*Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 4, 829–840*
DOI: 10.2478/v10006-012-0061-2

# MULTI–LABEL CLASSIFICATION USING ERROR CORRECTING OUTPUT CODES

TOMASZ KAJDANOWICZ, PRZEMYSŁAW KAZIENKO

Institute of Informatics
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: {tomasz.kajdanowicz,kazienko}@pwr.wroc.pl

A framework for multi-label classification extended by Error Correcting Output Codes (ECOCs) is introduced and empirically examined in the article. The solution assumes the base multi-label classifiers to be a noisy channel and applies ECOCs in order to recover the classification errors made by individual classifiers. The framework was examined through exhaustive studies over combinations of three distinct classification algorithms and four ECOC methods employed in the multi-label classification problem. The experimental results revealed that (i) the Bode–Chaudhuri–Hocquenghem (BCH) code matched with any multi-label classifier results in better classification quality; (ii) the accuracy of the binary relevance classification method strongly depends on the coding scheme; (iii) the label power-set and the RAkEL classifier consume the same time for computation irrespective of the coding utilized; (iv) in general, they are not suitable for ECOCs because they are not capable to benefit from ECOC correcting abilities; (v) the all-pairs code combined with binary relevance is not suitable for datasets with larger label sets.

**Keywords:** machine learning, supervised learning, multi-label classification, error-correcting output codes, ECOC, ensemble methods, binary relevance, framework.

## 1. Introduction

Error-Correcting Output Codes (ECOCs) have been used to address diverse problems in pattern recognition; for example, in designing combined classifiers (or ensemble) ECOCs may provide diversity among classifiers by means of dichotomies, which may boost the accuracy of classification. Their ability to reinforce the classification accuracy may be utilized in problems concerning classification of complex outputs such as multi-label classification. ECOC origins are based on theoretic achievements in communication. Studies over ECOCs were addressing the problem of recovering the original signal block transmitted through a noisy channel. Assuming that a signal represents binary-coded information of classes assigned to instances and a noisy channel is built of binary classifiers, ECOCs, in an appropriate settlement, may be successfully applied to multi-class classification. In such a setup the multi-class problem is reduced to several binary classification problems and thanks to the ECOC correcting ability a small portion of wrongly inferred decisions by binary classifiers may be recovered.

The main contribution of the paper is the framework presented in Section 4 and tested in Section 6. It introduces coding before the learning of the multi-label classifier and encoding after learning and testing to make use of ECOC correction abilities. Additionally, experimental results have shown that some combinations of ECOC and multi-label classification methods may be more accurate than a single multi-label classifier itself, while other combinations have lower quality (see Section 6).

**1.1. Problem description.** The standard, conventional classification assumes each instance is associated with exactly one of a finite set of possible classes. An extended classification problem may allow instances to be associated with several labels simultaneously, which is addressed by multi-label classification, usually denoted as a label-set. Classical classification aims to learn a function $f$ that maps an input $x \in \mathbb{X}$ to an output class $c \in \mathcal{C}$, i.e., results of classification—values $y$ belong to only one of the classes from $\mathcal{C}$, $\mathbb{X} \longrightarrow \mathcal{C}$. If the number of classes is two ($card(\mathcal{C}) = 2$), we have the simplest, binary classification. In the case of $card(\mathcal{C}) > 2$, classification

is called multi-class or multi-nomial but it still assigns a single class $c \in \mathcal{C}$ to each input instance (case) $x$. An example of the multi-class problem is the assignment of mother tongues to people. Each person $x$ can be a native speaker of only one language—class $c$ out of many in $\mathcal{C}$, e.g., John's mother tongue is English but Eve's is Chinese.

The goal of mapping to multi-label target is, in turn, to associate an input instance $x$ to a subset $y$ of all possible distinct labels $\mathcal{L}$, $y \subset \mathcal{L}$. The set $\Lambda$ of all subsets $y$ existing in the data ($y \in \Lambda$) is itself a subset of the power set $2^{\mathcal{L}}$ of all labels $\mathcal{L}$: $\Lambda \subset 2^{\mathcal{L}}$. This means that multi-label classification is a mapping: $\mathbb{X} \longrightarrow 2^{\mathcal{L}}$. In consequence, $x$ can have assigned many labels from $\mathcal{L}$. For example, people can speak many languages (not only their mother tongues). In such a case, the label set $\mathcal{L}$ represents all human languages (with, e.g., 200 elements), whereas $\Lambda$ is the set of all real combinations of languages (note that $\Lambda$ does not contain all $2^{200}$ possible combinations—one cannot speak as many as 100 languages) and $y \in \Lambda$ is a concrete set of languages spoken by a person $x$, e.g., $\Lambda = \{\{\text{Chinese}\}, \{\text{German}\}, \{\text{English}\}, \{\text{Polish}\}, \{\text{Swahili}\}, \{\text{English, Polish}\}, \{\text{English, Polish, Swahili}\}, \{\text{Chinese, German, Polish}\}\}$. Hence, classification may result in the following assignments: John $\longrightarrow$ {English, Polish, Swahili}, Eve $\longrightarrow$ {Chinese, German, Polish}, Peter $\longrightarrow$ {English, Polish}, Ubunu $\longrightarrow$ {Swahili}.

The difference between the class set $\mathcal{C}$ used in binary or multi-class classification and the set of distinct labels $\mathcal{L}$ is more philosophical, rather than formal. *A class* is usually understood in terms of common properties of its members $x$ (Sammut and Webb, 2011), whereas *a label* (also called *a class label*) can be treated as a short description of the class. Anyway, it should be emphasized that there exists exactly one label $l \in \mathcal{L}$ for each corresponding class $c \in \mathcal{C}$. Finally, one may state that in multi-label classification we actually predict many equivalent classes. The essential difference between multi-class (including binary) and multi-label classification consists in their either simple or complex output, respectively, i.e., a single class is predicted in the case of a multi-class problem but many labels (class labels) are assigned to a single instance $x$ in multi-label classification.

Due to implementation reasons (encoding/decoding), the target output values $y$ will not be treated as a subset of $\mathcal{L}$ but as an equivalent binary vector with the length of $card(\mathcal{L})$. Hence, we order all possible labels and each $y$'s vector coordinate equals '1' if the corresponding label occurs for a given instance $x$ or '0' otherwise, i.e., if $\mathcal{L}$ is ordered $\langle$English, German, Polish$\rangle$, then $y = \langle 1, 0, 1\rangle$ means that a person linked with $y$ speaks English and Polish.

In general, a multi-label classification problem can be solved either as a set of decomposed, independent binary (single-label) classification problems (binary relevance, see Section 3.1) or as a coherent complex task (see Sections 3.2, 3.3, and 4). The former treats the assignment of each label $l \in \mathcal{L}$ independently and aggregation of their results afterwards, e.g., a person $x$ (i) speaks English or not, (ii) speaks Polish or not, (iii) speaks Chinese or not, etc. However, then we lose some possible existing relations between labels, e.g., people tend to speak languages used in the neighbouring countries so Polish is more likely to co-occur with German, rather than with Swahili. In the latter, we consider all possible labels simultaneously inside the classification method, making good use of correlations between labels. This second ensemble approach is extensively studied in further sections and it is a basis for the new framework for multi-label classification with error correcting output codes (Section 4).

