as the original descriptive space

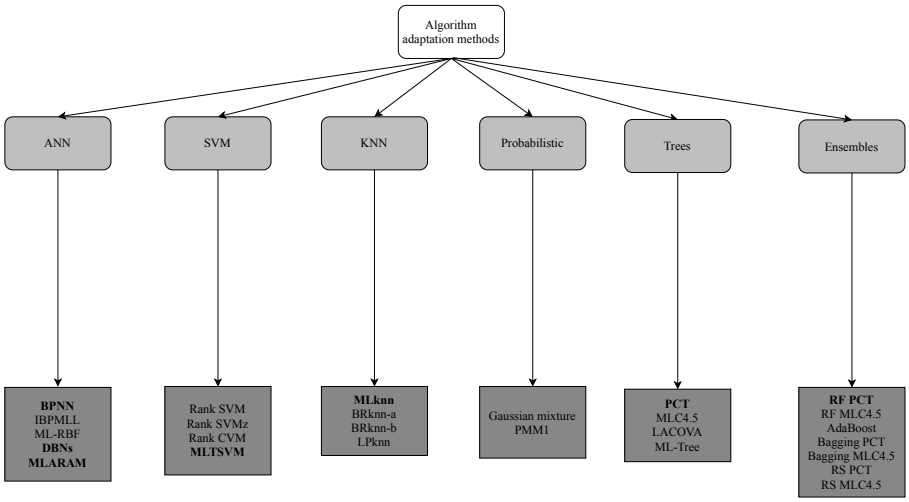

Figure 2: Algorithm adaptation methods for MLC.

of the multi-label problem. Each of the binary problems has assigned one of the labels as a target. It trains one base classifier for each of the transformed problems. It has only one hyper-parameter - the base classifier. This method generalizes beyond the label-sets present in the training samples. It is not suitable for a large number of labels and ignores the label correlations. Due to the necessity of building models for each label, the training of the method can be time-consuming, especially if the computational complexity of the base learning method is large.

2.2 Calibrated label ranking

Calibrated Label Ranking **(CLR)** is a pairwise technique for multi-label ranking. It provides a built-in mechanism to extract bipartitions and thus can be used as a MLC method. The core of the pairwise methods is creating $\frac{l(l-1)}{2}$ single target binary datasets from the multi-label dataset with label-set of size $|L|$, maintaining the original descriptive space. The binary target is generated in such a way that if one of the labels in a given pair, chosen as positive, is different from the other, the example in the newly created dataset obtains a value of 1 and 0 otherwise. If the labels are the same, the example is excluded. In such a way $\frac{l(l-1)}{2}$ binary datasets are created. A base classifier is built on these datasets. CLR introduces one artificial label [88, 89]. This artificial label acts as a complementary label for each of the original labels, thus introducing $|L|$ more models to be built. When ranking for each of the labels is obtained, the artificial variable acts as a split point between the relevant and irrelevant labels, producing bipartition. It has one hyper-parameter to be chosen - the base learner. The strong aspect of this method is that it generates ranking and bipartition. The main drawback is that it is not suitable for a large number of labels, due to the great space and time complexity.

2.3 Label powerset or label combination

Label Powerset **(LP or LC)** [53] transforms the MLC method into a multi-class classification problem in such a way that it treats each unique label-set as a separate class. Any classifier suitable for solving multi-class classifier can be applied to solve the newly created single target multi-class problem. It has only one hyper-parameter, the base multi-class classifier. The positive aspect for the method is that it can preserve label correlation. The negative aspects of this method are that it cannot predict novel label combinations and is prone to underfitting when the number of unique label-sets is large.

2.4 Pruned sets

The method of Pruned Sets **(PS)** [90] aims at reducing the number of unique classes (label-sets) appearing when a multi-label problem is treated as in LP. To achieve this goal, the method has two phases. The first step is the so-called pruning step. The pruning step removes the infrequently occurring label-sets from the training data. The decision on what the infrequent label-set means is left as the practitioner's choice and it is one parameter of the method. The second phase consists of introducing the removed

samples into the training set. This is done by subsampling the label-sets of the infrequent examples for label subsets which satisfy the pruning criterion. The method introduces this as a tuning parameter and it defines the maximal number of frequent label-sets to subsample from the infrequent label-sets. On such a newly created dataset, LP is trained. Additional improvement of the method is introducing a threshold function that enables new label combination to be created at prediction time [52]. In total, there are three hyperparameters for the method: the base multi-class classifier, the pruning value (if the count of the label-sets in the datasets exceeds this number the example is preserved) and the maximal number of frequent label-sets to be reintroduced. The strong point of this method is that it is much faster than Label Powerset. On the negative side, the assumptions can break, and the method (without threshold parameter) is not able to introduce novel multi-label sets which have not been seen in the training data.

