Circular Chain Classifiers

Jesús Joel Rivas [a,b] JRIVAS@CCC.INAOEP.MX

Felipe Orihuela-Espina [a] F.ORIHUELA-ESPINA@CCC.INAOEP.MX

Luis Enrique Sucar [a] ESUCAR@INAOEP.MX

[a] *Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, México, 72840*

[b] *Universidad de Carabobo, Facultad de Ciencias y Tecnología (FACYT), Carabobo, Venezuela, 2005*

Abstract

Chain Classifiers (CC) are an alternative for multi-label classification that is efficient and provides, in general, good results. However, it is not clear how to define the order of the chain. Different orders tend to produce different outcomes. We propose an extension to chain classifiers called "Circular Chain Classifiers" (CCC), in which the propagation of the classes of the previous binary classifiers is done iteratively in a circular way. After the first cycle, the predictions from the base classifiers are entered as additional attributes to the first one in the chain. This process continues for all the classifiers in the chain, and it is repeated for a prefixed number of cycles or until convergence. Using two datasets, we empirically established that CCC: (i) converges in few iterations (in general, 3 or 4), (ii) the initial order of the chain does not have a significant impact on the results. CCC performance was also compared against binary relevance and chain classifiers producing statistically superior results. The main contribution of CCC is its independence from the preestablished order of the chain, outperforming CC.

Keywords: multi-label classification; chain classifiers; class variables ordering.

1. Introduction

Several classification problems require assigning more than one class simultaneously to the feature vector $\vec{x}_u$. For example: in affective computing, a photo can evoke a mixture of affective states instead of just one affective state. This type of classification where the objects can be tagged with several simultaneous classes is called multidimensional classification (Van Der Gaag and De Waal, 2006; Bielza et al., 2011; Sucar, 2015). Formally: Given the objects $u = (\vec{x}_u, \vec{c}_u)$, where the vector $\vec{x}_u = (x_1, x_2, \cdots, x_d)_u$ is the feature vector, with $x_i \in \Omega_{X_i} = \mathbb{R}$ (or $\mathbb{Z}$), $i \in \{1, 2, ..., d\}$, and $\vec{c}_u = (c_1, c_2, \cdots, c_q)_u$ is the vector of class values assigned to $\vec{x}_u$, with $c_j \in \Omega_{C_j} \subseteq \mathbb{Z}$ (or $\Omega_{C_j} = \{-1, 1\}$ or $\Omega_{C_j} = \{0, 1\}$) , $j \in \{1, 2, ..., q\}$. The goal consists in learning the function $h : \Omega_{X_1} \times \Omega_{X_2} \times \cdots \times \Omega_{X_d} \to \Omega_{C_1} \times \Omega_{C_2} \times \cdots \times \Omega_{C_q}$, such that $h(\vec{x}_u) = \vec{c}_u \; \forall u$, i. e. h assigns the most likely combination of classes values to $\vec{x}_u$ (that minimizes misclassification), as represented in 1.

$$h(\vec{x}_u) = \underset{(c_1, c_2, \cdots, c_q)_u}{\arg\max} \left(P(C_1 = c_1, C_2 = c_2, \cdots, C_q = c_q | \vec{x}_u)\right) \tag{1}$$

where P is the probability.

$$(x_1, x_2, \cdots, x_d)_u \mapsto (c_1, c_2, \cdots, c_q)_u$$

When the classes c_j, $j \in \{1, 2, ..., q\}$ are binary, the multidimensional classification problem is called multi-label classification. There are two main approaches to tackle multi-label classification (Sucar et al., 2014): binary relevance (BR) and label power-set. In binary relevance, the problem is transformed into q binary classification problems, one for each class variable, $C_1, C_2, \cdots, C_q$ (Zhang and Zhou, 2007). Each classifier independently creates a model for predicting its class and the results of all of them are aggregated to produce the predicted class vector $\vec{c}_u$. This approach has low computational complexity and common single valued classification techniques can be directly applied, but its main drawback is that does not exploit potential interactions between classes (Sucar et al., 2014). In contrast, the label power-set approach (Tsoumakas and Katakis, 2007) transforms the problem into a single class problem defining a compound class variable C that represents the combination of the individual class variables. The values of C are all the possible combinations of values of the individual class variables. In this case, the interactions between class variables are considered, but the computational complexity increases exponentially with the number of individual class variables.