**1.2. Motivation.** The multi-label prediction problem has plenty of possible application domains, e.g., in meta-learning (Jankowski, 2012). We can find them in almost every place where we have many-to-many relationships between known ($\mathbb{X}$) and unknown ($\mathcal{L}$) objects. For example, we would like to recommend to each student a set of courses in a lifelong e-learning service based on prior selections and profiles of other students. We may also want to predict what package (set) of services a new customer is willing to buy. In another case, one can automatically assign a set of descriptive tags-labels to texts (Schapire and Singer, 2000) or multimedia objects—videos or pictures (Boutell *et al.*, 2004) in a multimedia sharing system. The authors of this paper have utilized multi-label classification in debt portfolio valuation where the output of classification for each debt is a set of repayments in the following periods (Kajdanowicz and Kazienko, 2009a; 2009b).

The main goal of the paper is to introduce a new ECOC-based framework for multi-label classification and evaluate it on real data sets in comparison with other methods. The conducted experiments provide a thorough study over several distinct multi-label classification algorithms (binary relevance, label power-sets and random $k$-label-sets) as well as various ECOC designs: the repetition code, the Bode–Chaudhuri–Hocquenghem (BCH) code or the all-pairs code applied to the problem.

This paper is a comprehensively extended and completely rewritten version of the work by Kajdanowicz *et al.* (2011). The main contribution of this paper compared with that by Kajdanowicz *et al.* (2011) is that the problem is explicitly and precisely placed in the domain of multi-label classification and the general framework is proposed. Additionally, new experiments on a wide range of datasets from different domains are carried out, using 12 combinations of ECOC and

multi-label classifiers (only one used by Kajdanowicz *et al.* (2011)), as well as various quality measures.

**1.3. Paper outline.** After an analysis of related work in Section 2, three main approaches to multi-label classification are presented in Section 3: binary relevance, label power-set and RAkEL. The crucial idea of the paper, i.e., the framework for application of ECOCs to multi-label classification, is introduced in Section 4. Three selected error correcting output codes methods as well as a brief discussion on application of ECOCs to the multi-label classification problem can be found in Section 5. Experimental studies on six data sets using combinations of the three multi-label classifiers discussed in Section 3 (binary relevance, label power-set and RAkEL) with four ECOC methods from Section 5 (none, repetition, BCH and all-pairs) together with the analysis of their results are described in Section 6. Section 7 contains general conclusions.

## 2. Related work

Multi-label classification as the supervised classification task assumes the general possibility of each data instance to be associated with multiple class labels. Thus, the problem of multi-label classification requires specialized methods and algorithms in order to provide satisfactory solutions. With respect to the current focus in the field, multi-label learning has three main challenges: (i) discovering and modelling label dependencies, (ii) dimensionality handling (output space of $2^{\mathcal{L}}$ instead of $\mathcal{L}$), and (iii) design of evaluation measures and loss functions (Read *et al.*, 2011).

From the point of view presented by Tsoumakas *et al.* (2011), multi-label classification algorithms might be divided into two groups, concerning the way of problem handling: (i) problem transformation methods and (ii) algorithm adaptation methods.

In the problem transformation method, multi-label classification is converted into many single-label (binary) problems making it more flexible, general and scalable as there exists a wide variety of classical classification algorithms which might be applied. Problem transformation methods may utilize any off-the-shelf single-label classifier, e.g., the $k$NN, decision trees, SVMs, naive Bayes, etc. There are relatively many different multi-label classification methods that make use of the problem transformation concept. For instance, Binary Relevance (BR), which trains one binary classifier separately for each label (see also Section 3.1), the Label Power-set (LP), where every label-set constitutes a single class-label (see Section 3.2), random $k$-label-sets (RAkEL) (Tsoumakas and Vlahavas, 2007), in which an ensemble of classifiers is trained by means of different small random subset of the label-set (cf. Section 3.3), the classifier chain (Read *et al.*, 2011) similar to binary relevance but modelling cascades of classifiers, ensembles of pruned sets (Read *et al.*, 2009), ranking by pairwise comparison (Hullermeier *et al.*, 2008), multi-label pairwise perceptron (Loza Mencia and Furnkranz, 2008), etc.

Another idea of multi-label classification handling is addressed by algorithm adaptation methods. In general, these methods adapt an existing single-label (binary class) classifier for the multi-label purpose, but such solutions are usually problem-specific. Some algorithm adaptation methods use the concept of problem transformation but only internally. Among others, one can enumerate modified decision trees (Clare and King, 2001), adapted version of conditional random fields (Ghamrawi and McCallum, 2005), a Back-propagation Perceptron for Multi-Label Learning (BP-MLL) (Zhang and Zhou, 2006), Multi-class Multi-label Perceptron (MMP) (Crammer and Singer, 2003), multi-label $k$NN (Zhang and Zhou, 2007), or AdaBoost.MH and AdaBoost.MR (Schapire and Singer, 2000).

Overall, the multi-label classification is a research problem that emerges in many application domains, among others in protein function classification (Zhang and Zhou, 2006), semantic classification of images (Boutell *et al.*, 2004), text categorization (Schapire and Singer, 2000) or repayment prediction in debt portfolio valuation (Kajdanowicz and Kazienko, 2009a; 2009b).

There exists almost no work combining ECOC methods with multi-label classification except preliminary work of the authors (Kajdanowicz *et al.*, 2011) followed by that of Ferng and Lin (2011) as well as Zhang and Schneider (2011). It should be noticed that the ideas presented by Ferng and Lin (2011) can be derived from the preceding work of Kajdanowicz *et al.* (2011) and the recent paper aligns all the achievements in ECOC support in multi-label classification. Additionally, the current paper provides competitively arranged experimental studies compared with the work of Ferng and Lin (2011) (see Section 6). However, the ECOC impact on multi-label classification has not been exhaustively studied and requires further research.

## 3. Multi-label classification

This section brings deeper insight into selected multi-label classification algorithms—the ones utilized in experiments.

**3.1. Binary relevance.** In the binary relevance approach each label constitutes a separate binary problem. Therefore, the method requires learning as many binary base classifiers as there are labels $l$ in the label-set $\mathcal{L}$. This makes the method competitive in time complexity ($\mathcal{L}$ binary models). However, the method has a huge

drawback—it does not explicitly model label correlations, which results in loosing some vital information and may provide poor prediction quality.

**3.2. Label power-set.** The idea of the label power-set classification algorithm is very simple, but this simplicity simultaneously brings some limitations. It is assumed in the algorithm that each existing combination (subset) $\lambda \in \Lambda$ of labels becomes a separate single-class and next a standard multi-class classification problem is solved. This implies that, in the worst case, the total number of distinct class-label sets may equal the minimum of two: $2^{card(\mathcal{L})}$ (the number of all combinations from the label-set $\mathcal{L}$, i.e., the quantity of the power set of $\mathcal{L}$) and the number of instances—$card(\mathbb{X})$. Nevertheless, this technique is reported to result in good performance. Special attention in application of label power-set classification should be paid to issues related with label sparsity and overfitting.