2.5 Predictive clustering tree

Predictive Clustering Trees **(PCTs)** [45] are decision trees viewed as a hierarchy of clusters. This method uses the standard `TDITD` algorithm for induction of the tree [91]. At the top node, all data samples belong to the same cluster. This cluster is recursively partitioned into smaller clusters, such that the variance (impurity measure) is reduced. The variance function and the prototype function are selected for the task at hand. In the case of MLC, the variance function is computed as the sum of the Gini indices of the labels. The prototype function returns a vector of probabilities that a sample is labelled with a particular label. As a stopping criteria for growing the tree F-test is used. That is the only hyperparameter of the model needed to be tuned. The positive aspects of this method include the fast time for training and prediction and it is one of the rear MLC methods that can provide interpretable results. As negative aspects are that a single tree may be poor in performance, however, an ensemble of PCT has the state of the art performance [35].

2.6 Back-propagation neural networks

Back-propagation Neural Networks **(BPNN)** [92, 93] is a neural network approach to the problem of MLC. It is the standard multi-layer perception algorithm tailored for multi-label problems. It uses the back-propagation algorithm to calculate the parameters of the network. The hyperparameters of the method are the learning rate, the number of epochs and the number of hidden units. The positive aspect for neural networks in general is that it can provide good performance if a large number of training samples are available. They face drawbacks on the time needed for hyperparameter optimization. A popular approach in this family of methods is the stacking of multiple layers of hidden units, thus increasing the neural network architecture in depth. This approach belongs to a specific family of methods referred to as deep-learning. Given its popularity, a separate paragraph is dedicated to one deep-learning approach used to model higher-level features, Restricted Boltzmann Machines **(RBMs)**.

2.7 Restricted boltzmann machines and deep belief networks

RBMs is a type of deep-learning method that aims to discover the underlying regularities of the observed data [94]. A Boltzmann machine can be represented as a fully connected neural network. The restricted Boltzmann machine additionally has the restriction of connections between neurons in the same layer. Usually, the parameters of the network are learned by minimizing Contrastive divergence [95]. Stacking of multiple RBMs creates the so-called Deep Belief Networks **(DBNs)**. The standard back-propagation algorithm can be used to fine-tune the parameters of the network in a supervised fashion. Using DBNs one can generate new features like a different representation of the data. Those features can be used as input to any multi-label classifier. It is expected that those features are close representatives of the labels being predicted. The hyperparameters of this method are the same as in BPNN, with additional two: the number of hidden layers and the output multi-label classifier. The novel representation of the input data provided by DBNs can lead to improved performance, on the cost of increased time and space complexity for training the method [93].

2.8 Multi-label adaptive resonance associative map

Multi-label ARAM **(MLARAM)** network [96] is an extension of Adaptive Resonance Associative Map neural-fuzzy networks. ARAM network for supervised learning consists of two self-organizing maps sharing the same output neurons. The first self-organizing map tries to encode the input space into

prototypes, while simultaneously tries to characterize the prototypes with a mapping which encodes the labels. A parameter called vigilance is used to control the specificity of the prototypes. Larger values indicated more specific prototypes [97]. MLARAM is an extension of ARAM in such a way that it allows flexibility in determining when a particular node is activated, taking into consideration label dependencies. The output predictions may vary due to the order in which training samples are presented. The flexibility of inclusion depends on a threshold parameter. The parameters to be tuned are the vigilance and threshold. The positive aspect is that it is fast to train and is useful in text classification of a large volume of data, however since it is based on Adaptive Resonance Theory neural-fuzzy networks it has generalization limitation if too many prototypes are built.

2.9 Twin multi-label support vector machine

Twin Multi-Label Support Vector Machine **(MLTSVM)** [98] tries to fit multiple nonparallel hyperplanes to the data to capture the multi-label information embedded in the data. It follows the Twin SVM concept [99], where one tries to find two nonparallel hyperplanes such that each one is closer to its own class, but it is further than the others. At the training phase, this method constructs multiple nonparallel hyperplanes with the purpose of exploiting the multi-label information via solving several quadratic programming problems using fast procedures. The prediction is obtained by calculating the distance of the test sample to the different hyperplanes. The hyperparameters of the method are the threshold above which a label is assigned, empirical risk penalty (determines the trade-off between the loss terms in the loss function) and a regularization parameter. [98] show that MLTSVM outperforms other SVM-based methods for MLC. Its advantage is that it is fast to train because of the underlying procedure for solving the quadratic problem. The authors in their paper do not comment on a performance measure, different than the ranking based and Hamming loss.

2.10 Multi-label k nearest neighbour

Multi-label k Nearest Neighbor **(MLkNN)** [100] method is an adaptation of the Nearest Neighbor [101] paradigm for multi-label problems. It finds the k nearest neighbours of a given sample as in single target kNN algorithm. It constructs prior and conditional probabilities from the training data and thus can use Bayesian inference to obtain the posterior probability for a given label on a given test sample. The parameter of the method is the number of neighbours. It is fast to build, but as a lazy method, obtaining the prediction is a more expensive operation. Thus it may not be suitable in a situation where fast predictions are required.