Intermediate strategies have been proposed to overcome the limitations of the two previous approaches. One of these is chain classifiers (CC), which incorporates class interactions to the binary relevance approach while maintaining computational efficiency (Read et al., 2009)

If the class vector is $\vec{c} = (c_1, c_2, \cdots, c_q)$, then a chain of q base binary classifiers (one per class) is built and linked so that each classifier incorporates between its input features, the values of the predicted classes by the preceding classifiers in the chain. The ordering of the class variables in the chain affects the results (Gonçalves et al., 2013; Sucar et al., 2014) and usually an ensemble of random orderings is developed, which requires more computational resources (Sucar et al., 2014). Two different class variables orderings in the chain tend to produce different results because the classifiers receive different previous class inputs (Dembczynski et al., 2010).

In this work we propose an extension to chain classifiers called “Circular Chain Classifiers” (CCC), in which the propagation of the classes of the previous binary classifiers is done in a circular way. In CCC, after the first cycle, the predictions of all the classifiers are entered as additional attributes to the first one in the chain. This process continues to classifier in position 2 and, so on, and it is repeated, to all the classifiers, for N cycles or until convergence. We aim to alleviate the problem of class variables ordering. We present empirical evidence that the performance of CCC does not depend on the order of the chain and further improves the performance of CC.

This paper is organized as follows, section 2 summarizes related work. Sections 3 introduces “Circular Chain Classifiers” (CCC). The methodology is presented in section 4. Section 5 highlights the experiments and results obtained with CCC, including the convergence process, the performance comparison with BR and CC, and the effects of the class variables ordering in CCC performance. Section 6 contains the discussion and, finally section 7 summarizes the main findings and describes future work.

2. Related Work

Multi-label classification has been reviewed in Tsoumakas and Katakis (2007) and in Zhang and Zhou (2014). Two main algorithmic strategies are distinguished: (1) problem transformation methods, and (2) algorithm adaptation methods. According to the first algorithmic strategy, Read et al. (2009) introduced chain classifiers as an option for multi-label classification, incorporating class dependencies, meanwhile maintaining the computational efficiency of the binary relevance method

(Zhang et al., 2018). In their work, they proposed a combination of several chain classifiers by changing the order of the class variables, setting up an ensemble of chain classifiers. Thereby, k chain classifiers, characterized by the variation of the class variables ordering in the chain and by the training data (both were settled randomly), were trained, and the final class values vector was obtained using a voting scheme; each class variable value c_j, $j \in \{1, 2, ..., q\}$ received a number of votes from the k chain classifiers, and a threshold was used to determine the final predicted multi-label values. Afterwards, Dembczynski et al. (2010) proposed probabilistic chain classifiers (PCCs), setting up chain classifiers under a probabilistic framework. Specifically, the chain rule of probability theory was applied to the probability of the vector of the q class values given the feature vector. Their approach provided better estimations than the chain classifiers, but with a much higher computational cost. Another alternative, under a probabilistic framework, is using Bayesian chain classifiers (Zaragoza et al., 2011), where the chain rule of probability theory is applied and the expression is simplified considering the independence relations between the class variables. A directed acyclic graph (DAG) is built representing the dependency relations between the class variables and only the parents of each class variable are included in the chain. The class variables ordering in the chain are defined through the paths in the DAG with the class variables parents. Alternative approaches use a genetic algorithm trying to search the best class variables ordering (Gonçalves et al., 2013), in what was called a Genetic Algorithm for ordering Chain Classifiers (GACC). This method outperforms BR and CC, but it is necessary to generate several orderings and search in the corresponding search space. A variation of chain classifiers called ring-based classifier was developed by Escalante et al. (2013) on the problem of detecting sexual predators in chat conversations. Each document that represented a chat conversation was divided into three parts for evaluating three different stages that a predator uses when approaching a child. The classification strategy consisted in training a local (base) classifier (they employed a neural network) for each part of a document and then combining the outputs through the idea of chain classifiers. Their proposal of ring-based classifier was developed through an iterative process where all the permutations of the 3 parts of a document were generated and incorporated continuously in the chain classifiers. They reported results that outperformed the results of traditional chain classifiers.