**3.3. Random *k*-label-set.** Random $k$-label-set builds an ensemble of $m$ classifiers for randomly sampled $k$-element label subsets (Tsoumakas and Vlahavas, 2007). Thanks to that the complexity is reduced to $m \times min(2k, N)$, where $N$ denotes the number of instances ($N = card(\mathbb{X})$). Each randomly sampled $k$-element label subset is used to train the label power-set classifier. After learning, the algorithm returns $m$ classifiers that in the inference phase accomplish a decision by weighted voting. The method is believed to be a robust classification technique with reasonable complexity.

## 4. Framework for multi-label classification with error correcting output codes

Multi-label classification with error correcting output codes is based on the phenomenon that the usage of a special expansion technique for information coding provides additional self-correcting abilities. If a given multi-label classification method is equipped with such additional training data pre- and post-processing, it can result in better classification quality—an improvement of the component multi-label classifier. The entire method is presented in Algorithm 1 and Fig. 1.

Therefore, the method used in the framework replaces, the original description of multiple labels assigned to the trained instances using the encoding technique. Then, instead of training the multi-label classifier on original labels, it tries to train on the encoded ones. Note that the classifier remains the multi-label one. In the inference phase, the method results in encoded classification output $y_m^{enc}$ and the final decision needs to be decoded into the original multi-label space $y_m$.

Hence, the framework provides a general method matching different coding and error correcting approaches

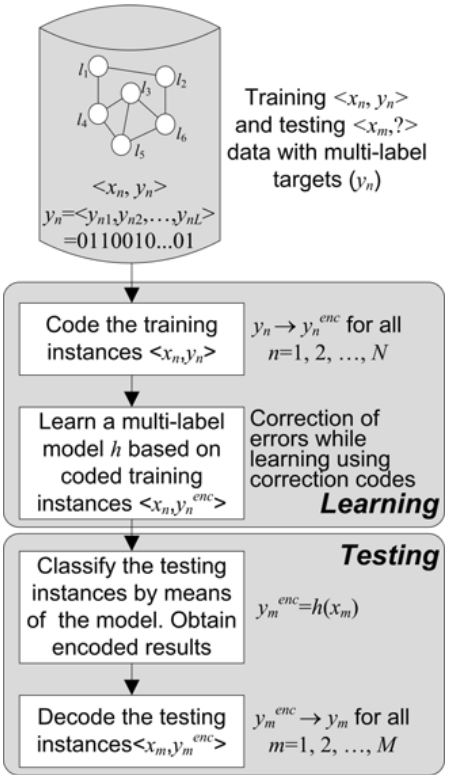


Fig. 1. Multi-label classification framework based on the coding and decoding of the label space.

(see Section 5) with different multi-label classification methods (see Section 3). Based on it, we can test whether some multi-label classification methods can really take advantage of ECOC correcting abilities. This means that even if the decision generated by the multi-label classifier in the encoded space is wrong, it can still be aligned to true results while decoding to the original multi-label space. This comes from the correction phenomenon of ECOC methods. It may eliminate, to some extent, the inability of particular multi-label classifiers to generalize properly or to overfit. On the other hand, some multi-label classification techniques may be resistant to code-based corrections. For example, experimental studies revealed that power-set and RAkELs multi-label classifications do not benefit from ECOCs, while the binary relevance method does (see Section 6).

The second property of the framework is rather its drawback. The coding of original label information always provides a higher dimensionality of the output modelled inside multi-label classifiers. This property makes the classification methods more computationally expensive and for some of them it may provide even worse accuracy.

The proposed method is in close relation to the

**Algorithm 1.** General process of multi-label classification by means of error correcting output codes.

**Require:** training dataset $D_{tr} = \{\langle x_n, y_n\rangle\}_{n=1}^{N}$, testing instances $D_{ts} = \{\langle x_m, \cdot\rangle\}_{m=1}^{M}$, ECOC design: encoder $enc(\cdot)$, decoder $dec(\cdot)$, multi-label classifier $h(\cdot)$

*Learning*
**for all** $\langle x_n, y_n\rangle \in D_{tr}$ **do**
  encode $y_n$ to $y_n^{enc} = enc(y_n)$
**end for**
learn $\hat{h} = h(\langle x_n, y_n^{enc}\rangle)$
*Inference*
**for all** $\langle x_m, \cdot\rangle \in D_{ts}$ **do**
  classify $\hat{y_m^{enc}} = \hat{h}(x_m)$
  decode $\hat{y_m} = dec(\hat{y_m^{enc}})$
**end for**

authors' previous work (Kajdanowicz *et al.*, 2011) and the currently emerging direction proposed by Ferng and Lin (2011).

## 5. Error correcting output codes in multi-label classification

Originally, error-correcting output codes were developed in the field of pattern recognition for problems with multiple classes. The idea was to avoid solving the multi-class problem directly and to decompose it into dichotomies instead. Therefore, a multi-class pattern recognition problem can be decomposed in the finite quantity of binary classification problems (Dietterich and Bakiri, 1995) (Fig. 2). Thus, the aggregated binary classifiers should be able to recognize a native set of predefined classes by dividing the pattern recognition problem into dichotomies (Hong *et al.*, 2008).

The problem of multi-class classification is decomposed to a combination of multiple binary classifications accomplished by individual binary learners. In order to obtain the final decision (a single class), the outputs from these learners need to be merged via a simple nearest-neighbour rule. It finds the class closest to the outputs of the binary classifiers (according to a given metric). This assignment to the closest neighbour may correct some errors introduced by individual binary classifiers.

The most common variations of the binary classifier combinations are one-against-one and one-against-all (Duan *et al.*, 2003). The former produces an intuitive multi-class classifier where at least one binary classifier corresponds to each class. The hypothesis that a given object belongs to the selected class is verified against its membership to one of the others. Such an approach, i.e., has a drawback in the case of conflicting answers from classifiers, which is not quite straightforward. The second approach the one-against-all method, usually uses the Winner Takes All (WTA) rule. Each classifier is trained on instances of the separate class which becomes the first class, all the other classes correspond to the second one. Final classification is made on the basis of support functions using the maximum rule.

However, all the above solutions have been applied only to the multi-class problem, which has different nature than the multi-label one (different output). Their application to the multi-label problem requires some adaptations. This especially refers to the coding/decoding method, i.e., transformation of original sets of labels $y$ (or rather their binary vector representations (see Section 1.1)). They require to be coded before the training of classifiers to enable corrections while decoding. Additionally, base multi-label classifiers applied to encoded training data should be adequately utilized to enable proper problem transformation.