2.11 Classifier chains

Classifier Chains **(CC)** [72] learning procedure involves two steps. It first starts with the training of $|L|$ single target binary classifiers as in BR. In the second step again $|L|$ single target binary classifiers are trained. However, at this stage, the base classifiers are assumed to coexist in a chain. Each classifier deals with single target problem of augmented feature space. The augmented feature space consists of all the descriptive features and the predictions obtained from the first step of all previous members of the chain. For example, l_j label is predicted by base classifier of the original feature space augmented by the l_1, l_2 ..., l_{j-1} previous chain links. The only hyperparameter to set is the base classifier. Regarding the strong aspects of this method, it enables introducing label correlation to some extent (the order of the labels in the chain is important) and can generalize beyond seen label-sets. On the negative aspect, it is not suitable for a large number of labels because it is applying BR twice.

2.12 Conditional dependency networks

Conditional Dependency Network **(CDN)** [102] aims at encapsulating the dependencies between the labels using dependency networks. Dependency networks are cyclic directed graphical models, where the parents of each variable are its Markov blanket [103]. Markov blanket in graphical model literature represents a set of nodes around a specific node that shields it. The Markov blanket of a node is the only knowledge needed to predict the behaviour of that node and its children [104]. The label dependency information is encoded into the graphical model parameters - the conditional probabilistic distribution associated with each label. The probabilistic distributions are modelled via simple binary classifier models, that as input take the whole feature space augmented by all other labels, with the exclusion of the label being modelled. In the inference phase, it uses the standard model for inference in graphical

models - the Gibbs sampling method. This method assumes that one of the labels can change, assuming that all others are fixed. First, random ordering of the labels is chosen and each label is initialized to some value. In each sampling iteration, all the nodes modeling the labels are visited and the new value of the label being modelled is re-sampled according to the probability model that represents the current label being predicted. It has 3 hyperparameters to tune, the model trained at each node of the network, the number of iterations to perform in the network and the number of operations used to compute the output class probabilities (the final number of operations). This model preserves the label dependencies, however, if there are many numbers of labels the inference performed by the Gibbs sampling method, as part of the Markov Chain Monte Carlo family of algorithms, may need greater time for convergence.

2.13 Meta binary relevance

Meta Binary Relevance **(MBR)** [105], also known as the 2BR method, consists of two consecutive stages of applying BR. First, $|L|$ binary base models are built. At the second (meta) stage, the feature space is augmented with the predictions from the first stage ($|L|$ features are added). New $|L|$ binary models are trained as in BR. A few approaches exist of how the predictions are gathered in the first stage. The predictions can be generated using the full training set, via k fold cross-validation or by ignoring the irrelevant variables into the meta-level. The cross-validation approach is very slow since it requires training of each model at the first level k times, but non-biased information to the meta-level proceeds. The irrelevant information can be filtered using the Φ correlation coefficient to determine if two labels are correlated or not. If they are not correlated, the label is not introduced into the meta-level. [39] show that the version of this method, where the full train set is used at the first stage, is better regarding the other two. To further reduce the bias towards the label being predicted, [106] suggest reducing the number of meta-labels to $|L|-1$ (the label being predicted is excluded). This method is known as **BR+**. MBR has one parameter to tune, the single target base method. On the negative side, MBR and its variants inherit the drawbacks of BR, being not suitable for a large number of labels, due to the large time needed to build the model.

2.14 Ensemble of classifier chains

Ensembles of Classifier Chains **(ECC)** [72] create a meta architecture of CC built on sampled instance from the original dataset. The sampling is done with replacement. [72] argue that sampling with replacement provides better results compared with sampling without replacement. Choosing the percentage of the data for building the models (bag size) is allowed. So the hyper-parameters of the method are the number of CC models in the meta architecture and the bag size. In this method also a random ordering of the chain is considered to provide compensation for introducing non-existence dependency between labels. Using the different random subspace of the training set and utilizing different ordering in the chain impose a diversity in the meta architecture. This method takes into account the label correlation but has a drawback of the large time for training.

2.15 Ensemble of binary relevance

Ensembles of Binary Relevance **(EBR)** [72] build meta architecture of BR as a base learner in the meta architecture. The sampling is done with replacement. The hyperparameter of the method is the number of BR models in the meta architecture. Although it can provide novel label-sets at prediction, it still has the assumption of labels independence.

2.16 Ensemble of chi-dep's

Chi-dep [107] is a multi-label method that is based on the identification of label dependencies using statistical tests between the labels. χ^2 statistical test for independence for each possible combination of two labels is used. It first tries to identify groups of dependent and independent labels. After their identification, the BR approach for the independent groups, and LP for the dependent labels are trained. At prediction time, the sample is processed by each of the models and the prediction is generated accordingly. This method provides a trade-off between the assumption of independence of the BR method, and the problem of a large number of unique label-sets the LP method is facing. The hyperparameters of the method are the base learners for BR and LP method. The positive aspect of the method is that it provides a trade-off between high bias and variance of the BR method, and the low bias and high variance of the LP method.