3. Circular Chain Classifiers (CCC)

In this work, we propose an extension of CC where the class variables ordering does not matter because the system involves a cyclic process in which all the base classifiers in the chain receive the class information from all the other ones.

The circular chain classifier consists of q base binary classifiers linked circularly in a chain, generating a ring architecture (see Figure 1). As in chain classifiers, each binary classifier in positions $2, 3, \ldots, q$ incorporates the predicted classes of the previous classifiers as other attributes. The circular configuration is obtained after the first iteration or "cycle" when the predicted classes of the classifiers (in positions $2, \ldots, q$) are entered as additional attributes to the first one in the chain. The propagation of the classes continues to classifier in position 2 and, so on. This process is repeated, to all the classifiers, for N cycles or until convergence.

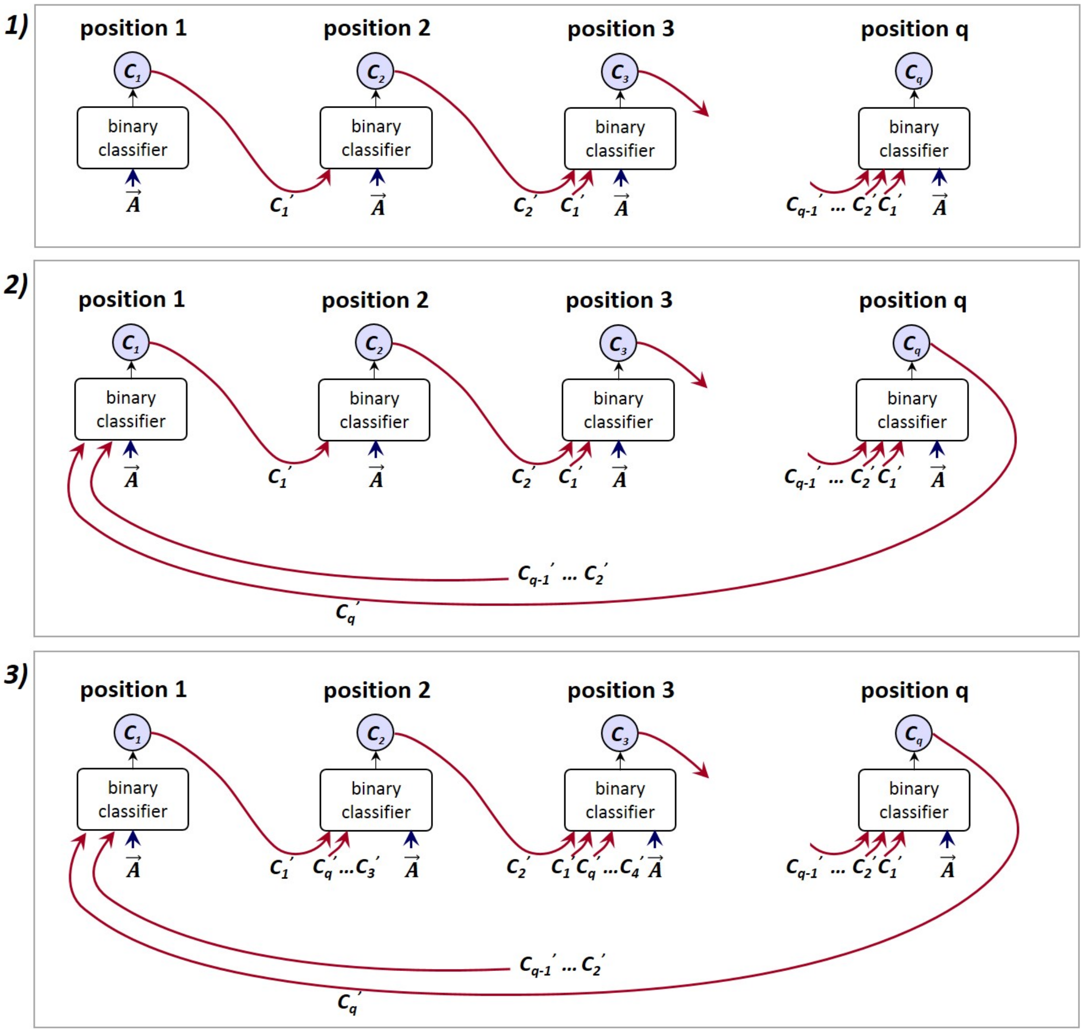


Figure 1: Schematic depiction of the propagation process at the CCC classifier. 1) In the first iteration, the predicted classes $C_j^{\prime}, j \in \{1, 2, \ldots, q-1\}$ are propagated as chain classifiers. $\overrightarrow{A}$ is the vector of attributes. 2) As from the second iteration, the classifier in position 1 receives the predicted classes from the last classifier (the one in position q) and the other classifiers (positions $2, 3, \ldots, q-1$). 3) the propagation process continues to the following binary classifiers in the chain.

4. Methodology

Experimental description

The performance of CCC was evaluated against BR (used as baseline) and CC. Naïve Bayes (NB) was used as the base classifier for all the chain, and we made additional experiments using Semi-Naïve Bayesian classifiers (SNB) (Pazzani, 1996; Martínez-Arroyo and Sucar, 2006) to evaluate changes in the performance attributable to the base classifier for all the chain. The performance comparison is described in Section 5.