As described above, the final classification was based on decomposition to several binary decisions undertaken individually but afterwards fused using the assumed coding scheme. However, the problem can be considered more generally in terms of properties of codes used to represent the classification target. The ECOC design patterns provide a wide variety of techniques that can be used to enrich the original signal (class or labels assignment) with redundancy that may partially recover individual classification errors. In multi-label classification, this settlement for each sequence of bits representing original labels $y$ forms a codeword—an encoded version $y_m^{enc}$ of $y$. Below, three basic practical coding methods (ECOC designs) are presented. The correcting abilities of selected coding methods are provided in Table 1. By $K$ we denote the length of the original message (vector representation of $y$, $K = card(\mathcal{L})$, (see Section 1.1) and by $M$ the length of the coded message ($y_m^{enc}$). More coding methods are studied, for instance, by Mackay (2003), Morelos-Zaragoza (2006), and Kuriata (2008)

**5.1. Repetition code.** The repetition code is one of the most basic error correcting output codes. Its coding idea is accomplished just by repetition of the original message (binary vector representation of $y$) several times, one after

Table 1. Correcting abilities of error correcting output codes ($K$: length of the original message, $M$: length of the coded message).

| Coding method | Max corrections |
|---|---|
| Repetition code | $\frac{1}{2}\lfloor\frac{M}{K}\rfloor - 1$ |
| BCH code | $\frac{M-K}{p}$ |
| All-pairs | $\frac{1}{2}\lfloor\frac{K-1}{2}\rfloor - 1$ |

another (Mackay, 2003). The method assumes that the transmission channel may be corrupted only in a minority of these repetitions. Each individual value of the original message ($y$) might be recovered by consideration of values in the encoded message ($y_m^{enc}$) that occur most frequently. Therefore, the decoding takes a majority vote based on all copies of the original bit.

**5.2. Bose–Chaudhuri–Hocquenghem code.** The BCH code, named according to the names of its inventors, is a cyclic polynomial code over a finite field with a particularly chosen generator polynomial (Reed and Chen, 1999). The BCH code for the length $M = 2^p - 1$ provides an ability to correct $(M - K)/p$ errors. The code is much stronger with respect to correction abilities than the repetition code, and its advantage is achieved by the more complicated decoder (Reed and Chen, 1999).

**5.3. All-pairs code and multi-label classification.** The all-pairs code has been initially proposed by the authors and applied to multi-label classification (Kajdanowicz *et al.*, 2011). In this method, each original message (a binary version of $y$) is assigned a temporal class (see Fig. 2: temporal class A, B, C corresponding to 101 (English, –, Polish), 111 (English, German, Polish), 100 (English, –, –)—binary vector representations of multi-label $y$ ($\Lambda$), respectively). Then, for such a temporal class, a dichotomy of original messages is constructed according to the chosen dichotomy construction method.

In the case of the all-pairs code, we create all possible pairs of classes, i.e., $\binom{card(\Lambda)}{2}$ pairs. There are three such pairs in Fig. 2: AB, BC, AC. Each pair of temporal classes constitutes a separate dichotomy to train a separate classifier. A dichotomy is a binary partition to '1', equivalent to a given pair (e.g., AB) and '0', the other temporal classes (refers only class C, other than AB). Separate classifiers are trained on these dichotomies on the training set. Hence, we have as many different pairs of classes, as there are multi-label classifiers i.e., $\binom{card(\Lambda)}{2}$. These classifiers provide their output that means either a given pair of temporal classes (1) or not (0), i.e., either AB or other classes (in our case only class C).

For test data, the multi-label classifiers provide a list of satisfied class pairs (AB, BC, AC), which form an output codeword (110), $y_m^{enc}$. Next, a temporal class (A, B, C) with the codeword nearest to the one just obtained is identified using the Hamming distance. It is class B with the same codeword 110 (Fig. 2). This is an ensemble of classifiers (Kuncheva, 2005). Note that codewords delivered by classifiers are not necessarily the same as temporal codewords (like in Fig. 2). The possible differences in both codewords may represent potential errors and through transformation from classifiers' output to temporal class codewords we are provided with error correcting abilities. Once a closest temporal class is discovered, its codeword is decoded into the corresponding original multiple label vector ($y_m$), 111, and assigned to the testing instance $x$.

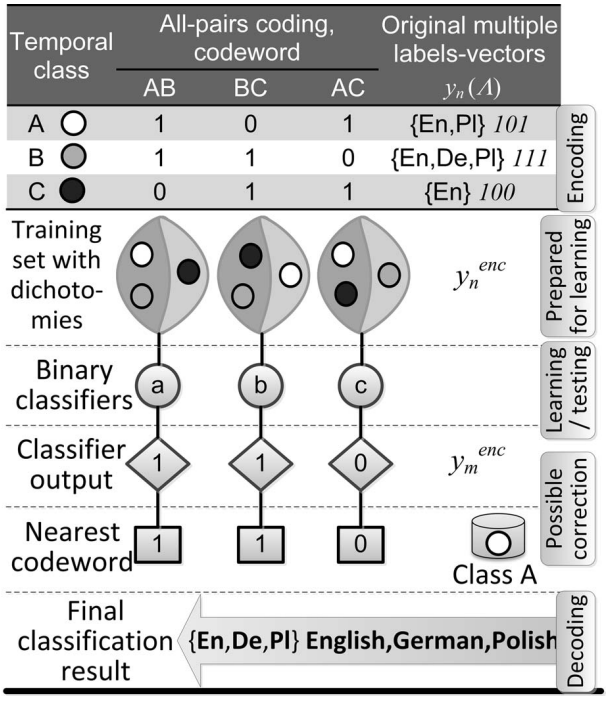


Fig. 2. Classification based on all-pairs encoding (ECOC) and binary relevance multi-label classification. Set $\mathcal{L}$ = {En—English, De—German, Pl—Polish}, set $\Lambda$ = {{En, Pl}, {En, De, Pl}, {En}}. $\Lambda$'s elements are coded to single temporal classes: {En, Pl}—A, {En, De, Pl}—B, {En}—C.

The all-pairs method generates a code with length $M = \binom{K}{2} = \frac{K^2-K}{2}$, $K = card(\mathcal{L})$. As the method generates a code with a constant minimal Hamming distance between codewords, it has the ability to recover $\frac{1}{2}\lfloor\frac{M}{K}\rfloor - 1$ bits. Combining both the above observations, all-pairs coding is able to correct $\frac{1}{2}\lfloor\frac{K-1}{2}\rfloor - 1$ errors, which makes it powerful.

## 6. Experiments and results

**6.1. Datasets.** The experiments were carried out on six distinct datasets from four diverse application domains: semantic scene analysis, bioinformatics, music categorization and text processing. The image dataset *scene* (Boutell *et al.*, 2004) semantically indexes still scenes. The biological dataset *yeast* (Elisseeff and Weston, 2001) concerns micro-array expressions and phylogenetic profiles for genes classification. In turn, the

music dataset *emotions* (Trohidis *et al.*, 2008) contains data about songs categorized into one or more classes of emotions. The *medical* (Pestian *et al.*, 2007) dataset is based on the Computational Medicine Center's 2007 Natural Medical Natural Language Processing Challenge and contains clinical free text reports labelled with disease codes. Another dataset, *enron*, is based on annotated email messages exchanged between Enron Corporation employees. The last dataset, *genbase* (Diplaris *et al.*, 2005), refers to protein classification.

The basic statistics of datasets used in experiments, such as the number of data instances, the number of attributes, the number of labels, are presented in Table 2.