Ensemble of Chi-dep **(ECD)** [58] builds several Chi-dep models. First, it generates a large number of possible label-sets partitions at random. Each of the partitions is represented by normalized χ^2 score of all the label pairs inside the partition, based on the inside pairwise χ^2 scores. Then, the top m distinct sets with the highest scores are included in the meta architecture. The hyperparameters of the method are the number of meta architecture members and the number partitions to evaluate. The positive aspects of the method are that it can further reduce the variance of a single Chi-dep method, however, it suffers from large time complexity. Thus a fast base learning method should be utilized.

2.17 Ensemble of label powerset

Ensembles of Label Powersets **(ELP)** [39] create a meta architecture of LP method on sampled prototypes from the original set. The sampling is done with replacement. The hyperparameter of the method is the number of LP models built in the meta architecture. It provides an opportunity to obtain voting for Label Powerset method, however, it inherits its large computational complexity, and it is virtually inefficient for datasets with a large number of unique label-sets.

2.18 Ensemble of pruned sets

Ensemble of Pruned Sets **(EPS)** [90] create a meta architecture of MLC methods of PS method on sampled prototypes from the original set. The sampling is done without replacement with 63 % of the dataset being sampled. Additional parameters of the method are the number of members of the meta architecture as well as the number of samples to be sampled from the training size (bag size). This method can predict novel label-sets, thus diminishing one of the disadvantages of a standalone Pruned Set method without thresholding. It is good for small and large datasets. Its disadvantage is that it is not able to perform well when there are many diverse label-sets without frequent reoccurring of some of the label-sets. This is due to reducing the training set to a handful training example due to pruning strategy.

2.19 Random k labelsets

Random k Labelsets **(RAkEL)** [108] is a meta architecture of MLC methods. It uses multiple LP models trained on random partitions of the label space. Usually the size of the label set is small. Each of the LP methods should learn 2^k classes instead of $2^{|L|}$, where $k << |L|$. Moreover, the resulting multi-class problems have a much better distribution of the classes in terms of class balance. In [108] two versions of the method are introduced. The first version does not allow for overlap between the groups when creating the label-sets and is called RAkEL disjoint. The second version allows for overlapping between the labels in the created label-sets. This gives the advantage for the same label to be included by the different LP models. The predictions are obtained by voting. Further improvements of the method are proposed in [109]. They propose using the classification confidence intervals instead of voting. However, [39] shows that voting versions of RAkEL achieve better results. There are two hyperparameters to be tuned: the size of label-sets k and the number of models m. The positive aspect of RAkEL is that it uses a smaller number of classifiers than BR, can provide better generalization and is not underfitting as LP. It does not scale well in time, as the number of labels and number of instances increases.

2.20 Hierarchy of multi-label classifiers

Hierarchy of Multi-label Classifiers **(HOMER)** [85] is a meta architecture based on the transformation of the problem into a tree-shaped hierarchy of smaller multi-label problems, utilizing the divide and conquer strategy. The tree is constructed in such a manner that, at the leaves, there are the singleton labels, whether the internal nodes represent joint label-sets. A node will contain a training sample iff the sample is annotated with at least one of the labels of the label-set contained in a node. The method consists of two phases: first, the tree is built such that labels from the parent node are distributed to the children nodes using balanced clustering algorithm. Second, the multi-label model is trained on a reduced label-subset, and the process is repeated until all nodes are with one label. Such an approach provides the opportunity to cluster dependent labels into a single node. The hyperparameters of the method are the number of children for a parent node (number of clusters) and the base learner. It is predominantly useful in tasks with a large number of labels where it is shown to has the best predictive performance [35]. However, the constructed hierarchy is not utilized in problems with a smaller number of labels, hence this method does not show its full potential on datasets with such property [39].

2.21 Random subspace for multi-label classification

Random Subspace **(RS)** multi-label method is an extension of the Random Subspace methodology for single target prediction [110] into the area of MLC. It works with a random sampling of the features. Additionally one can subsample the instances from the training set. For each of the subsamples generated alongside the two dimensions of features and instances, either problem transformation or algorithm adaptation method can be used. There are four hyperparameters to tune, the percentage of the attribute space to be used, the percentage of sample space to be used, the number of models in the meta architecture to be built and the multi-label classifier at the base level. This meta architecture method is usually faster than bagging and other meta architecture methods likewise. This property is hugely dependent on the properties of the base multi-label learner utilized. This method for multi-label problems is first considered in [52].

2.22 Random forest of predicting clustering trees

Random Forest of Predictive Clustering Trees **(RFPCT)** [111] follows the traditional random forest method being a combination of Bagging and Random Subspace ensemble methods [112]. It samples both the instance space (sampling with replacement) and feature (random). As base models, methods with low bias and high variance are used. In a multi-label context as a base model, RFPCT uses the algorithm adaptation method PCT. The tree is built with randomly sampled feature subset and they are fully grown. The parameters of the method are the number of features to be used when building the trees and the number of ensemble members. The positive aspects of the method are that it is fast, has state-of-the-art predictive performance [35] and can tackle the correlation between the labels inherently. On the negative side, RFPCT is not suitable for datasets with large sparse feature vectors. Due to the process of a random selection of attributes, it can often happen that these sparse features will be chosen for building the trees. In such a scenario, the trees will have a low predictive performance and that will hurt the overall predictive performance of the ensemble.

