Dynamic classifier chains for multi-label learning

Pawel Trajdos · Marek Kurzynski

Abstract In this paper, we deal with the task of building a dynamic ensemble of chain classifiers for multi-label classification. To do so, we proposed two concepts of classifier chains algorithms that are able to change label order of the chain without rebuilding the entire model. Such modes allows anticipating the instance-specific chain order without a significant increase in computational burden. The proposed chain models are built using the Naive Bayes classifier and nearest neighbour approach as a base single-label classifiers. To take the benefits of the proposed algorithms, we developed a simple heuristic that allows the system to find relatively good label order. The heuristic sort labels according to the label-specific classification quality gained during the validation phase. The heuristic tries to minimise the phenomenon of error propagation in the chain. The experimental results showed that the proposed model based on Naive Bayes classifier the above-mentioned heuristic is an efficient tool for building dynamic chain classifiers.

Keywords multi-label, classifier chains, naive bayes, dynamic chains, nearest neighbour

1 Introduction

Under well-known single-label classification framework, an object is assigned to only one class which provides a full description of the object. However, many real-world datasets contain objects that are assigned to different categories at the same time. All of these categories constitute a full description of the object. Omitting of one of these concepts induces a loss of information. Classification process in which such kind of data is involved is called multi-label classification [14]. A great example of a multi-label dataset is a gallery of tagged photos. Each photo may be described using such tags as mountains, sea, forest, beach, sunset, etc. Multi-label classification is a relatively new idea that is explored extensively for last
two decades. As a consequence, it was employed in a wide range of practical applications including text classification [22], multimedia classification [33] and bioinformatics [15] to name a few.

Multi-label classification algorithms can be broadly partitioned into two main groups i.e. dataset transformation algorithms and algorithm adaptation approaches [14].

Methods belong to the group of algorithm adaptation approaches provides a generalisation of an existing multi-class algorithm. The generalised algorithm is able to solve multi-label classification problem in a direct way. Among the others, the most known approaches from this group are: multi label KNN algorithm [22], the Structured SVM approach [9] or deep-learning-based algorithms [43].

In this paper, we investigate only dataset transformation algorithms that decompose a multi-label problem into a set of single-label classification tasks. To reconstruct a multi-label response, during the inference phase, outputs of the underlying single-label classifiers are combined in order to create a multi-label prediction.

Let’s focus on one of the simplest decomposition methods. That is the binary relevance (BR) approach that decomposes a multi-label classification task into a set of one-vs-rest binary classification problems [1]. This approach assumes that labels are conditionally independent. However the assumption does not hold in most of real-life recognition problems, the BR framework is one of the most widespread multi-label classification methods [42]. This is due to its excellent scalability and acceptable classification quality [25].

To preserve scalability of BR systems, and provide a model of inter-label relations, Read et al. [31,32] provided us with the Classifier Chain model (CC) which establish a linked chain of modified one-vs-rest binary classifiers. The modification consists of an extension of the input space of single-label classifiers along the chain sequence. To be more strict, for a given label sequence, the feature space of each classifier along the chain is extended with a set of binary variables corresponding to the labels that precede the given one. The model implies that, during the training phase, input space of given classifier is extended using the ground-truth labels extracted from the training set. During the inference step, due to lack of the ground-truth labels, we employ binary labels predicted by preceding classifiers. The inference is done in a greedy way that makes the best decision for each of considered labels. That is, the described approach passes along the chain, information allowing CC to take into account inter-label relations at the cost of allowing the label-prediction-errors to propagate along the chain [32]. This way of performing classification induces a major drawback of the CC system. That is, the CC classifier uses a kind of greedy strategy during the inference phase. This design allows classification errors to propagate along the chain. As a consequence, the performance of a chain classifier strongly depends on chain configuration [34]. To overcome these effects, the authors suggested to generate an ensemble of chain classifiers (ECC). The ensemble consists of classifiers trained using different label sequences [34].

The originally proposed ECC ensemble uses randomly generated label orders. Additionally, each chain classifier is built using a resampled dataset. This approach provides an additional diversity into the ensemble classifier. This simple, yet effective approach allows improving the classification quality significantly in comparison to single chain classifier. However, the intuition says that there is still room for improvement when we employ a more data-driven approach.
Indeed, later research shows that the members of the ensemble may be chosen in such a way that provides further improvement of classification quality. That is, Read et al. proposed a strategy which uses Monte Carlo sampling to explore the label sequence space in order to find a classifier chain that offers the highest classification quality [29]. Another approach was proposed by Liu et al. [24]. They introduced a method that builds a model of inter-label relations as a directed acyclic graph (DAG). The weights of the graph are calculated using confusion and support for each pair of labels. Then, the ensemble is generated using topological sorting of the graph. Chen et al. [5] proposed a method that makes clusters of labels. Then, for each cluster of labels, an undirected graph describing inter-label relations is built. Then, a minimum spanning tree is created for the graph. After that, breadth-first search algorithm determines sequences for a cluster-specific ensemble of CC classifiers. A similar approach was proposed by Huang et al [21]. They proposed to build the clusters using a meta-space that mixes input space and label space. Then inter-label relations are modeled using correlation. The model is expressed using DAG structure. Finally, the CC classifier is built for each cluster. The chain structure may also be induced using Bayesian Network approach [48].

Chain sequence can be also found using meta-heuristic approach. That is, Goncalves et al. developed a strategy that utilises a genetic algorithm (GA) to find a good chain structure for the entire dataset [16,15]. The proposed approach using wrapper-based approach. That is each chromosome codes different chain order. To evaluate those label orders each corresponding classifier must be built and evaluated using a validation set. A similar approach was also used by Trajdos and Kurzynski who proposed to use a multi-objective genetic algorithm to optimize classification quality and chain diversity simultaneously [41]. Although those methods are rather time-consuming, they provide a significant improvement in terms of classification quality.

Another way of dealing with the error propagation is to build a classifier that combines CC algorithm BR-based approach [27]. The authors proposed a stacking based architecture to combine the above-mentioned classifiers. That is, the first layer is a simple BR classifier that predicts each label separately. The attribute set of the classifiers from the second layer is extended using all labels except the predicted one. During the training phase, both layers are trained separately. During the prediction phase, on the other hand, classifiers from the second layer mix outcomes of the BR classifiers with the outcomes provides by preceding classifiers. That is, the first classifier of the chain structure has its attribute space extended by the outcomes of the BR classifier. The second one uses the prediction of the first one and the remaining attributes are taken from the prediction of the BR classifier. Finally, the last classifier along the chain has the attribute space extended using only labels predicted by the preceding classifiers. Another way of combining the CC classifier with the BR classifier is described in [26].

The previously cited methods build ensemble structure during the training procedure. Consequently, throughout this paper, this kind of methods will be called static methods. The dynamic chain classifiers, on the other hand, determines the best label order at the prediction phase [35]. The above-mentioned classifier produces a set of randomly generated label sequences and then validates the chain classifiers. During the validation phase, each point from the validation set is assigned with a label order that produces the most accurate output vector for this
point. As the experimental research shows, the dynamic methods of building a label order may achieve better classification quality \cite{35}.

We observed that during the building of a dynamic chain classifier, multiple chain classifiers must be learned. These classifiers are built using the same training set and differ only in chain order. As a consequence, the computational burden of the algorithm may be reduced if there exists a classifier that is trained once and changing the label sequence is done without rebuilding the model. To address this issue, we built two models based on the Naive Bayes \cite{19} approach and the nearest neighbour approach \cite{8} that meet the above-mentioned properties.

Additionally, we proposed a dynamic method of determining the chain order based on classification quality for each label separately.

A part of this paper was previously published in \cite{40}. This paper is an extended version of the previously-published work. The main elements that has been changed/extended:

- The literature review has been extended.
- We have added the results of the experimental comparison of the BR and CC versions of different base classifiers.
- We have proposed a new model of the dynamic chain classifier. That is, we introduce the CC model based on the nearest neighbour approach.
- New experimental results have been provided.