**6.2. Experimental setup.** The main objective of the performed experiments was to evaluate the accuracy and efficiency of distinct multi-label classification methods over various ECOC methods. All combinations of coding schemes (four methods) and classification algorithms (three methods) were examined in terms of the Hamming loss, classification accuracy and computation time separately for six distinct datasets.

According to the nature of multi-label classification, the typical, single-label evaluation measures were insufficient and therefore some standard evaluation measures of multi-class classifiers from the previous work have been used in the experiments. The utilized measures are calculated based on the differences of the actual and predicted sets of labels over all instances in the test set. The first measure is the Hamming Loss $HL$, which was proposed by Schapire and Singer (2000) and is defined as

$$HL = \frac{1}{N}\sum_{i=1}^{N}\frac{Y_i \triangle F(x_i)}{|Y_i|}, \tag{1}$$

where $N$ is the total number of instances $x_i$ in the test set, $Y_i$ denotes the actual (real) list of labels, $F(x_i)$ is a sequence of labels predicted by the classifier and $\triangle$ stands for the symmetric difference of two vectors, which is the vector-theoretic equivalent of the exclusive disjunction in Boolean logic.

The second evaluation measure used in the experiments is the Classification Accuracy $CA$ (Ghamrawi and McCallum, 2005), defined as

$$CA = \frac{1}{N}\sum_{i=1}^{N} I(Y_i = F(x_i)), \tag{2}$$

where $N, Y_i, F(x_i)$ have the same meaning as in Eqn. (1), $I(\text{true}) = 1$ and $I(\text{false}) = 0$.

The measure $CA$ provides a very strict evaluation as it requires the predicted set of labels to be an exact match of the true set of labels.

The performance of the analysed methods was evaluated on the original training-test splitting of datasets using evaluation measures from Eqns. (1) and (2). These two metrics are accompanied with empirical computational time measured during experiments. The computational time is in fact the sum of times consumed by coding, learning, inference and decoding. Preserving the original splitting enables verification of results through independent research carried out by other scientists.

In the experiments, three multi-label classification algorithms were considered (Section 4): binary relevance, the label power-set and random $k$-label-sets, each matched separately with four ECOC methods (Section 5): the repetition code, the Bode–Chaudhuri–Hocquenghem code, the all-pairs code and *no coding*. This gave 12 combinations in total. As the examined classification methods represent the problem transformation approach to multi-label classification, they required the base learner. For that purpose, the random forest classifier was used with 200 trees and 20% of features selected randomly at every third stage. The experiments were implemented in the Matlab environment.

To enable comprehensive comparison between different method combinations (ECOC with a multi-label classifier), their results, i.e., Hamming Loss $HL$, Classification Accuracy $CA$ and processing time, were ordered from the best to the worst separately for all data sets analysed; the best was assigned the 1st position, the next—position 2, and so on. Afterwards, the Friedman test was applied to figure out if at least one of the classification methods accompanied by a particular ECOC is significantly different than the others. A statistically significant result is declared if the $p$ value is less than 0.05 (5%).

Additionally, the Wilcoxon test for pairwise comparison is performed with the $p$ value set to 0.05 to evaluate all ECOC-multi-label combinations significantly different than the others (according to the Friedman test).

Table 2. Datasets used in the experiments.

| Dataset | Instances | Attributes | Labels | Domain |
|---|---|---|---|---|
| *emotions* | 593 | 72 | 6 | music |
| *scene* | 2 407 | 294 | 6 | images |
| *yeast* | 2 417 | 103 | 14 | biology |
| *medical* | 978 | 1 449 | 45 | text |
| *enron* | 1 702 | 1 001 | 53 | text |
| *genbase* | 662 | 1 186 | 27 | biology |

**6.3. Results and discussion.** As the usage of all-pairs coding in binary relevance classification requires a large number of base classifiers (equal to the number of all pairs of distinct label patterns), it was impossible to perform experiments on datasets with more than 27 labels. Hence, processing failed for two datasets: *enron* with 53

labels and *medical*—45 labels. Note that using binary relevance with the all-pairs code in the processing of the *enron* dataset required 283,128 models to be trained. Finally the rest of the analysis was performed, in fact, on 11 ECOC–classification method combinations. We can observe that, since the code has better correcting abilities and generates longer codewords, the binary relevance technique is not able to construct so many classifiers. On the other hand—all other examined methods are not capable to benefit from strong correcting abilities of the all-pairs code, which results in worse classification quality (HL and CA)—the codes are too complex and classifiers are too weak to cover the whole label space.

All reported results which three distinct multi-label classification methods examined and four ECOCs: no coding, repetition code, the BCH code and the all-pairs code, revealed that the best accuracy in terms of Hamming loss measures is achieved for binary relevance classification method (see Figs. 3 and 4). For the *genbase* dataset (Fig. 5), binary relevance is one order of magnitude better than all label power-set and several times better than RAkEL, regardless the ECOC used. Matching binary relevance with BCH is always slightly better than all other ECOC methods. Similar results can be observed for other datasets.

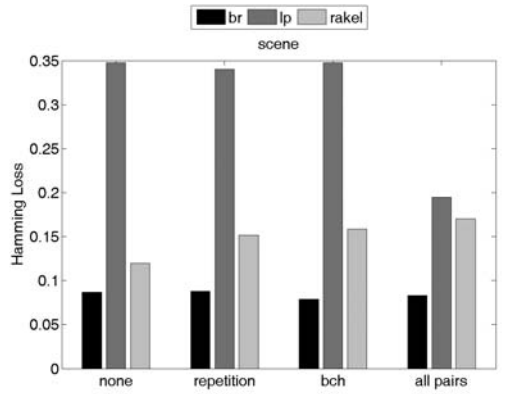


Fig. 3. Evaluation on the *scene* dataset: Hamming loss measure for *binary relevance*, *label power-set* and *RAkEL* multi-label classifiers using *none*, *repetition*, *bch* and *all-pairs* coding.

Concerning results for the classification accuracy, we can state that, for the *scene* dataset (Fig. 6), binary relevance outperforms all other multi-label classifiers and, combined with BCH and all-pairs coding, is better than with no and repetition coding. This phenomenon can be more clearly derived from Fig. 7.

While analysing computational time for the *medical* dataset (Fig. 8), combination of binary relevance and all-pairs code is not presented since it was too complex to be calculated. However, we can observe that the BCH

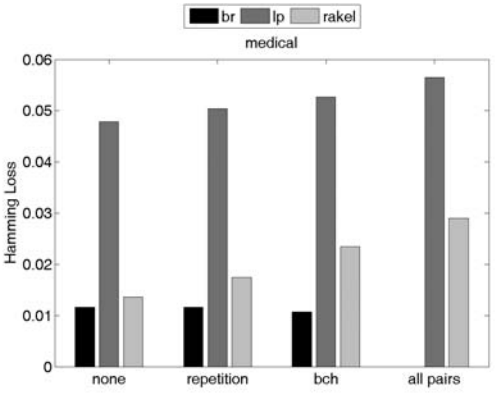


Fig. 4. Evaluation on the *medical* dataset: Hamming loss measure for *binary relevance*, *label power-set* and *RAkEL* multi-label classifiers using *none*, *repetition*, *bch* and *all-pairs* coding.