Datasets

BR, CC and CCC were tested on 2 benchmark multi-label data sets: flags (Gonçalves et al., 2013) and emotions (Trohidis et al., 2008). Most of the features of these datasets are numeric and we handled them using the Proportional k-Interval Discretization (PKID) method (Yang and Webb, 2001) which has been suggested to be a suitable discretization alternative for Bayesian classifiers (Yang and Webb, 2001). All class variables of emotions and flags, are binary. The details of the datasets are summarized in Table 1.

No	Dataset	p	d	q	Domain	Reference
1	flags	194	19	7	images	(Gonçalves et al., 2013)
2	emotions	593	72	6	music	(Trohidis et al., 2008)

Table 1: Multi-label datasets used in the experiments. p is the number of examples of the dataset, d is the number of features, q is the number of binary class variables or labels.

Evaluation Metrics

To evaluate the performance of multi-label classifiers we used Global accuracy ($GAcc$), Mean accuracy ($MAcc$), Multi-label accuracy ($MLAcc$) and $F\text{-}measure$; metrics for multi-label classification (Bielza et al., 2011; Sorower, 2010; Godbole and Sarawagi, 2004). The following notation is adopted to describe the metrics:

p: number of examples in the dataset.
$\vec{c}_u$: vector of true classes for example u.
$\vec{c}'_u$: vector of predicted classes for the example u.
$c_{u,j}$: true value of the class variable j for the example u.
$c'_{u,j}$: value predicted of the class variable j for the example u.

Exact Match Ratio (EMR) or Global Accuracy ($GAcc$**)** represents the extension of accuracy that is used in the traditional classification of a single class. $GAcc$ is the accuracy by all the classes of the examples.

$$GAcc = \frac{1}{p}\sum_{u=1}^{p}\bigwedge_{j=1}^{q}(c'_{u,j} = c_{u,j}) \tag{2}$$

where the result of the operator $\bigwedge_{j=1}^{q}$ is 1 to indicate true in all the expressions depending on j and 0 to indicate false in any of the expressions depending on j.