The rest of the paper is organised as follows. Next Section 2 provides a formal description of the multi-label classification problem and describes the developed algorithms. Section 3 contains a description of the conducted experiments. The results are presented and discussed in Section 4. Finally, Section 5 concludes the paper.

2 Proposed Methods

In this section, we introduce a formal notation of multi-label classification problem and provide a description of the proposed method.

2.1 Preliminaries

Under the multi-label (ML) formalism a $d$-dimensional object $x = [x_1, x_2, \ldots, x_d] \in \mathcal{X}$ is assigned to a set of labels indicated by a binary vector of length $L$: $y = [y_1, y_2, \ldots, y_L] \in \mathcal{Y} = \{0, 1\}^L$, where $L$ denotes the number of labels.

In this paper, we follow a statistical classification framework. As a consequence, it is assumed that object $x$ and its set of labels $y$ are realizations of corresponding random vectors $\mathbf{X} = [X_1, X_2, \ldots, X_d]$, $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_L]$ and the joint probability distribution $P(\mathbf{X}, \mathbf{Y})$ on $\mathcal{X} \times \mathcal{Y}$ is known.

Because the above-mentioned assumption is never meet in real world, in this study, we assume that multi-label classifier $h : \mathcal{X} \mapsto \mathcal{Y}$, which maps feature space $\mathcal{X}$ to the set $\mathcal{Y}$, is built in a supervised learning procedure using the training set $\mathcal{T}$ containing $N$ pairs of feature vectors $x$ and corresponding class labels $y$:

$$\mathcal{T} = \{(x^1, y^1), (x^2, y^2), \ldots, (x^N, y^N)\}.$$  (1)
2.2 Naive Bayes Classifier for Dynamic Classifier Chains

In this paper, we consider ML classifiers build according to the chain rule. That is, the classifier $h$ is an ensemble of $L$ single-label classifiers $\psi_l$ that constitutes a linked chain which is built according to a permutation of label sequence $\pi$. As it was mentioned earlier, in this paper we follow the statistical classification framework. Consequently, each single-label classifier $h_{\pi(i)}$ along with the chain makes its decision according to the following rule:

$$h_{\pi(i)}(x) = \text{argmax}_{y \in \{0, 1\}} P(Y_{\pi(i)} = y | B_{\pi(i)}(x)),$$

(2)

where $B_{\pi(i)}(x)$ is a random event defined below:

$$B_{\pi(i)}(x) = (X = x, Y_{\pi(i-1)} = h_{\pi(i-1)}(x)Y_{\pi(i-2)} = h_{\pi(i-2)}(x), \ldots, Y_{\pi(1)} = h_{\pi(1)}(x)), \forall i \in \{2, 3, \ldots, L\},$$

(3)

and for $i = 1$

$$B_{\pi(1)}(x) = (X = x).$$

(4)

Conditioning on the random event $B_{\pi(i)}(x)$ instead of $X = x$ allows the chain to take inter-label dependencies into account. The above-mentioned classification rule is a greedy rule that calculates the probability (2) using predictions of preceding classifiers. The optimal prediction under the chaining rule may be found using the PCC approach [6]. However, the approach requires the number of calculations that grows exponentially with the number of labels.

The probability defined in (2) is then computed using the Bayes rule:

$$P(Y_{\pi(i)} = y | B_{\pi(i)}(x)) = \frac{P(Y_{\pi(i)} = y) P(B_{\pi(i)}(x) | Y_{\pi(i)} = y)}{P(B_{\pi(i)}(x))}. $$

(5)

The term $P(B_{\pi(i)}(x))$ does not depend on event $Y_{\pi(i)} = y$. Consequently, the decision rule (2) is rewritten:

$$h_{\pi(i)}(x) = \text{argmax}_{y \in \{0, 1\}} P(Y_{\pi(i)} = y) P(B_{\pi(i)}(x) | Y_{\pi(i)} = y)$$

(6)

Now, to improve the readability we simplify the notation:

$$P(B_{\pi(i)}(x) | Y_{\pi(i)} = y) = P(B_{\pi(i)}(x) | y).$$

(7)

Then, following the Naive Bayes rule, we assume that all random variables that constitute $B_{\pi(i)}(x)$ are conditionally independent given $Y_{\pi(i)} = y$. Consequently, $P(B_{\pi(i)}(x) | y)$ is defined using the following formula:

$$P(B_{\pi(i)}(x) | y) = \prod_{m=1}^{d} P(X_m = x_m | y) \prod_{l=1}^{i} P(Y_{\pi(l)} = h_{\pi(l)}(x) | y).$$

(8)

Now, it is easy to see that the term $\prod_{m=1}^{d} P(Y_{\pi(l)} = h_{\pi(l)}(x) | y)$, contrary to $\prod_{m=1}^{d} P(X_m = x_m | y)$, depends on the chain structure. Furthermore, all probability distributions used in the above-mentioned terms can be estimated during the training phase when the chain structure is unknown.

The training and inference phases are described in detail using pseudocode shown in Algorithms [4] and [2].
In this section, we assess the increase in computational complexity that the proposed algorithm causes.

First of all, it is easy to see that for both the original and the proposed algorithm the number of estimators that must be built to assess \( P(X_m|Y_{\pi(i)} = y) \forall i \in \{1, 2, \cdots, L\}, y \in \{0, 1\} \) is \( 2Ld \).

The number of estimators of \( P(Y_{\pi(i)}|Y_{\pi(i)} = y) \) that must be built is also the same for both classifiers: \( L \).

The key difference is in the number of estimators of \( P(Y_{\pi(i)}|Y_{\pi(i)} = y) \) that must be built. For the original CC classifier the number of estimators that is built is \( L(L - 1) \). On the other hand our method builds \( 2L^2 \) estimators.

At the inference phase, the only additional calculations are performed to determine the permutation of labels. Since the validation set is involved in this process, a number of calculations is proportional to \( O(|\mathcal{V}|L) \).
Algorithm 3 Pseudocode of the learning procedure of the nearest neighbour classifier.

Input data:
\( T \) - training set;

BEGIN
Split \( T \) into \( T_A \) and \( V \) so that:
\[ |T_A| = t|T| \text{ and } |V| = (1-t)|T|, \quad t \in (0,1) \]
\( T_A \cap V = \emptyset \);
Save the training set \( T_A \)
END

2.3 KNN Classifier for Dynamic Classifier Chains

In this section, we define a dynamic classifier chain algorithm based on the nearest neighbours approach. The nearest neighbour algorithm is an instance-based classifier that does not build an explicit model of mapping between the feature space and the label space. Instead, the classifier performs classification in a lazy manner. That is, the \( R \) nearest instances and then the class is predicted using labels of the neighbour instances.

Let's begin with the definition of a distance function that depends on label permutation and the position along the chain. The distance function is defined in the extended feature space that combines the input space and the label space. For the first position, the distance is a simple Euclidean distance in the input space:

\[
\delta_{\pi,i} \left( \left( x, y \right), \left( x', y' \right) \right) = \sqrt{\sum_{j=1}^{d} (x_j - x'_j)^2}. \quad (9)
\]

For the other positions, the distance function uses both the input space and the label space:

\[
\delta_{\pi,i} \left( \left( x, y \right), \left( x', y' \right) \right) = \sqrt{\sum_{j=1}^{d} (x_j - x'_j)^2 + \sum_{l=1}^{i-1} (y_{\pi(l)} - y'_{\pi(l)})^2}. \quad (10)
\]

Such defined distance function allows us to make the prediction using chaining rule. Since the distance is modified in order to fit the chain structure. During the inference phase, the distance calculates the extended distance using labels predicted at the preceding steps of the procedure. The above-defined distance function is used to build the neighbourhood of a given point in the extended feature space: \( M^R_{\pi,i}(\left( x, y \right)) \). The neighbourhood contains the \( R \) closest instances selected from the training set according to the distance function \( \delta_{\pi,i} \).