2.23 Boosting

The AdaBoost **(AdaBoost)** [56] method is introducing a set of weights maintained both on the samples (as in classical AdaBoost method [113]) and the labels. The formula for calculating the weights incorporates the sample-label pairs that are miss-classified by the base classifier. At each iteration, the method builds a simpler classifier as a `Decision Stump` (decision tree of depth 1). The classifier uses weights to focus more on the samples that are hard to predict. The base classifier should provide confidences that are used to obtain a prediction. The final prediction is obtained by combining the confidences of each of the base models, weighted by the corresponding model weights. The parameter of the method is the number of boosted decision trees. This method is the same as applying AdaBoost to $|L|$ binary datasets as in BR.

2.24 Cost-sensitive multi-label embedding

Cost-Sensitive Multi-label Embedding **(CLEMS)** [114] belongs to a special type of family of multi-label methods, known as Label Embedding methods. In general, these methods try to embed the label-space into a particular number of dimensions using some embedding technique. It is assumed that the embedded space represents the hidden structure of the labels. For learning, either problem transformation or algorithm adaptation method is applied to the augmented feature space. At prediction time, embedding methods employ a regression technique to predict the value of the embedded features. One type of label embedding method is known as Cost-Sensitive Embedding. It considers the cost being optimized as additional criteria. In particular, the method considered here employs weighted multidimensional scaling as embedding technique [115]. It embeds cost-matrix of unique label combinations. The cost matrix contains the cost of mistaking a given label combination for another. The hyperparameters of CLEMS are the cost function, the underlying MLC method, the regression method used to predict the values of the embedding features and the number of embedding dimensions. The most effective value for the number of embedding dimensions is the number of labels. The positive aspects of the method are that it can provide good results for a specific cost function being optimized. On the negative side, this method is dependent on the underlying MLC method and requires building a specific model for each cost function for optimal performance per measure.

2.25 Triple random ensemble

Triple Random Ensemble **(TREMLC)** [57] is a meta architecture for MLC. It is a combination of 3 strategies: sampling of the instance space, sampling of the feature space and sampling of the target space. It is in essence a combination of Random Forest built with RAkEL as a base classification method. The parameters of the method are bag size, number of features to subsample, the size of label-sets and the number of models to be built.

2.26 Subset mapper

Subset Mapper **(SM)** [116] uses Hamming distance in order to make mapping between the output of multi-label classifier and a known label combination seen in the training set. From the predicted probability distribution, SM will produce a label subset and will calculate the hamming distance to the labels of the training instances. The new test sample as prediction will take the labelset that resulted in the smallest distance. The parameter of this method is the base learning MLC method. The negative aspect of this method is that it is not able to generalize beyond the seen examples in the training set.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 10971105.

[2] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Cambridge, USA: MIT Press, 2001, pp. 681–687.

[3] J. K. Cornelis, H. Matthias, T., K. Dragi, S. Elisabeth van, R. Omid, F. Kees LMC, W. Louis, S. Nigel DL, D. Sašo, H. Marille C., and O. Tom HM, "Combined chemical genetics and data-driven bioinformatics approach identifies receptor tyrosine kinase inhibitors as host-directed antimicrobials," *Nature Communications*, vol. 9, pp. 1–14, 2018.

[4] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation : a publication of the IEEE Neural Networks Council.*, vol. 1, p. 67, 1997.

[5] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proc. of KDD-2013*, 2013, pp. 847–855.

[6] R. D. KING, C. FENG, and A. SUTHERLAND, "Statlog: Comparison of classification algorithms on large real-world problems," *Applied Artificial Intelligence*, vol. 9, pp. 289–333, 1995.

[7] G. Widmer. [Online]. Available: http://www.ofai.at/research/impml/metal/metal-metalearning.html

[8] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, pp. 77–95, 2002.

[9] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*. Berlin: Springer, 2009.

[10] C. Lemke, M. Budka, and B. Gabrys, "Metalearning: a survey of trends and technologies," *Artificial Intelligence Review*, vol. 44, no. 1, pp. 117–130, 2015.

[11] P. Brazdil and C. Giraud-Carrier, "Metalearning and algorithm selection: progress, state of the art and introduction to the 2018 special issue," *Machine Learning*, vol. 107, no. 1, pp. 1–14, 2018.

[12] P. Brazdil, C. Giraud-Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to Data Mining*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[13] J. Vanschoren, "Meta-learning: A survey," *CoRR*, vol. abs/1810.03548, 2018.

[14] S. Sohn, "Meta analysis of classification algorithms for pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 1137–1144, 1999.

[15] L. Y. Pratt, "Discriminability-based transfer between neural networks," in *Advances in Neural Information Processing Systems 5*. Morgan-Kaufmann, 1993, pp. 204–211.

[16] S. Thrun and L. Pratt, Eds., *Learning to Learn*. USA: Kluwer Academic Publishers, 1998.

[17] M. B. Ring, "Child: A first step towards continual learning," in *Machine Learning*, 1997, pp. 77–104.