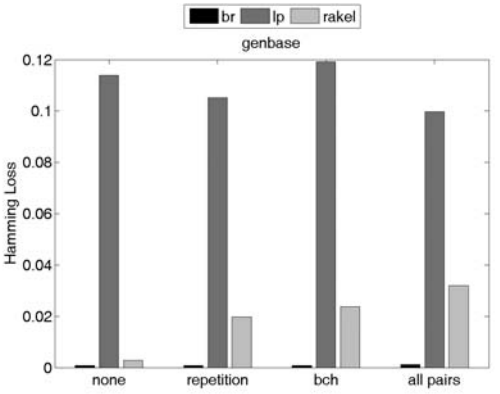


Fig. 5. Evaluation on the *genbase* dataset: Hamming loss measure for *binary relevance*, *label power-set* and *RAkEL* multi-label classifiers using *none*, *repetition*, *bch* and *all-pairs* coding.

coding makes the classification last much longer. On the other hand, the computation time for the *scene* dataset is constant for the label power-set and RAkEL regardless of the coding utilized, cf. Fig. 9. We can conclude that these two multi-label classification methods are much less dependent on code complexity while considering computational costs.

Actually, all the above findings can be confirmed by the results of tests on other datasets, although to various extent. Moreover, aggregated conclusions can be derived from the outcome of the Friedman and Wilcoxon tests performed on $HL$, $CA$ and time measures over six distinct datasets. According to the rankings in Table 3, the best combinations of the classification technique coding method measured with respect to the Hamming loss are binary relevance accompanied by

Table 3. Mean ranks over six examined datasets on the Hamming loss (HL), classification accuracy (CA) and processing time (Time) measures obtained by ECOC methods in three Friedman tests (applied separately for HL, CA, and time).

| Multi-label classifier | Code | Mean rank | | |
|---|---|---|---|---|
| | | HL | CA | Time |
| binary relevance | none | **2.0** | **2.0** | 5.33 |
| | repetition | **2.5** | **2.67** | 9.67 |
| | bch | **2.0** | **1.33** | 11.0 |
| label power-set | none | 9.5 | 7.42 | **2.5** |
| | repetition | 9.0 | 8.83 | **2.83** |
| | bch | 9.17 | 7.83 | **1.17** |
| | all-pairs | 9.83 | 8.92 | 3.83 |
| RAkEL | none | 3.67 | 5.33 | 8.17 |
| | repetition | 5.17 | 7.67 | 6.5 |
| | bch | 6.17 | 8.17 | 8.67 |
| | all-pairs | 7.0 | 5.83 | 6.33 |
| | $p$ value | $6.37{\cdot}10^{-8}$ | $4.14{\cdot}10^{-6}$ | $2.77{\cdot}10^{-8}$ |

Table 4. Results of the Wilcoxon signed rank test for HL/CA/Time.

| | BR-none | BR-rep | BR-bch | LP-none | LP-rep | LP-bch | LP-all-pairs | RAkEL-none | RAkEL-rep | RAkEL-bch | RAkEL-all-pairs |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BR-none | | ≈/≈/≈ | ≈/≈/+ | +/+/≈ | +/+/≈ | +/+/≈ | +/+/≈ | ≈/≈/≈ | ≈/≈/≈ | ≈/+/≈ | ≈/≈/≈ |
| BR-rep | ≈/≈/≈ | | ≈/≈/≈ | +/+/- | +/+/- | +/+/- | +/+/≈ | ≈/≈/≈ | ≈/≈/≈ | ≈/+/≈ | ≈/≈/- |
| BR-bch | ≈/≈/- | ≈/≈/≈ | | +/+/- | +/+/- | +/+/- | ≈/+/- | ≈/≈/≈ | ≈/+/≈ | ≈/+/≈ | ≈/≈/- |
| LP-none | -/-/≈ | -/-/+ | -/-/+ | | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/+ | ≈/≈/+ | ≈/≈/+ | ≈/≈/≈ |
| LP-rep | -/-/≈ | -/-/+ | -/-/+ | ≈/≈/≈ | | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/+ | ≈/≈/+ | ≈/≈/+ | ≈/≈/≈ |
| LP-bch | -/-/≈ | -/-/+ | -/-/+ | ≈/≈/≈ | ≈/≈/≈ | | ≈/≈/≈ | ≈/≈/+ | ≈/≈/+ | ≈/≈/+ | ≈/≈/+ |
| LP-all-pairs | -/-/≈ | -/-/≈ | ≈/-/+ | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/≈ | | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/≈ | ≈/-/≈ |
| RAkEL-none | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/- | ≈/≈/- | ≈/≈/- | ≈/≈/≈ | | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/≈ |
| RAkEL-rep | ≈/≈/≈ | ≈/≈/≈ | ≈/-/≈ | ≈/≈/- | ≈/≈/- | ≈/≈/- | ≈/≈/≈ | ≈/≈/≈ | | ≈/≈/≈ | ≈/≈/≈ |
| RAkEL-bch | ≈/-/≈ | ≈/-/≈ | ≈/-/≈ | ≈/≈/- | ≈/≈/- | ≈/≈/- | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/≈ | | ≈/≈/≈ |
| RAkEL-all-pairs | ≈/≈/≈ | ≈/≈/+ | ≈/≈/+ | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/- | ≈/+/≈ | ≈/≈/≈ | ≈/≈/≈ | ≈/≈/≈ | |

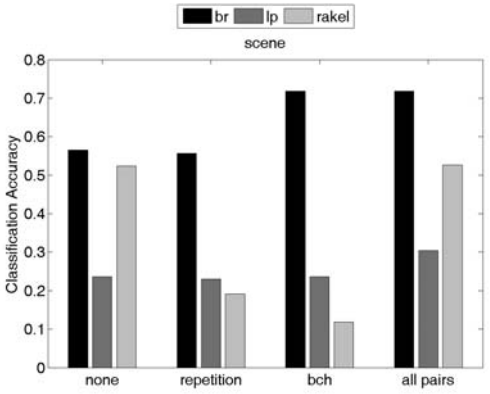


Fig. 6. Evaluation on the *scene* dataset: classification accuracy measure for *binary relevance*, *label power-set* and *RAkEL* multi-label classifiers using *none*, *repetition*, *bch* and *all-pairs* coding.

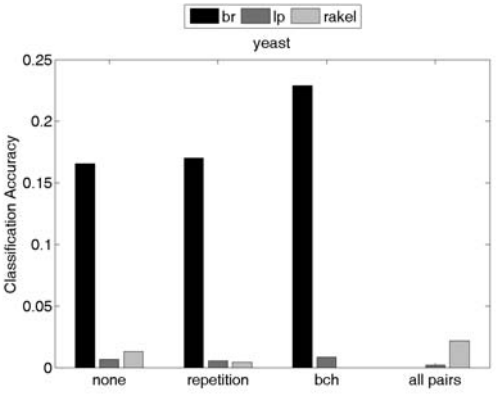


Fig. 7. Evaluation on the *yeast* dataset: classification accuracy measure for *binary relevance*, *label power-set* and *RAkEL* multi-label classifiers using *none*, *repetition*, *bch* and *all-pairs* coding.