Given the neighbourhood, the probability \( P(Y_{\pi(i)} = y | B_{\pi(i)}(x)) \) is estimated as follows:

\[
P(Y_{\pi(i)} = y | B_{\pi(i)}(x)) \approx \frac{\left\{ \left( x^n, y^n \right) | (x^n, y^n) \in M^R_{\pi,i}(\left( x, h(x) \right)), y^n_i = y \right\}}{R}. \quad (11)
\]

The label \( \pi(i) \) is also predicted using rule \( \mathbb{R} \).

The training procedure is described in Algorithm 3. The procedure is very simple. That is, it splits the original training set into actual training set \( T_A \) and the validation set \( V \).
The inference procedure begins with assigning undefined values $\emptyset$ into the prediction vector $h(x)$. Then the predictions are updated sequentially according to the permutation $\pi$. The precedure is shown in Algorithm 4.

2.4 Dynamic Chain order

In this subsection, we define a local measure of classification quality. To do so, we employed a modified version of the well-known $F_1$ measure.

First of all, we defined a fuzzy neighbourhood in the input space. The neighbourhood of an instance $x$ is defined using the following fuzzy set [17]:

$$N(x) = \{(x^n, y^n, \mu(x, x^n)) : (x^n, y^n) \in V\}$$

(12)

where each tripplet $(x^n, y^n, \zeta)$ defines fuzzy set with the membership coefficient $\zeta$. The membership function $\mu(x, x^n)$ is defined using gaussian potential function:

$$\mu(x, x^n) = \exp(-\beta \delta(x, x^n)^2).$$

(13)

The above-mentioned classifier responses are related to the binary relevance classifier that can be built without knowing the order of the chain. The classifier is defined using the following classification rule:

$$h^{BR}_{\pi(i)}(x) = \arg\max_{y \in \{0, 1\}} P(Y_{\pi(i)} = y) \cdot \prod_{m=1}^{d} P(X_m = x_m | Y_{\pi(i)} = y)$$

(16)
Since the neighbourhood of a given instance is defined as a fuzzy set, consistently the above-mentioned sets are also defined as fuzzy. However, the sets are fuzzy singletons. The visualisation of aforementioned sets is provided in Fig. 1.

Using the above-mentioned sets we define local True Positive rate, False Positive rate, False Negative rate respectively:

\[
TP_l(x) = |\mathcal{V}_l \cap \mathcal{D}_l \cap \mathcal{N}(x)|, \\
FP_l(x) = |(\mathcal{D}_l \setminus \mathcal{V}_l) \cap \mathcal{N}(x)|, \\
FN_l(x) = |(\mathcal{V}_l \setminus \mathcal{D}_l) \cap \mathcal{N}(x)|,
\]

where \(|·|\) is the cardinality of a fuzzy set \([11]\). Then, we define the local measure of classification quality:

\[
F_l(x) = \frac{2TP_l(x)}{2TP_l(x) + FP_l(x) + FN_l(x)}
\]

Finally, the label order \(\pi\) is chosen so that the following inequalities are met:

\[
F_{\pi(1)}(x) \geq F_{\pi(2)}(x) \geq \cdots \geq F_{\pi(L)}(x).
\]

That is labels for whom the classification quality is higher precedes other labels in the chain structure. In other words, this simple heuristic is aimed at dealing with error propagation in the chain structure by employing the most accurate models at the beginning of the chain. A similar approach based on the Jaccard quality criterion was proposed in [36].

2.5 The Ensemble Classifier

Now, let us define a ML \(K\) - element classifier ensemble: \(eH = \{H_1, \ldots, H_K\}\). The ensemble is built using classifier chain algorithms defined in previous sections. Each ensemble classifier is built using a subset of the original dataset. The size of subset is 66% of the original training set.

The BR transformation may produce imbalanced single-label dataset. To prevent the classifier from learning from a highly imbalanced dataset, we applied the random undersampling technique [13]. The majority class is undersampled when imbalance ratio is higher than 20. The goal of undersampling is to keep the imbalance ratio at the level of 20.

The research on the application of Naive Bayes algorithm under the CC framework shows that when the number of features in the input space is significantly higher in comparison to the number of labels the Naive Bayes classifier may not perform well [35]. To prevent the proposed system from being affected by this
phenomenon, we applied the feature selection procedure for each single-label separately. That is, the attributes are selected in order to improve the classification quality for given label. The feature selection removes only attributes related to the original input space. Features related to labels are passed through the chain without selection. We employed the selection procedure based on correlation. In other words, we select attributes that are highly correlated to the predicted label and their inter-correlations are low [18]. Additionally, if the number of selected features is higher than 300, we select 300 random features from the set of previously selected features.

The final prediction vector of the ensemble is obtained via a simple averaging of response vectors corresponding to base classifiers of the ensemble followed by the thresholding procedure:

\[
\tilde{h}_i(x) = \left\lfloor K^{-1} \sum_{k=1}^{K} h^k_i(x) > 0.5 \right\rfloor,
\]

where \(\lfloor \cdot \rfloor\) is the Iverson bracket.

3 Experimental Setup

The experimental study is divided into three main sections. The first one assesses the impact of employing chaining approach. In the section, we compare binary relevance and classifier chains algorithms built using the following base classifiers:

- J48 Classifier (C4.5 algorithm implemented in Weka) [28].
- SVM algorithm with radial based kernel [7, 2].
- Naive Bayes Classifier [19].
- Nearest Neighbour classifier [8].

In this section, we compare BR and CC ensembles built using a genetic algorithm tailored to optimise the macro-averaged \(F_1\) loss. For each ensemble, the size of the committee was set to \(K = 20\). For the algorithm based on the genetic algorithm, the initial size of the committee was set to \(3K\). Each numeric attribute in the training and validation datasets was also standardised. After the standardisation, the mean value of the attribute is 0 and its standard deviation is 1.

During the experimental study, the parameters of the SVM classifier \((C \in \{0.001, 1, 2, \ldots, 10\}, \gamma \in \{0.001, 1, 2, \ldots, 5\})\) were tuned using grid search and 3-fold cross validation. The number of nearest neighbours was also tuned using 3-fold cross validation. The number of neighbours was chosen among the following values \(R \in \{1, 3, 5, \ldots, 11\}\).

In two remaining sections, the conducted experimental study provides an empirical evaluation of the classification quality of the proposed methods and compares it to reference methods. Namely, we conducted our experiments using the following algorithms of building a CC ensemble:

1. The proposed approach (Section 2.4).
2. Static ensemble generated using a genetic algorithm [41]. The ensemble is tuned to optimise the macro-averaged \(F_1\) measure.
3. ECC ensemble with randomly generated chain orders [31].
4. OOCC dynamic method proposed by da Silva et al. [35]. The ensemble is tuned to optimise the example based $F_1$ measure. Additionally, the reference method uses single split into training and validation sets.

The above-mentioned methods of building CC systems were evaluated using Naive Bayes and the nearest neighbour algorithms as base classifiers. Systems built using different base classifiers are investigated in two separate sections. In the sections, we will refer to the investigated algorithms using the above-said numbers.

The reference algorithm also uses Naive Bayes/nearest neighbour algorithm with data preprocessing procedures described in Section 2.5.

The extraction of training and test datasets was performed using 10 fold cross-validation. For each ensemble, the proportion of the training set $T_A$ was fixed at $t = 0.6$ of the original training set (see Algorithm 1). For each ensemble, the size of the committee was set to $K = 20$. For the algorithm based on the genetic algorithm, the initial size of the committee was set to $3K$. Each numeric attribute in the training and validation datasets was also standardised. After the standardisation, the mean value of the attribute is 0 and its standard deviation is 1.