Mean Accuracy ($MAcc$**)** represents the accuracy by class, in this case the results that are partially correct are taken into account.

$$MAcc = \frac{1}{q}\sum_{j=1}^{q} Acc_j = \frac{1}{q}\sum_{j=1}^{q}\frac{1}{p}\sum_{u=1}^{p}\delta(c'_{u,j}, c_{u,j}) \tag{3}$$

where Acc_j is the calculation of accuracy for the class j and $\delta(c'_{u,j}, c_{u,j}) = 1$ if $c'_{u,j} = c_{u,j}$ and 0 otherwise.

Multi-label Accuracy ($MLAcc$**), or Jaccard Measure** is the proportion of predicted correct labels to the total number of labels (predicted and true) for that example, averaged over all examples.

$$MLAcc = \frac{1}{p}\sum_{u=1}^{p}\frac{|\vec{c}_u' \wedge \vec{c}_u|}{|\vec{c}_u' \vee \vec{c}_u|} \tag{4}$$

where $|\vec{c}_u' \wedge \vec{c}_u| = \sum_{j=1}^{q}(c_{u,j}' \wedge c_{u,j})$ and $|\vec{c}_u' \vee \vec{c}_u| = \sum_{j=1}^{q}(c_{u,j}' \vee c_{u,j})$.

F-measure is the harmonic mean between precision and recall.

$$F - measure = \frac{1}{p}\sum_{u=1}^{p}\frac{2\,|\vec{c}_u' \wedge \vec{c}_u|}{|\vec{c}_u'| + |\vec{c}_u|} \tag{5}$$

Internal validity of the BR, CC and CCC models was established using the stratified 10 fold cross-validation across all the examples.

Three experiments were carried out with the following purposes:

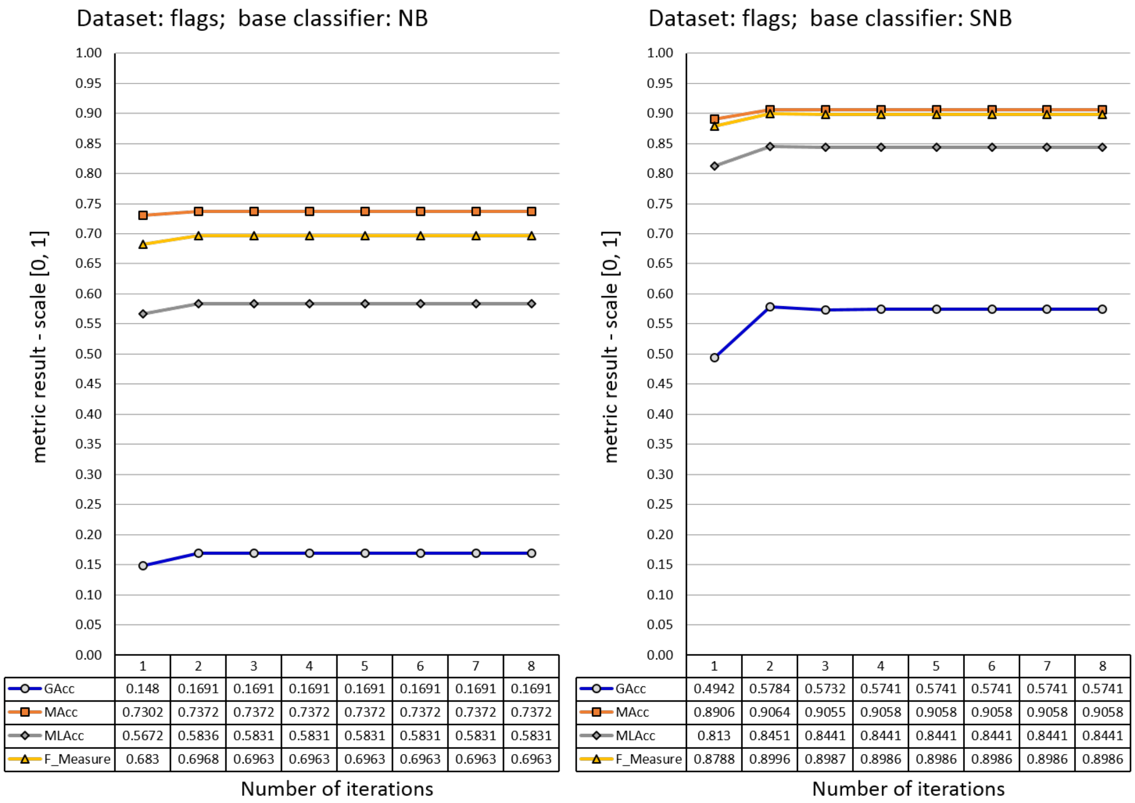
1. Determine experimentally the convergence of CCC on the aforementioned datasets, using as stopping criteria a fixed number of iterations.
2. Performance comparison of the three classifiers: BR, CC and CCC. First, using naïve Bayes (NB) as base classifier for all the classifiers, and then using the Semi-Naïve Bayesian classifier (SNB) as base classifier for all the classifiers.
3. Evaluation of the class variables ordering in the CCC. We want to determine whether the order is or not relevant for the results of CCC.