BCH coding and with no coding (mean rank 2.0 for both). However, in the case of the classification accuracy measure, binary relevance–BCH is much better than the second one—binary relevance with no coding (1.33 versus 2.0 mean rank). Among all combinations, the third best classification setup is binary relevance with repetition code (2.5: HL and 2.67: CA). Simultaneously, binary relevance with repetition and BCH are the worst with respect to processing time (mean rank 9.67 and 11.0, respectively). This means that better quality goes with longer processing. However, binary relevance with no codding appears to be the best trade-off between quality and efficiency (as of 5.33 mean rank for time). Significance of the Friedman tests was very high since $p$ values were close to 0 (see the last row of Table 3).

The fastest methods are based on label power-set

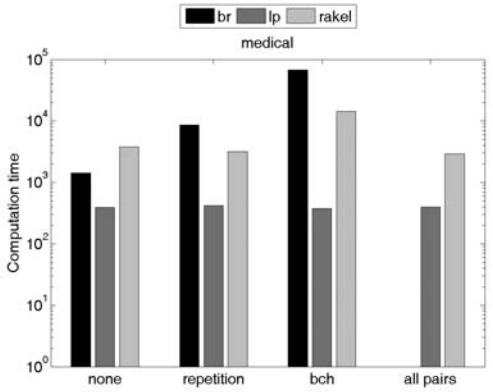


Fig. 8. Evaluation on the *medical* dataset: computation time [s] for *binary relevance*, *label power-set* and *RAkEL* multi-label classifiers using *none*, *repetition*, *bch* and *all-pairs* coding.

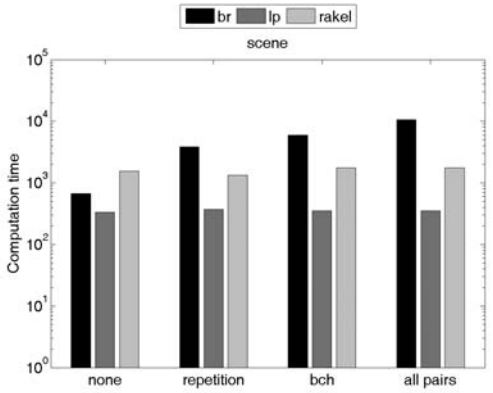


Fig. 9. Evaluation on the *scene* dataset: computation time [s] for *binary relevance*, *label power-set* and *RAkEL* multi-label classifiers using *none*, *repetition*, *bch* and *all-pairs* coding.

multi-label classification, among which the combination with the BCH code is the fastest out of all analysed (1.17 mean rank).

The results of the Wilcoxon test for pairwise comparison presented in Table 4 support the superiority of binary relevance in terms of accuracy. The acceptance of the null hypothesis stating there is no statistically significant difference between a pair of compared classification scenarios is marked with '$\approx$'. On the other hand, '$+$' and '$-$' are used to mark rejection of the null hypothesis. Intuitively, '$+$' denotes that the scenario in a row performed better than the one in the column and '$-$' stands for the opposite. The symbol '/' is used to separate the results for each of the evaluation measures: the Hamming loss, classification accuracy and computation time.

As can be seen, the binary relevance method with no ECOC is better than four other methods in HL (look for '+' symbols in the first row of Table 4) and better than five methods with respect to AC; simultaneously, it is neither better nor worse regarding the processing time. If it is combined with repetition, such combination results are the same, but it loses against three other methods in time efficiency. Matching BR with BCH coding provides even better results for accuracy (better than as many as six methods), but a bit worse for HL (better than only three methods). The worst methods are based on the label power-set classifier: worse than three other methods for HL and CA. However, they are significantly faster than five-six other tested methods.

It can be observed that the population of tested combinations is much more diverse in time efficiency than in the Hamming loss: 19 pairwise significant improvements-deteriorations out of a total 55 possible ones for time and only 11 for HL (16 for CA). This means that computational complexity of the methods is more diverse than the differences in the possible quality achievements.

Similarly to findings based on the Friedman test (Table 3), the Wilcoxon test proved that the binary relevance with no codding appears to be the best trade-off between quality and efficiency.

## 7. Conclusions

In this paper, a new framework for multi-label classification is presented. It enables combining various methods of error correcting output codes with regular multi-label classification algorithms. The main idea of the framework is based on the assumption that the base multi-label classifiers are a noisy channel and the application of ECOCs may correct classification errors made by individual classifiers.

Additionally, thorough experimental studies over binary relevance, label power-set and random $k$-label-sets classification algorithms combined with four ECOC methods: no coding, the repetition code, BCH code and the all-pairs code were performed separately on six distinct datasets.

The main conclusions from these experimental studies are the following: (i) using the proposed framework for multi-label classification with the BCH code results in better classification accuracy (according to the Hamming Loss $HL$ and Classification Accuracy $CA$) compared with solutions without any coding and it refers all classification methods; (ii) binary relevance accuracy strongly depends on the coding scheme and the best one is BCH; (iii) the label power-set and RAkEL consume the same time for computation irrespective of the coding utilized; (iv) in general, they are not suitable for ECOCs because they are not able to take advantage from

the ECOC correction capabilities; (v) the all-pairs code combined with binary relevance is not suitable for datasets with a label set larger than 27 items since it requires too many classifiers inside.

## Acknowledgment

This work was partially supported by the Polish Ministry of Science and Higher Education within a research project for the years 2011–2012 and 2011–2014, a fellowship co-financed by the European Union within the European Social Fund and the European Commission within FP7-ICT-2009-4 under the grant agreement no. 247787.

## References

Boutell, M.R., Luo, J., Shen, X. and Brown, C.M. (2004). Learning multi-label scene classification, *Pattern Recognition* **37**(9): 1757–1771.

Clare, A. and King, R.D. (2001). Knowledge discovery in multi-label phenotype data, *in* L.D. Raedt and A. Siebes (Eds.), *PKDD: 5th European Conference on Machine Learning and Knowledge Discovery*, Lecture Notes in Computer Science, Vol. 2168, Springer, Berlin/Heidelberg, pp. 42–53.

Crammer, K. and Singer, Y. (2003). A family of additive online algorithms for category ranking, *Journal of Machine Learning Research* **3**: 1025–1058.

Dietterich, T.G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes, *Journal of Artificial Intelligence Research* **2**: 263–286.

Diplaris, S., Tsoumakas, G., Mitkas, P. and Vlahavas, I. (2005). Protein classification with multiple algorithms, *in* P. Bozanis and E.N. Houstis (Eds.), *10th Panhellenic Conference on Informatics (PCI 2005)*, Lecture Notes in Computer Science, Vol. 3746, Springer-Verlag, Berlin/Heidelberg, pp. 448–456.

Duan, K., Keerthi, S.S., Chu, W., Shevade, S.K. and Poo, A.N. (2003). *Multi-Category Classification by Soft-Max Combination of Binary Classifiers*, Lecture Notes in Computer Science, Vol. 2709, Springer, Berlin/Heidelberg.