The $\beta$ coefficient was tuned during the training procedure using 3 CV approach. The best value among $\{1, 2, \ldots, 10\}$ is chosen.

Single label classifiers were implemented using WEKA software [17]. Multi-label classifier were implemented using Mulan software [37].

The experiments were conducted using 30 multi-label benchmark sets. The main characteristics of the datasets are summarized in Table 1. We used datasets from the sources abbreviated as follows: A [4], B [30], M [38], W [45], X [46], Z [49], T [39], O [23]. Some of the employed sets needed some preprocessing. That is, we used multi-label multi-instance [49] sets (sources Z and W) which were transformed to single-instance multi-label datasets according to the suggestion made by Zhou et al. [49]. Multi-target regression sets (No 9, 30) were binarised using simple thresholding strategy. That is if the response is greater than 0 the resulting label is set relevant. Two of the used datasets are synthetic ones (source T) and they were generated using algorithm described in [39]. To reduce the computational burden, we use only a subset of original Tmc2007 and IMDB sets. Additionally, the number of labels in Stackex datasets is reduced to 15.

The algorithms were compared in terms of 11 different quality criteria coming from three groups [25]: Instance-based (Hamming, Zero-One, $F_1$, False Discovery Rate, False Negative Rate); Label-based. The last group contains the following measures: Macro Averaged (False Discovery Rate (FDR, 1- Precision), False Negative Rate (FNR, 1-Recall), $F_1$) and Micro Averaged versions of the above-mentioned criteria.

Statistical evaluation of the results was performed using the Wilcoxon signed-rank test [11] and the family-wise error rates were controlled using the Holm procedure [11]. For all statistical tests, the significance level was set to $\alpha = 0.1$. Additionally, we also applied the Friedman [12] test followed by the Nemenyi post-hoc procedure [10].
### Table 1

| No | Name       | Sr | N  | d  | L  | LC | LD | avIR |
|----|------------|----|----|----|----|----|----|------|
| 1  | Arts1      | M  | 7484 | 1743 | 20 | 1.654 | 0.964 | 94.74 |
| 2  | Azotobacter| W  | 407  | 20  | 13 | 1.469 | 0.113 | 2.225 |
| 3  | Birds      | M  | 645  | 266  | 19 | 1.014 | 0.053 | 5.407 |
| 4  | Caenorhabditis | W | 2512 | 20  | 22 | 2.419 | 0.115 | 2.347 |
| 5  | Drosophila | W  | 2005 | 20  | 22 | 2.656 | 0.121 | 1.744 |
| 6  | Emotions   | M  | 593  | 72   | 6  | 1.868 | 0.311 | 1.478 |
| 7  | Enron      | M  | 1702 | 1001 | 53 | 3.378 | 0.063 | 73.95 |
| 8  | Flags      | X  | 194  | 43   | 7  | 3.392 | 0.485 | 2.655 |
| 9  | Flare2     | M  | 1066 | 27   | 3  | 0.289 | 0.707 | 14.15 |
| 10 | Genbase    | M  | 662  | 1186 | 27 | 1.252 | 0.046 | 37.32 |
| 11 | Geobacter  | W  | 379  | 20   | 11 | 1.264 | 0.115 | 2.750 |
| 12 | Halaerula  | W  | 304  | 20   | 13 | 1.692 | 0.123 | 2.419 |
| 13 | Human      | X  | 3106 | 440  | 14 | 1.185 | 0.085 | 15.29 |
| 14 | Image      | M  | 2000 | 284  | 5  | 1.126 | 0.247 | 1.193 |
| 15 | IMDB       | M  | 3042 | 1001 | 28 | 1.987 | 0.071 | 24.61 |
| 16 | LLOG       | B  | 1460 | 1004 | 75 | 1.180 | 0.016 | 39.27 |
| 17 | Medical    | M  | 978  | 1449 | 45 | 1.245 | 0.028 | 89.50 |
| 18 | Mollfrog   | Z   | 2000 | 135  | 5  | 1.236 | 0.247 | 1.193 |
| 19 | Ohsumed    | O   | 13929| 1002 | 23 | 1.663 | 0.072 | 7.869 |
| 20 | Plant      | X   | 978  | 440  | 12 | 1.079 | 0.090 | 6.990 |
| 21 | Pyrocococcus| W  | 425  | 20   | 18 | 2.130 | 0.119 | 2.421 |
| 22 | Saccharomyces| W | 3509 | 20   | 27 | 2.275 | 0.084 | 2.077 |
| 23 | Scene      | X   | 2407 | 294  | 6  | 2.205 | 0.179 | 1.254 |
| 24 | SimpleHC   | T   | 3000 | 30   | 10 | 1.900 | 0.010 | 1.388 |
| 25 | SimpleHS   | T   | 3000 | 30   | 10 | 2.207 | 0.251 | 2.022 |
| 26 | SLASHDOT   | B   | 3782 | 1079 | 22 | 1.181 | 0.054 | 17.69 |
| 27 | Stackex_chess| A | 1675 | 585  | 15 | 1.137 | 0.076 | 4.744 |
| 28 | Tmc2007-500| M   | 2857 | 500  | 22 | 2.222 | 0.101 | 17.15 |
| 29 | water-quality | M   | 1060 | 16   | 14 | 5.073 | 0.362 | 5.967 |
| 30 | yeast      | M   | 2417 | 103  | 14 | 4.237 | 0.303 | 7.197 |

4 Results and Discussion

4.1 Assessing the impact of chaining approach

In this section, we assess the consequences of employing chaining approach. That is, we compare binary relevance ensembles with classifier chain ensembles built using the same base classifier. The results are shown in Figure 2 and Table 2. Full results are presented in the appendix in Tables 5, 6 and 7. The compared algorithms are numbered as follows:

1. BR ensemble built using J48 algorithm.
2. CC ensemble built using J48 algorithm.
3. BR ensemble built using SVM algorithm.
4. CC ensemble built using SVM algorithm.
5. BR ensemble built using NB algorithm.
6. CC ensemble built using NB algorithm.
7. BR ensemble built using KNN algorithm.
8. CC ensemble built using KNN algorithm.

The analysis of the results clearly shows that there is a noticeable difference between two groups of measures. That is, the differences between BR-based and CC-based algorithms are greater in terms of example-based criteria. On the other hand, the differences in mean ranks are lower for example-based measures.

For the example based measures, the average ranks achieved by CC-based algorithms are lower than for BR-based algorithms. However, only for algorithms based on J48 classifier, the differences are significant for example-based FDR, FNR and $F_1$ measures. A similar trend is observed for the zero-one loss. In this case,
only differences for the nearest neighbour classifier are insignificant. It means that CC-based classifiers obtain the higher number of 'perfect match' results.

On the other hand, for the label-based measures and Hamming loss, almost no significant differences are observed. However, the average ranks suggest that for this group of measures, the classification quality may deteriorate.

The results clearly show that although label-specific quality measures do not change in a significant way, the prediction of the entire label-vector improves. This is an expected result since the CC-based approach incorporates the inter-label relations. This is a well-known fact that has been reported by authors that have previously compared both approaches [26].

The results also show that there are almost no significant differences between J48, NB and KNN based algorithms. Contrary, SVM algorithm tends to outperform the remaining ones in terms of example-based criteria, hamming loss and zero-one loss. It means that although J48 algorithm takes the biggest advantage of employing the chain rule, NB and KNN based classifiers are comparable to J48-based ensembles.

4.2 Naive Bayes Classifier

The results of the experimental study are presented in Table 3 and Figure 3a. Tables 8, 6 and 10 show full results of the experiment. Table 11 provides results of the statistical evaluation of the experiments. Figure 3a visualises the average ranks and provides a view of the Nemenyi post-hoc procedure.