All the experiments were executed in both datasets.

5. Experiments and Results

Experiment 1: Determine experimentally the convergence of CCC:

Convergence was studied empirically setting a fixed number of iterations N and observing whether the metrics ($GAcc$, $MAcc$, $MLAcc$ and F-$measure$) outcomes tended towards a fixed value asymptotically. CCC was executed for N = 8, 20 and 30 iterations. Figures 2 and 3 show that at iteration 3 the system reaches a fixed point in almost all the cases. The CCC behaviour when NB was used as base classifier was stable for both datasets, it got a fixed value at iteration 3 and it maintained it over all the following iterations for the cases of 8, 20 and 30 iterations. When SNB was used as base classifier, CCC exhibited the same behaviour in the flags dataset. For the emotions dataset, convergence occurred at iteration 4 but got minimal fluctuations within the range of the value achieved (see Figure 3).



	1	2	3	4	5	6	7	8
GAcc	0.148	0.1691	0.1691	0.1691	0.1691	0.1691	0.1691	0.1691
MAcc	0.7302	0.7372	0.7372	0.7372	0.7372	0.7372	0.7372	0.7372
MLAcc	0.5672	0.5836	0.5831	0.5831	0.5831	0.5831	0.5831	0.5831
F_Measure	0.683	0.6968	0.6963	0.6963	0.6963	0.6963	0.6963	0.6963