[18] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ser. ICML'93. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, p. 4148.

[19] J. S. Bridle and S. J. Cox, "Recnorm: Simultaneous normalisation and classification applied to speech recognition," in *Advances in Neural Information Processing Systems 3*. Morgan-Kaufmann, 1991, pp. 234–240.

[20] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[21] S. Thrun, "Is learning the n-th thing any easier than learning the first?" in *Advances in Neural Information Processing Systems*. The MIT Press, 1996, pp. 640–646.

[22] J. R. Rice, "The algorithm selection problem," in *Advances in computers*. Elsevier, 1976, vol. 15, pp. 65–118.

[23] C. Giraud-Carrier, "Beyond predictive accuracy: What?" 1998.

[24] D. W. Aha, "Generalizing from case studies: A case study," in *In Proceedings of the Ninth International Conference on Machine Learning*. Morgan Kaufmann, 1992, pp. 1–10.

[25] J. Gama and P. Brazdil, "Characterization of classification algorithms," in *Proceedings of the 7th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence*. Berlin, Heidelberg: Springer-Verlag, 1995, p. 189200.

[26] P. Domingos, "A unified bias-variance decomposition and its applications," in *In Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, 2000, pp. 231–238.

[27] R. Gomes Mantovani, A. Rossi, J. Vanschoren, B. Bischl, and A. Carvalho, "To tune or not to tune: recommending when to adjust svm hyper-parameters via meta-learning," in *2015 International Joint Conference on Neural Networks (IJCNN), 12-17 July 2015, Killarney, Ireland*. United States: Institute of Electrical and Electronics Engineers, 2015, pp. 1–8.

[28] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York, USA: Wiley-Interscience, 2000.

[29] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc., 2001.

[31] P. Ridd and C. Giraud-Carrier, "Using metalearning to predict when parameter optimization is likely to improve classification accuracy," in *Proceedings of the 2014 International Conference on Meta-Learning and Algorithm Selection - Volume 1201*, 2014, p. 1823.

[32] J. Levatić, D. Kocev, and S. Džeroski, "The importance of the label hierarchy in hierarchical multi-label classification," *Journal of Intelligent Information Systems*, vol. 45, 12 2014.

[33] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," 2017.

[34] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 42894300.

[35] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45, pp. 3084 – 3104, 2012.

[36] F. Herrera, A. J. Rivera, M. J. del Jesus, and F. Charte, *Multilabel Classification*. Springer Cham, Switzerland: Springer, 2016.

[37] J. Bogatinovski, "A comprehenisve study of multi-label classification methods," Master's thesis, Joef Stefan International Postgraduate School, Ljubljana, Slovenia, 2019.

[38] M. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 107–120, 2015.

[39] J. M. Moyano, E. L. G. Galindo, K. J. Cios, and S. Ventura, "Review of ensembles of multi-label classifiers: Models, experimental study and prospects," *Information Fusion*, vol. 44, pp. 33–45, 2018.

[40] E. Gibaja and S. Ventura, "A tutorial on multilabel learning," *ACM Computing Surveys*, vol. 47, no. 3, pp. 52:1–52:38, 2015.

[41] A.-O. Reem, P. Flach, and K. Meelis, "Multi-label classification: A comparative study on threshold selection methods," in *1st International Workshop on Learning over Multiple Contexts (LMCE) at ECML-PKDD 2014*, 2014.

[42] J. M. Moyano, E. L. G. Gibaja, and S. Ventura, "MLDA: A tool for analyzing multi-label datasets," *Knowledge-Based Systems*, vol. 121, pp. 1–3, 2017.

[43] L. Chekina, L. Rokach, and B. Shapira, "Meta-learning for selecting a multi-label classification algorithm," in *Proceedings of the 11th International Conference on Data Mining Workshops*. Washington, DC, USA: IEEE Computer Society, 2011, pp. 220–227.

[44] L. Rokach, "Decomposition methodology for classification tasks: A meta decomposer framework," *Pattern Anal. Appl.*, vol. 9, pp. 257–271, 2006.

[45] H. Blockeel, L. D. Raedt, and J. Ramon, "Top-down induction of clustering trees," in *Proceedings of the 15th International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1998, pp. 55–63.

[46] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, "Tree ensembles for predicting structured outputs," *Pattern Recognition*, vol. 46, pp. 817–833, 2013.

[47] M. Wever, F. Mohr, and E. Hllermeier, "Automated multi-label classification based on ML-Plan," 2018.

[48] F. Charte and D. Charte, "Working with multilabel datasets in r: The mldr package," *The R Journal*, vol. 7, no. 2, p. 149, 2015.

[49] M. Lundberg, S., G. Erion, and H. e. a. Chen, "From local explanations to global understanding with explainable AI for trees." *Nature Machine Intelligence*, vol. 2, pp. 56–67, 2020.

[50] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*. New York, USA: ACM, 2006, pp. 161–168.

[51] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the stratification of multi-label data," in *Machine Learning and Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer, 2011, pp. 145–158.

[52] J. Read, "Scalable multi-label classification," Ph.D. dissertation, University of Waikato, Hamilton, New Zeland, 2010.