First, let's analyse differences between the proposed heuristic and the simple ECC ensemble. The proposed method is tailored to optimise the macro-averaged $F_1$ loss so we begin with investigating macro-averaged measures. It is easy to see that both methods are comparable in terms of recall but the proposed one is significantly better in terms of precision. It means that the proposed method
Table 2: Base classifier comparison. Result of the statistical evaluation. Rnk stands for average rank over all datasets, Frd is the p-value obtained using the Friedman test and Wp-i denotes the p-value associated with the Wilcoxon test that compares the i-th algorithm against the others.

| Alg. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| Hamming | | | | | | | | |
| Zero-One | | | | | | | | |
| EX FDR | | | | | | | | |
| EX FNR | | | | | | | | |
| Rank | | | | | | | | |
| Frd | | | | | | | | |
| Wp-1 | | | | | | | | |
| Wp-2 | | | | | | | | |
| Wp-3 | | | | | | | | |
| Wp-4 | | | | | | | | |
| Wp-5 | | | | | | | | |
| Wp-6 | | | | | | | | |
| Wp-7 | | | | | | | | |

| Alg. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| Macro FDR | | | | | | | | |
| Macro FNR | | | | | | | | |
| Macro F1 | | | | | | | | |
| Rank | | | | | | | | |
| Frd | | | | | | | | |
| Wp-1 | | | | | | | | |
| Wp-2 | | | | | | | | |
| Wp-3 | | | | | | | | |
| Wp-4 | | | | | | | | |
| Wp-5 | | | | | | | | |
| Wp-6 | | | | | | | | |
| Wp-7 | | | | | | | | |

| Alg. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|------|---|---|---|---|---|---|---|---|
| Micro FDR | | | | | | | | |
| Micro FNR | | | | | | | | |
| Micro F1 | | | | | | | | |
| Rank | | | | | | | | |
| Frd | | | | | | | | |
| Wp-1 | | | | | | | | |
| Wp-2 | | | | | | | | |
| Wp-3 | | | | | | | | |
| Wp-4 | | | | | | | | |
| Wp-5 | | | | | | | | |
| Wp-6 | | | | | | | | |
| Wp-7 | | | | | | | | |
makes significantly less false positive predictions. Consequently, under the macro-averaged $F_1$ loss the proposed method outperforms the ECC ensemble. The same pattern is also present in results related to micro-averaged measures. However, the difference for the micro-averaged $F_1$ measure is not significant. In contrast, under example based measures, except the Hamming loss, there are no significant differences between investigated methods.

The results show that the proposed heuristic provides an effective way of improving classification quality for classifier chains ensemble. Moving the best performing label-specific models at the beginning of the chain reduces the error that propagates along the chain. What is more, the experimental study also showed that the Naive Bayes classifier combined with proper data preprocessing may be effectively employed in classifier chain ensembles.

Now, let’s compare the proposed method to the other algorithm based on the dynamic chain approach. When we investigate the example-based criteria it is easy to see that the OOC algorithm outperforms the proposed one in terms of FDR and Hamming loss. Those results combined with results achieved in terms of macro and micro averaged measures shows that the OOC method seems to be too much conservative. That is, it tends to makes many false negative predictions in comparison to the other methods. The outstanding results for the Hamming loss are a consequence of the imbalanced nature of the multi-label data. That is, the presence of labels is relatively rare and the prediction that contains many false negatives may achieve inadequately high performance under the Hamming loss [25].

On the other hand, the average ranks clearly show that the method based on genetic algorithm achieves the best results in comparison to the other investigated methods. The main reason is that the GA-based approach optimises the entire ensemble structure, whereas the investigated dynamic chain methods, choose the best label order for single classifier chain. Then the locally chosen chains are combined into an ensemble. It gives us an important clue. That is when we consider an algorithm for dynamic chain order selection, we should think about a single chain and the global structure of the entire ensemble as well.
4.3 Nearest Neighbour Classifier

The results of the experimental study are presented in Table 4 and Figure 3b. Tables 11, 12 and 13 show full results of the experiment. Table 3 provides results of the statistical evaluation of the experiments. Figure 3b visualises the average ranks and provides a view of the Nemenyi post-hoc procedure.

The results show that for the group of example based measures and the zero-one loss, there are no significant differences in classification quality between all investigated algorithms.

For macro and micro averaged measures, the best performing algorithm is an ensemble optimised using the genetic algorithm. The proposed nearest-neighbour-based classifier does not differ significantly from ECC OCCC algorithms. However, it tends to be more conservative because it achieves lower FDR and higher FDR. In other words, the classifier tends to decrease the false positive rate at the cost of decreasing the true positive rate. This phenomenon causes the highest classification quality in terms of the Hamming loss. The reason is that for the multi-label set with low label density, it is easy to obtain high classification by setting all possible labels as irrelevant.

The results confirm the findings described in Section 4.1. The nearest-neighbour-based CC algorithm is unable to take all benefits of the chaining approach. On the other hand, the method is still comparable to chains built using different base classifiers. The main goal of this paper was to propose the model that can change the chain structure without retraining. This goal was achieved.

5 Conclusions

The main goal of this research was to provide an effective chain classifier that allows changing label order at relatively low computational cost. We achieved it using a classifier based on the Naive Bayes approach. To prove that the proposed
method allows handling inter-label relations in an efficient way, we proposed a simple heuristic method that determines label order that should minimise label propagation error. Indeed, the experimental results showed that the proposed method is able to produce a good chain structure at a low computational cost. However, the proposed method of building a dynamic ensemble does not allow to outperform the static system that optimizes the entire ensemble structure. The obtained results are very promising. We believe that there is still a room for improvement. In our opinion, the performance of the system may be improved if we provide better, a better heuristic that optimises the entire ensemble in a dynamic way. The proposed dynamic classifier is a first step in the process of investigating dynamic classifier chain ensembles.

Another way of improving this idea is to build different classifiers that would be able to change the chain structure without retraining the entire model.

Acknowledgment

The work was supported by the statutory funds of the Department of Systems and Computer Networks, Wroclaw University of Science and Technology, under agreement 0401/0159/16.

References

1. Alvares Cherman, E., Metz, J., Monard, M.C.: A simple approach to incorporate label dependency in multi-label classification. In: Advances in Soft Computing, pp. 33–43. Springer Berlin Heidelberg (2010). DOI 10.1007/978-3-642-16773-7_4
2. Chang, C.C., Lin, C.J.: LIBSVM. ACM Transactions on Intelligent Systems and Technology 2(3), 1–27 (2011). DOI 10.1145/1961189.1961199
3. Charte, F., Rivera, A., del Jesus, M.J., Herrera, F.: Concurrency among imbalanced labels and its influence on multilabel resampling algorithms. In: Lecture Notes in Computer Science, pp. 110–121. Springer International Publishing (2014). DOI 10.1007/978-3-319-07617-1_10
4. Charte, F., Rivera, A.J., del Jesus, M.J., Herrera, F.: Quinta: A question tagging assistant to improve the answering ratio in electronic forums. In: IEEE EUROCON 2015 - International Conference on Computer as a Tool (EUROCON). IEEE (2015). DOI 10.1109/eurocon.2015.7313677

5. Chen, B., Li, W., Zhang, Y., Hu, J.: Enhancing multi-label classification based on local label constraints and classifier chains. In: 2016 International Joint Conference on Neural Networks (IJCNN). IEEE (2016). DOI 10.1109/jcnn.2016.7727370

6. Cheng, W., Hülsemeyer, E., Dembczynski, K.J.: Bayes optimal multilabel classification via probabilistic classifier chains. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp. 279–286 (2010)

7. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995). DOI 10.1007/bf00994018

8. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13(1), 21–27 (1967). DOI 10.1109/tit.1967.1053964

9. Díez, J., Luaces, O., del Coz, J.J., Bahamonde, A.: Optimizing different loss functions in multilabel classifications. Progress in Artificial Intelligence 3(2), 107–118 (2014). DOI 10.1007/s13748-014-0060-7

10. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 7, 1–30 (2006)