Elisseeff, A. and Weston, J. (2001). A kernel method for multi-labelled classification, *in* T.G. Dietterich, S. Becker and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14*, MIT Press, Cambridge, MA, pp. 681–687.

Ferng, C.-S. and Lin, H.-T. (2011). Multi-label classification with error-correcting codes, *Journal of Machine Learning Research* **20**: 281–295.

Ghamrawi, N. and McCallum, A. (2005). Collective multi-label classification, *in* O. Herzog, H.-J. Schek, N. Fuhr, A. Chowdhury and W. Teiken (Eds.), *International Conference on Information and Knowledge Management, CIKM*, ACM, New York, NY, pp. 195–200.

Hong, J., Min, J., Cho, U. and Cho, S. (2008). Fingerprint classification using one-vs-all support vector machines dynamically ordered with naive Bayes classifiers, *Pattern Recognition* **41**(2): 662–671.

Hullermeier, E., Furnkranz, J., Cheng, W. and Brinker, K. (2008). Label ranking by learning pairwise preferences, *Artificial Intelligence* **172**(16–17): 1897–1916.

Jankowski, N. (2012). Graph-based generation of a meta-learning search space. *International Journal of Applied Mathematics and Computer Science* **22**(3): 647–667, DOI: 10.2478/v10006-012-0049-y.

Kajdanowicz, T. and Kazienko, P. (2009a). Hybrid repayment prediction for debt portfolio, *in* N.T. Nguyen, R. Kowalczyk and S.-M. Chen (Eds.), *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, Lecture Notes in Artificial Intelligence, Vol. 5796, Springer, Berlin/Heidelberg, pp. 850–857.

Kajdanowicz, T. and Kazienko, P. (2009b). Prediction of sequential values for debt recovery, *in* E. Bayro-Corrochano and J.-O. Eklundh (Eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Lecture Notes in Computer Science, Vol. 5856, Springer, Berlin/Heidelberg, pp. 337–344.

Kajdanowicz, T., Wozniak, M. and Kazienko, P. (2011). Multiple classifier method for structured output prediction based on error correcting output codes, *in* N. Nguyen, C.-G. Kim and A. Janiak (Eds.), *Intelligent Information and Database Systems*, Lecture Notes in Computer Science, Vol. 6592, Springer, Berlin/Heidelberg, pp. 333–342.

Kuncheva, L.I. (2005). Using diversity measures for generating error-correcting output codes in classifier ensembles, *Pattern Recognition Letters* **26**(1): 83–90.

Kuriata, E. (2008). Creation of unequal error protection codes for two groups of symbols, *International Journal of Applied Mathematics and Computer Science* **18**(2): 251–257, DOI: 10.2478/v10006-008-0023-x.

Loza Mencia, E. and Furnkranz, J. (2008). Pairwise learning of multilabel classifications with perceptrons, *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN-08), Hong Kong, China*, pp. 2900–2907.

Mackay, D.J.C. (2003). *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, Cambridge.

Morelos-Zaragoza, R. (2006). *The Art of Error Correcting Coding*, Wiley, West Sussex.

Pestian, J., Brew, C., Matykiewicz, P., Hovermale, D., Johnson, N., Bretonnel Cohen, K. and Duch, W. (2007). A shared task involving multi-label classification of clinical free text, *Proceedings of ACL BioNLP*, Association of Computational Linguistics, Stroudsburg, PA.

Read, J., Pfahringer, B., Holmes, G. and Frank, E. (2009). Classifier chains for multi-label classification, *13th European Conference on Principles and Practice of Knowledge Discovery in Databases/20th European Conference on Machine Learning, Bled, Slovenia*, pp. 254–269.

Read, J., Pfahringer, B., Holmes, G. and Frank, E. (2011). Classifier chains for multi-label classification, *Machine Learning* **85**(3): 333–359.

Reed, I.S. and Chen, X. (1999). *Error-Control Coding for Data Networks*, Kluwer Academic Publishers, Norwell, MA.

Sammut, C. and Webb, G.I. (2011). *Encyclopedia of Machine Learning*, Springer, Berlin/Heidelberg.

Schapire, R.E. and Singer, Y. (2000). Boostexter: A boosting-based system for text categorization, *Machine Learning* **39**(2/3): 135–168.

Trohidis, K., Tsoumakas, G., Kalliris, G. and Vlahavas, I. (2008). Multilabel classification of music into emotions, *9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, PA, USA*, pp. 325–330.

Tsoumakas, G., Katakis, I. and Vlahavas, I. (2011). Random k-labelsets for multilabel classification, *IEEE Transactions on Knowledge and Data Engineering* **23**(7): 1079–1089.

Tsoumakas, G. and Vlahavas, I. (2007). *Random k-labelsets: An Ensemble Method for Multilabel Classification*, Lecture Notes in Artificial Intelligence, Vol. 4701, Springer, Berlin/Heidelberg.

Zhang, M.-L. and Zhou, Z.-H. (2006). Multilabel neural networks with applications to functional genomics and text categorization, *IEEE Transactions on Knowledge and Data Engineering* **18**(10): 1338–1351.

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

Zhang, Y. and Schneider, J. (2011). Multi-label output codes using canonical correlation analysis, *Journal of Machine Learning Research* **15**: 873–882.

**Tomasz Kajdanowicz** received his M.Sc. degree from the Wrocław University of Technology, Poland, in 2008. Currently he is finalizing his Ph.D. studies at the Wrocław University of Technology. He serves as a researcher in various research projects and simultaneously acts as an independent consultant working with IT companies in Poland. He was the organizing chair of the international workshops *MMAML'11* and *MMAML'12*. He has also served as a member of international programme committees and a reviewer for international journals and scientific conferences. His research interests focus on social network analysis and hybrid information systems as well as their applications, especially in the industry. While participating in multiple research and development projects, he collaborates with leading financial enterprises in Poland. He has authored more than thirty scientific papers and articles.

**Przemyslaw Kazienko** received his M.Sc. and Ph.D. degrees in computer science with honours, both from the Wrocław University of Technology, Poland, in 1991 and 2000, respectively. He obtained his habilitation degree from the Silesian University of Technology, Poland, in 2009. Presently, he serves as a professor of the Wrocław University of Technology at the Institute of Informatics. He has been also a research fellow at the Intelligent Systems Research Centre, British Telecom, UK, in 2008. For several years, he held the position of the deputy director for development at the Institute of Applied Informatics. He was a co-chair of many international scientific events and a guest editor of several special issues in JCR-listed journals. He is a member of the editorial board of *Social Network Analysis and Mining*, the *International Journal of Knowledge Society Research*, the *International Journal of Human Capital and Information Technology Professionals*, as well as *Social Informatics*. He has authored over 130 peer-reviewed papers on a variety of topics related to multiple model classification, collective classification and relational learning, social and complex network analysis, knowledge management, collaborative systems, data mining, recommender systems, information retrieval, data security, and system integration. He has also initialized and led over 25 projects chiefly in cooperation with commercial companies, including large international corporations.

Received: 10 September 2011
Revised: 12 April 2012