[53] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 2007, pp. 1–13, 2007.

[54] J. N. van Rijn and F. Hutter, "Hyperparameter importance across datasets," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, USA: ACM, 2018, pp. 2367–2376.

[55] R. Mantovani, T. Horvath, R. Cerri, S. Barbon, J. Vanschoren, and A. de Carvalho, "An empirical study on hyperparameter tuning of decision trees," 12 2018.

[56] R. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, pp. 135–168, 2000.

[57] G. Nasierding, A. Kouzani, and G. Tsoumakas, "A triple-random ensemble classification method for mining multi-label data," in *IEEE International Conference on Data Mining Workshops*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 49–56.

[58] L. Tenenboim, L. Rokach, and B. Shapira, "Identification of label dependencies for multi-label classification," in *2nd International Workshop on Learning from Multi-Label Data*, 2010, pp. 53–60.

[59] K. Douibi, M. Benabid, N. Settouti, and M. Chikh. (2017) An analysis of ambulatory blood pressure monitoring. [Online]. Available: https://data.mendeley.com/datasets/y4dh3b3tfx/1

[60] A. Rivolli, L. C. Parker, and A. C. P. d. L. F. de Carvalho, "Food truck recommendation using multi-label classification," in *Progress in Artificial Intelligence*. Cham, Switzerland: Springer, 2017, pp. 585–596.

[61] E. C. Goncalves, A. Plastino, and A. A. Freitas, "A genetic algorithm for optimizing the label ordering in multi-label classifier chains," in *Proceedings of the 25-th International Conference on Tools with Artificial Intelligence*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 469–476.

[62] G. P. Liu, G. Z. Li, Y. L. Wang, and Y. Q. Wang, "Modelling of inquiry diagnosis for coronary heart disease in traditional chinese medicine by using multi-label learning," *BMC Complementary and Alternative Medicine*, vol. 10, p. 37, 2010.

[63] H. Blockeel, S. Džeroski, and J. Grbović, "Simultaneous prediction of multiple chemical parameters of river water quality with TILDE," in *Principles of Data Mining and Knowledge Discovery*. Berlin, Heidelberg: Springer, 1999, pp. 32–40.

[64] A. Wieczorkowska, P. Synak, and Z. Ra's, "Multi-label classification of emotions in music," in *Intelligent Information Processing and Web Mining*, M. A. Klopotek, S. Wierzchon, and K. Trojanowski, Eds. Heilderberg, Berlin: Springer, 2006.

[65] J. Xu, J. Liu, J. Yin, and C. Sun, "A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously," *Knowledge-Based Systems*, vol. 98, pp. 172–184, 2016.

[66] F. Briggs, B. Lakshminarayanan, L. Neal, X. Z. Fern, R. Raich, S. Hadley, A. Hadley, and M. Betts, "Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach," *The Journal of the Acoustical Society of America*, vol. 131, no. 6, pp. 4640–4650, 2012.

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

[68] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 467–476, 2008.

[69] S. Diplaris, G. Tsoumakas, P. A. Mitkas, and I. Vlahavas, "Protein classification with multiple algorithms," in *Advances in Informatics*. Berlin, Heidelberg: Springer, 2005, pp. 448–456.

[70] S. Hitesh, S. Vaibhav, K. Kusum, G. Eugenia, C. Pramit, and L. Cristina. (2013) The Yelp dataset challenge - multilabel classification of Yelp reviews into relevant categories. [Online]. Available: https://www.ics.uci.edu/~vpsaini/

[71] K. Crammer, M. Dredze, K. Ganchev, P. P. Talukdar, and S. Carroll, "Automatic code assignment to medical text," in *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 129–136.

[72] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine Learning*, vol. 85, p. 333, 2011.

[73] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *Machine Learning: ECML 2004. Lecture Notes in Computer Science,*. Berlin, Heidelberg: Springer, 2004, pp. 217–226.

[74] A.-S. Bassam, A. Masri, K. Graham, A. Shahrul, and Noah. (2018) RTA news. [Online]. Available: https://data.mendeley.com/datasets/322pzsdxwy/1

[75] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "QUINTA: A question tagging assistant to improve the answering ratio in electronic forums," in *EUROCON 2015 - International Conference on Computer as a Tool*. Washington, DC, USA: IEEE Computer Society, 2015, pp. 1–6.

[76] H. Zhou, Y. Yang, and H.-B. Shen, "Hum-mPLoc 3.0: Prediction enhancement of human protein subcellular localization through modeling the hidden correlations of gene ontology and functional domain features," *Bioinformatics*, vol. 33, p. 843853, 2016.

[77] H.-B. Shen and K.-C. Chou, "Virus-mPLoc: A fusion classifier for viral protein subcellular location prediction by incorporating multiple sites." *Journal of Biomolecular Structure and Dynamics*, vol. 28, pp. 175–186, 2010.

[78] C. Dong-Sheng, X. Qing-Song, and L. Yi-Zeng, "propy: a tool to generate various modes of Chou's PseAAC," *Bioinformatics*, vol. 29, no. 7, pp. 960–962, 2013.