11. Dhar, M.: On cardinality of fuzzy sets. International Journal of Intelligent Systems and Applications 5(6), 47–52 (2013). DOI 10.5815/ijsa.2013.06.06

12. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. The Annals of Mathematical Statistics 11(1), 86–92 (1940). DOI 10.1214/aoms/1177731944

13. García, V., Sánchez, J., Molleda, R.: On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowledge-Based Systems 25(1), 13–21 (2012). DOI 10.1016/j.knosys.2011.06.013

14. Gibaja, E., Ventura, S.: Multi-label learning: a review of the state of the art and ongoing research. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 4(6), 411–444 (2014). DOI 10.1002/widm.1139

15. Gonçalves, E.C., Plastino, A., Freitas, A.A.: Simpler is better. In: Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO ’15. ACM Press (2015). DOI 10.1145/2739480.2754650

16. Gonçalves, E.C., Plastino, A., Freitas, A.A.: A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence. IEEE (2013). DOI 10.1109/ictai.2013.76

17. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software. ACM SIGKDD Explorations Newsletter 11(1), 10 (2009). DOI 10.1145/1656274.1656278

18. Hall, M.A.: Correlation-based feature selection for machine learning. Ph.D. thesis, The University of Waikato (1999)

19. Hand, D.J., Yu, K.: Idiot’s bayes: Not so stupid after all? International Statistical Review / Revue Internationale de Statistique 69(3), 385 (2001). DOI 10.2307/1403452

20. Holm, S.: A Simple Sequentially Rejective Multiple Test Procedure. Scandinavian Journal of Statistics 6(2), 65–70 (1979). DOI 10.2307/4615733

21. Huang, J., Li, G., Wang, S., Zhang, W., Huang, Q.: Group sensitive classifier chains for multi-label classification. In: 2015 IEEE International Conference on Multimedia and Expo (ICME). IEEE (2015). DOI 10.1109/icme.2015.7177400

22. Jiang, J.Y., Tsai, S.C., Lee, S.J.: Fsknn: Multi-label text categorization based on fuzzy similarity and k nearest neighbors. Expert Systems with Applications 39(3), 2813–2821 (2012). DOI 10.1016/j.eswa.2011.08.141

23. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Proc. 10th European Conference on Machine Learning, pp. 137–142 (1998)

24. Liu, X., Shi, Z., Li, Z., Wang, X., Shi, Z.: Sorted label classifier chains for learning images with multi-label. In: Proceedings of the international conference on Multimedia - MM ’10. ACM Press (2010). DOI 10.1145/1873951.1874121

25. Luaces, O., Díez, J., Barranquero, J., del Coz, J.J., Bahamonde, A.: Binary relevance efficacy for multilabel classification. Progress in Artificial Intelligence 1(4), 303–313 (2012). DOI 10.1007/s13748-012-0030-x
26. Madjarov, G., Kocev, D., Gjorgjevikj, D., Džeroski, S.: An extensive experimental comparison of methods for multi-label learning. Pattern Recognition 45(9), 3084–3104 (2012). DOI 10.1016/j.patcog.2012.03.004
27. Montańes, E., Senge, R., Barrancaero, J., Ramón Quevedo, J., José del Coz, J., Hüllermeier, E.: Dependent binary relevance models for multi-label classification. Pattern Recognition 47(3), 1494–1508 (2014). DOI 10.1016/j.patcog.2013.09.029
28. Quinlan, J.R.: C4.5 : Programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
29. Read, J., Martino, L., Luengo, D.: Efficient monte carlo methods for multi-dimensional learning with classifier chains. Pattern Recognition 47(5), 1535–1546 (2014). DOI 10.1016/j.patcog.2013.10.006
30. Read, J., Peter, R.: Meka:http://meka.sourceforge.net/ (2017). URL http://meka.sourceforge.net/
31. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. In: Machine Learning and Knowledge Discovery in Databases, pp. 254–269. Springer Berlin Heidelberg (2009). DOI 10.1007/978-3-642-04174-7_17
32. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning 85(3), 333–359 (2011). DOI 10.1007/s10994-011-5256-5
33. Sanden, C., Zhang, J.Z.: Enhancing multi-label music genre classification through ensemble techniques. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR ’11. ACM Press (2011). DOI 10.1145/2009916.2010011
34. Senge, R., del Coz, J.J., Hüllermeier, E.: On the problem of error propagation in classifier chains for multi-label classification. In: Studies in Classification, Data Analysis, and Knowledge Organization, pp. 163–170. Springer International Publishing (2013). DOI 10.1007/978-3-319-01595-8_18
35. da Silva, P.N., Gonçalves, E.C., Plastino, A., Freitas, A.A.: Distinct chains for different instances: An effective strategy for multi-label classifier chains. In: Machine Learning and Knowledge Discovery in Databases, pp. 453–468. Springer Berlin Heidelberg (2014). DOI 10.1007/978-3-662-44851-9_29
36. Soonsiripanichkul, B., Murata, T.: Domination dependency analysis of sales marketing based on multi-label classification using label ordering and cycle chain classification. 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI) (2016). DOI 10.1109/iiai-aai.2016.61
37. Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., Vlahavas, I.: Multi-target regression via input space expansion: treating targets as inputs. Machine Learning 104(1), 55–98 (2016). DOI 10.1007/s10994-016-5546-z
38. Spyromitros-Xioufis, E., Tsoumakas, G., Groves, W., Vlahavas, I.: Multi-target regression via input space expansion: treating targets as inputs. Machine Learning 104(1), 55–98 (2016). DOI 10.1007/s10994-016-5546-z
39. Tomás, J.T., Spolaór, N., Cherman, E.A., Monard, M.C.: A framework to generate synthetic multi-label datasets. Electronic Notes in Theoretical Computer Science 302, 155–176 (2014). DOI 10.1016/j.entcs.2014.01.025
40. Trajdos, P., Kurzyński, M.: Naive bayes classifier for dynamic chaining approach in multi-label learning. International Journal of Education and Learning Systems 2, 133–142 (2017). URL http://www.iaras.org/iaras/filedownloads/ijels/2017/002-0019(2017).pdf
41. Trajdos, P., Kurzyński, M.: Permutation-based diversity measure for classifier-chain approach. In: Advances in Intelligent Systems and Computing, pp. 412–422. Springer International Publishing (2017). DOI 10.1007/978-3-319-50162-9_23
42. Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and efficient multilabel classification in domains with large number of labels, p. 30–44 (2008)
43. Wei, Y., Xia, W., Lin, M., Huang, J., Ni, B., Dong, J., Zhao, Y., Yan, S.: Hpc: A flexible cnn framework for multi-label image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(9), 1901–1907 (2016). DOI 10.1109/tpami.2015.249129
44. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin 1(6), 80 (1945). DOI 10.2307/3001968
45. Wu, J.S., Huang, S.J., Zhou, Z.H.: Genome-wide protein function prediction through multi-instance multi-label learning. IEEE/ACM Transactions on Computational Biology and Bioinformatics 11(5), 891–902 (2014). DOI 10.1109/tcbb.2014.2323058
46. Xu, J.: Fast multi-label core vector machine. Pattern Recognition \textbf{46}(3), 885–898 (2013). DOI 10.1016/j.patcog.2012.09.003
47. Zadeh, L.: Fuzzy sets. Information and Control \textbf{8}(3), 338–353 (1965). DOI 10.1016/s0019-9958(65)90241-x
48. Zhang, P., Yang, Y., Zhu, X.: Approaching multi-dimensional classification by using bayesian network chain classifiers. In: 2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics. IEEE (2014). DOI 10.1109/ihmsc.2014.129
49. Zhou, Z.H., Zhang, M.L., Huang, S.J., Li, Y.F.: Multi-instance multi-label learning. Artificial Intelligence \textbf{176}(1), 2291–2320 (2012). DOI 10.1016/j.artint.2011.10.002

A Full results
Table 5 Base classifier comparison – full results – example based criteria.

| No. | Macro FDR | Macro FNR | Macro $F_1$ | Macro $F_1$ |
|-----|-----------|-----------|-------------|-------------|
| 1   | 0.668     | 0.622     | 0.758       | 0.645       |
| 2   | 0.639     | 0.608     | 0.507       | 0.527       |
| 3   | 0.386     | 0.402     | 0.384       | 0.418       |
| 4   | 0.526     | 0.449     | 0.449       | 0.449       |
| 5   | 0.582     | 0.476     | 0.411       | 0.433       |
| 6   | 0.351     | 0.365     | 0.330       | 0.333       |
| 7   | 0.466     | 0.510     | 0.442       | 0.498       |
| 8   | 0.557     | 0.481     | 0.425       | 0.468       |
| 9   | 0.210     | 0.216     | 0.196       | 0.198       |
| 10  | 0.007     | 0.006     | 0.007       | 0.007       |
| 11  | 0.617     | 0.543     | 0.532       | 0.552       |
| 12  | 0.507     | 0.538     | 0.527       | 0.546       |
| 13  | 0.730     | 0.703     | 0.700       | 0.640       |
| 14  | 0.416     | 0.405     | 0.380       | 0.336       |
| 15  | 0.838     | 0.785     | 0.765       | 0.805       |
| 16  | 0.713     | 0.728     | 0.716       | 0.699       |
| 17  | 0.252     | 0.298     | 0.446       | 0.411       |
| 18  | 0.525     | 0.471     | 0.516       | 0.488       |
| 19  | 0.800     | 0.761     | 0.737       | 0.706       |
| 20  | 0.703     | 0.694     | 0.694       | 0.696       |
| 21  | 0.345     | 0.346     | 0.289       | 0.366       |
| 22  | 0.261     | 0.232     | 0.165       | 0.177       |
| 23  | 0.509     | 0.506     | 0.419       | 0.510       |
| 24  | 0.456     | 0.389     | 0.392       | 0.422       |
Table 6 Base classifier comparison – full results – macro-averaged criteria and Hamming.

| No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----|---|---|---|---|---|---|---|---|
| Macro FDR | Macro FNR | Macro F1 | Hamming |
| 1  | 599 | 613 | 563 | 648 | 583 | 635 | 564 | 649 |
| 2  | 797 | 879 | 804 | 909 | 835 | 901 | 807 | 856 |
| 3  | 717 | 739 | 708 | 738 | 713 | 713 | 729 | 697 | 739 |
| 4  | 606 | 917 | 282 | 941 | 502 | 866 | 567 | 870 |
| 5  | 724 | 981 | 520 | 975 | 584 | 980 | 682 | 856 |
| 6  | 355 | 345 | 310 | 327 | 391 | 394 | 329 | 341 |
| 7  | 763 | 770 | 751 | 767 | 759 | 773 | 759 | 766 |
| 8  | 320 | 342 | 324 | 357 | 338 | 308 | 326 | 347 |
| 9  | 618 | 625 | 547 | 618 | 546 | 563 | 650 | 657 |
| 10 | 250 | 245 | 255 | 239 | 242 | 245 | 246 | 249 |
| 11 | 847 | 873 | 831 | 859 | 844 | 864 | 829 | 821 |
| 12 | 820 | 843 | 859 | 844 | 806 | 883 | 819 | 834 |
| 13 | 837 | 832 | 800 | 789 | 795 | 802 | 817 | 862 |
| 14 | 291 | 304 | 268 | 502 | 488 | 502 | 439 | 449 |
| 15 | 911 | 903 | 906 | 907 | 909 | 900 | 908 | 914 |
| 16 | 653 | 662 | 668 | 609 | 651 | 650 | 658 | 657 |
| 17 | 382 | 380 | 383 | 380 | 383 | 388 | 383 | 384 |
| 18 | 407 | 417 | 342 | 357 | 477 | 504 | 398 | 381 |
| 19 | 513 | 505 | 513 | 505 | 508 | 506 | 510 | 509 |
| 20 | 849 | 848 | 814 | 805 | 835 | 819 | 816 | 849 |
| 21 | 809 | 841 | 857 | 895 | 853 | 904 | 830 | 887 |
| 22 | 825 | 952 | 826 | 950 | 828 | 934 | 882 | 894 |
| 23 | 185 | 182 | 172 | 173 | 346 | 332 | 249 | 207 |
| 24 | 151 | 131 | 106 | 091 | 202 | 211 | 141 | 119 |
| 25 | 680 | 721 | 733 | 762 | 697 | 574 | 744 | 750 |
| 26 | 516 | 505 | 522 | 400 | 535 | 517 | 523 | 502 |
| 27 | 598 | 597 | 586 | 608 | 574 | 605 | 571 | 608 |
| 28 | 449 | 466 | 433 | 455 | 438 | 454 | 423 | 450 |
| 29 | 532 | 533 | 491 | 531 | 534 | 526 | 533 | 522 |
| 30 | 625 | 668 | 572 | 602 | 588 | 649 | 608 | 606 |
### Table 7: Base classifier comparison – full results – micro-averaged criteria and zero one.

| No | Micro FDR | Micro FNR | Micro $F_1$ | Zero one |
|----|-----------|-----------|-------------|----------|
|    | 1         | 2         | 3          | 4        | 5        | 6        | 7        | 8        |
| 1  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 2  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 3  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 4  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 5  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 6  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 7  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 8  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 9  | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 10 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 11 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 12 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 13 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 14 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 15 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 16 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 17 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 18 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 19 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 20 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 21 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 22 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |
| 23 | 0.008     | 0.008     | 0.008      | 0.008    | 0.008    | 0.008    | 0.008    | 0.008    |

Dynamic classifier chains for multi-label learning 23
Table 8: Naive Bayes – Full results – micro-averaged criteria.

| No. | Micro FDR | Micro FNR | Micro F1 |
|-----|-----------|-----------|----------|
| 1   | 0.753   | 0.703   | 0.727   |
| 2   | 0.534   | 0.578   | 0.556   |
| 3   | 0.462   | 0.513   | 0.487   |
| 4   | 0.559   | 0.516   | 0.535   |
| 5   | 0.742   | 0.707   | 0.725   |
| 6   | 0.592   | 0.581   | 0.586   |
| 7   | 0.676   | 0.661   | 0.668   |
| 8   | 0.628   | 0.640   | 0.634   |
| 9   | 0.575   | 0.571   | 0.573   |
| 10  | 0.415   | 0.458   | 0.439   |
| 11  | 0.234   | 0.315   | 0.274   |
| 12  | 0.583   | 0.578   | 0.578   |
| 13  | 0.578   | 0.586   | 0.582   |
| 14  | 0.479   | 0.489   | 0.484   |
| 15  | 0.344   | 0.345   | 0.345   |
| 16  | 0.553   | 0.568   | 0.559   |
| 17  | 0.448   | 0.479   | 0.464   |
| 18  | 0.262   | 0.235   | 0.248   |
| 19  | 0.461   | 0.441   | 0.451   |

Table 9: Naive Bayes – Full results – Example based criteria, Hamming and zero-one.