[79] C. Buckley, G. Salton, and J. Allan, "The effect of adding relevance information in a relevance feedback environment," in *Proceedings of the 7th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. London: Springer, 1994, pp. 292–300.

[80] A. Srivastava and B. Zane-Ulman, "Discovering recurring anomalies in text reports regarding complex space systems," in *IEEE Aerospace Conference*. IEEE, 2005, pp. 3853–3862.

[81] J. Thorsten, "Text categorization with suport vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*. Berlin, Heidelberg: Springer, 1998, pp. 137–142.

[82] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the 12th International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 1995, pp. 331–339.

[83] P. Duygulu, K. Barnard, J. F. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Computer Vision — ECCV 2002*. Berlin, Heidelberg: Springer, 2002, pp. 97–112.

[84] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *Proceedings of the ECML/PKDD 2008 Discovery Challenge*, 2008.

[85] G. Tsoumakas, I. Katakis, and I. P. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proceedings of the Workshop on Mining Multidimensional Data at ECML/PKDD 2008*, 2008, pp. 53–59.

[86] K.-C. Chou and H.-B. Shen, "Plant-mPLoc: A top-down strategy to augment the power for predicting plant protein subcellular localization," *PLoS ONE*, vol. 5, 2010.

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

[88] K. Brinker, "On active learning in multi-label classification," in *From Data and Information Analysis to Knowledge Engineering*. Berlin, Heidelberg: Springer, 2006, pp. 206–213.

[89] J. Fürnkranz, E. Hüllermeier, E. LozaMencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[90] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Proceedings of the 8th IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 995–1000.

[91] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[92] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 1338–1351, 2006.

[93] J. Read and F. Perez-Cruz, "Deep learning for multi-label classification," 2014.

[94] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[95] G. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computing*, vol. 14, pp. 1771–1800, 2002.

[96] E. Sapozhnikova, "ART-based neural networks for multi-label classification," in *Advances in Intelligent Data Analysis VIII*. Berlin, Heidelberg: Springer, 2009, pp. 167–177.

[97] A. H. Tan, "Adaptive resonance associative map," *Neural Networks*, vol. 8, pp. 437 – 446, 1995.

[98] W. J. Chen, Y. H. Shao, C. N. Li, and N. Y. Deng, "MLTSVM: A novel twin support vector machine to multi-label learning," *Pattern Recognition*, vol. 52, pp. 61–74, 2016.

[99] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 905–910, 2007.

[100] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," in *IEEE International Conference on Granular Computing*. Washington, DC, USA: IEEE, 2005, pp. 718 – 721.

[101] E. V. Ruiz, "An algorithm for finding nearest neighbours in (approximately) constant average time," *Pattern Recognition Letters*, vol. 4, pp. 145–157, 1986.

[102] Y. Guo and S. Gu, "Multi-label classification using conditional dependency networks," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. Barcelona, Spain: AAAI Press, 2011, pp. 1300–1305.

[103] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency networks for inference, collaborative filtering, and data visualization," *Journal of Machine Learning Research*, vol. 1, pp. 49–75, 2001.

[104] J. Pearl, "Markov and bayesian networks: Two graphical representations of probabilistic knowledge," in *Probabilistic Reasoning in Intelligent Systems*, J. Pearl, Ed. San Francisco (CA): Morgan Kaufman Publishers, 1988, pp. 77–141.

[105] G. Tsoumakas, D. Anastasios, S. Eleftherios, M. Vasileios, K. Ioannis, and I. P. Vlahavas, "Correlation-based pruning of stacked binary relevance models for multi-label learning," in *1st International Workshop on Learning from Multi-Label Data*, 2009, pp. 101–116.

[106] E. Alvares-Cherman, J. Metz, and M. C. Monard, "Incorporating label dependency into the binary relevance framework for multi-label classification," *Expert Systems with Applications*, vol. 39, no. 2, pp. 1647 – 1655, 2012.

[107] L. Tenenboim, L. Rokach, and B. Shapira, "Multi-label classification by analyzing labels dependencies," in *Proceedings of the 1st International Workshop on Learning from Multi-Label Data*, 2009, pp. 117–131.

[108] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multi-label classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, pp. 1079–1089, 2011.

[109] L. Rokach, A. Schclar, and E. Itach, "Ensemble methods for multi-label classification," *Expert Systems with Applications*, vol. 41, pp. 7507 – 7523, 2014.

[110] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[111] D. Kocev, "Ensembles for predicting structured outputs," Ph.D. dissertation, Joef Stefan International Postgraduate School, Ljubljana, Slovenia, 2011.

[112] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[113] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.

[114] K. H. Huang and H. T. Lin, "Cost-sensitive label embedding for multi-label classification," *Machine Learning*, vol. 106, no. 9, pp. 1725–1746, 2017.

[115] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, pp. 1–27, 1964.

[116] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[117] A. G. C. de Sá, G. L. Pappa, and A. Freitas, "Multi-label classification search space in the MEKA software," 2018.