| No. | Hamming | Zero-One | EX FDR | EX FNR | EX F1 |
|-----|---------|----------|--------|--------|-------|
| 1   | 0.764   | 0.780   | 0.771   | 0.773   | 0.772   |
| 2   | 0.712   | 0.711   | 0.711   | 0.711   | 0.711   |
| 3   | 0.653   | 0.653   | 0.653   | 0.653   | 0.653   |
| 4   | 0.592   | 0.598   | 0.595   | 0.595   | 0.595   |
| 5   | 0.534   | 0.528   | 0.531   | 0.531   | 0.531   |
| 6   | 0.479   | 0.479   | 0.479   | 0.479   | 0.479   |
| 7   | 0.423   | 0.423   | 0.423   | 0.423   | 0.423   |
| 8   | 0.367   | 0.367   | 0.367   | 0.367   | 0.367   |
| 9   | 0.323   | 0.323   | 0.323   | 0.323   | 0.323   |
| 10  | 0.289   | 0.289   | 0.289   | 0.289   | 0.289   |
| 11  | 0.256   | 0.256   | 0.256   | 0.256   | 0.256   |
| 12  | 0.222   | 0.222   | 0.222   | 0.222   | 0.222   |
| 13  | 0.190   | 0.190   | 0.190   | 0.190   | 0.190   |
| 14  | 0.164   | 0.164   | 0.164   | 0.164   | 0.164   |
| 15  | 0.141   | 0.141   | 0.141   | 0.141   | 0.141   |
| 16  | 0.128   | 0.128   | 0.128   | 0.128   | 0.128   |
| 17  | 0.116   | 0.116   | 0.116   | 0.116   | 0.116   |
| 18  | 0.105   | 0.105   | 0.105   | 0.105   | 0.105   |
| 19  | 0.097   | 0.097   | 0.097   | 0.097   | 0.097   |
| 20  | 0.086   | 0.086   | 0.086   | 0.086   | 0.086   |
| 21  | 0.078   | 0.078   | 0.078   | 0.078   | 0.078   |
| 22  | 0.070   | 0.070   | 0.070   | 0.070   | 0.070   |
| 23  | 0.063   | 0.063   | 0.063   | 0.063   | 0.063   |
| 24  | 0.055   | 0.055   | 0.055   | 0.055   | 0.055   |
| 25  | 0.047   | 0.047   | 0.047   | 0.047   | 0.047   |
| 26  | 0.040   | 0.040   | 0.040   | 0.040   | 0.040   |
| 27  | 0.034   | 0.034   | 0.034   | 0.034   | 0.034   |
| 28  | 0.029   | 0.029   | 0.029   | 0.029   | 0.029   |
| 29  | 0.025   | 0.025   | 0.025   | 0.025   | 0.025   |
| 30  | 0.021   | 0.021   | 0.021   | 0.021   | 0.021   |
### Table 10 Naive Bayes – Full results – Macro averaged criteria.

| No. | Macro FDR | Macro FNR | Macro F1 |
|-----|-----------|-----------|----------|
| 1   | 0.152     | 0.161     | 0.157    |
| 2   | 0.365     | 0.394     | 0.377    |
| 3   | 0.416     | 0.429     | 0.420    |
| 4   | 0.508     | 0.510     | 0.509    |
| 5   | 0.616     | 0.623     | 0.620    |
| 6   | 0.739     | 0.743     | 0.741    |
| 7   | 0.867     | 0.872     | 0.869    |
| 8   | 0.965     | 0.967     | 0.966    |
| 9   | 0.847     | 0.850     | 0.849    |
| 10  | 0.665     | 0.667     | 0.666    |
| 11  | 0.416     | 0.417     | 0.416    |
| 12  | 0.298     | 0.300     | 0.299    |
| 13  | 0.215     | 0.217     | 0.216    |
| 14  | 0.111     | 0.112     | 0.111    |
| 15  | 0.089     | 0.090     | 0.089    |
| 16  | 0.069     | 0.070     | 0.069    |
| 17  | 0.051     | 0.052     | 0.051    |
| 18  | 0.036     | 0.037     | 0.036    |
| 19  | 0.026     | 0.027     | 0.026    |
| 20  | 0.016     | 0.017     | 0.016    |

### Table 11 Nearest Neighbours – Full results – micro-averaged criteria.

| No. | Micro FDR | Micro FNR | Micro F1 |
|-----|-----------|-----------|----------|
| 1   | 0.267     | 0.268     | 0.268    |
| 2   | 0.486     | 0.487     | 0.486    |
| 3   | 0.605     | 0.606     | 0.606    |
| 4   | 0.723     | 0.724     | 0.724    |
| 5   | 0.841     | 0.842     | 0.842    |
| 6   | 0.959     | 0.960     | 0.960    |
| 7   | 0.813     | 0.814     | 0.813    |
| 8   | 0.685     | 0.686     | 0.686    |
| 9   | 0.557     | 0.558     | 0.558    |
| 10  | 0.429     | 0.430     | 0.429    |
| 11  | 0.301     | 0.302     | 0.301    |
| 12  | 0.173     | 0.174     | 0.173    |
| 13  | 0.049     | 0.050     | 0.049    |
| 14  | 0.026     | 0.027     | 0.026    |
| 15  | 0.015     | 0.016     | 0.015    |
| 16  | 0.009     | 0.010     | 0.009    |
| 17  | 0.006     | 0.007     | 0.006    |
| 18  | 0.004     | 0.005     | 0.004    |
| 19  | 0.002     | 0.003     | 0.002    |
| 20  | 0.001     | 0.001     | 0.001    |
| No. | Hamming | Zero-One | EX FDR | EX FNR | EX F₁ |
|-----|---------|----------|--------|--------|--------|
| 1   | 0.069   | 0.070    | 0.069  | 0.069  | 0.070  |
| 2   | 0.092   | 0.092    | 0.100  | 0.103  | 0.103  |
| 3   | 0.649   | 0.649    | 0.649  | 0.652  | 0.652  |
| 4   | 0.120   | 0.181    | 0.121  | 0.205  | 0.205  |
| 5   | 0.058   | 0.055    | 0.066  | 0.065  | 0.065  |
| 6   | 0.118   | 0.162    | 0.117  | 0.117  | 0.117  |
| 7   | 0.308   | 0.344    | 0.325  | 0.314  | 0.314  |
| 8   | 0.157   | 0.183    | 0.176  | 0.179  | 0.179  |
| 9   | 0.472   | 0.471    | 0.470  | 0.471  | 0.471  |
| 10  | 0.393   | 0.387    | 0.388  | 0.389  | 0.389  |
| 11  | 0.939   | 0.895    | 0.936  | 0.936  | 0.936  |
| 12  | 0.227   | 0.243    | 0.236  | 0.226  | 0.226  |
| 13  | 0.344   | 0.325    | 0.324  | 0.323  | 0.323  |
| 14  | 0.746   | 0.744    | 0.744  | 0.745  | 0.745  |
| 15  | 0.386   | 0.373    | 0.371  | 0.370  | 0.370  |
| 16  | 0.593   | 0.627    | 0.699  | 0.773  | 0.773  |
| 17  | 0.324   | 0.317    | 0.348  | 0.333  | 0.333  |
| 18  | 0.938   | 0.906    | 0.960  | 0.965  | 0.965  |
| 19  | 0.893   | 0.898    | 0.922  | 0.947  | 0.947  |
| 20  | 0.766   | 0.782    | 0.785  | 0.791  | 0.791  |
| 21  | 0.464   | 0.468    | 0.469  | 0.470  | 0.470  |
| 22  | 0.910   | 0.934    | 0.916  | 0.916  | 0.916  |
| 23  | 0.940   | 0.930    | 0.955  | 0.955  | 0.955  |
| 24  | 0.649   | 0.649    | 0.652  | 0.652  | 0.652  |
| 25  | 0.939   | 0.895    | 0.936  | 0.936  | 0.936  |
| 26  | 0.222   | 0.222    | 0.222  | 0.222  | 0.222  |
| 27  | 0.702   | 0.708    | 0.706  | 0.706  | 0.706  |
| 28  | 0.958   | 0.958    | 0.958  | 0.958  | 0.958  |
| 29  | 0.736   | 0.736    | 0.736  | 0.736  | 0.736  |
| 30  | 0.972   | 0.972    | 0.972  | 0.972  | 0.972  |
| 31  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 32  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 33  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 34  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 35  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 36  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 37  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 38  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 39  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 40  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 41  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 42  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 43  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 44  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 45  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 46  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 47  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 48  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 49  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |
| 50  | 0.973   | 0.973    | 0.973  | 0.973  | 0.973  |