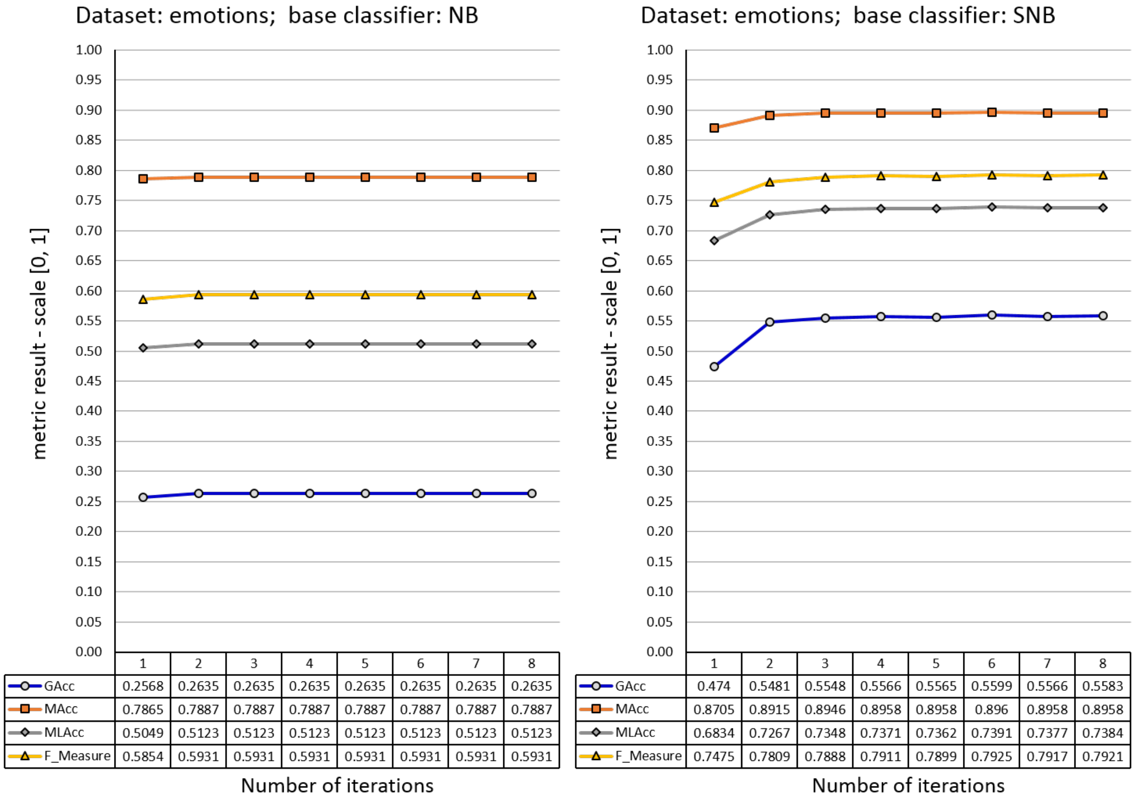
	1	2	3	4	5	6	7	8
GAcc	0.4942	0.5784	0.5732	0.5741	0.5741	0.5741	0.5741	0.5741
MAcc	0.8906	0.9064	0.9055	0.9058	0.9058	0.9058	0.9058	0.9058
MLAcc	0.813	0.8451	0.8441	0.8441	0.8441	0.8441	0.8441	0.8441
F_Measure	0.8788	0.8996	0.8987	0.8986	0.8986	0.8986	0.8986	0.8986

Figure 2: Convergence process of CCC over flags dataset when NB is the base classifier and then when SNB is the base classifier. In both cases, 8 iterations are showed and the system reached a fixed value at iteration 3 or 4.

Experiment 2: Performance comparison of BR, CC and CCC:

The class variables ordering for CC and for CCC was defined considering the BR results of ROC area under the curve (AUC) of each class variable. They were sorted in decreasing order according to AUC of BR results, interpreting that the class variables with worse results should be in the last positions so they could receive more information from the class variables of the preceding positions. Table 2 describes the ordering for each dataset and each base classifier.

Table 3 summarizes the classification results, $mean \pm std\ deviation$ (across the 10 folds of the cross-validation), of BR, CC and CCC, for both base binary classifiers: NB and SNB. The best mean results for each performance metric, each dataset and each base classifier are highlighted in bold type. CCC outperformed BR and CC for all the metrics whether using NB and SNB. Better results were obtained when SNB was the base classifier. Significant differences (Friedman test, $p < 0.05$, post hoc analysis with Wilcoxon signed-rank tests with Bonferroni correction, $p < 0.017$) were obtained for CCC (with SNB) when emotion dataset was used. CCC was run with 8 iterations.



	1	2	3	4	5	6	7	8
GAcc	0.2568	0.2635	0.2635	0.2635	0.2635	0.2635	0.2635	0.2635
MAcc	0.7865	0.7887	0.7887	0.7887	0.7887	0.7887	0.7887	0.7887
MLAcc	0.5049	0.5123	0.5123	0.5123	0.5123	0.5123	0.5123	0.5123
F_Measure	0.5854	0.5931	0.5931	0.5931	0.5931	0.5931	0.5931	0.5931

	1	2	3	4	5	6	7	8
GAcc	0.474	0.5481	0.5548	0.5566	0.5565	0.5599	0.5566	0.5583
MAcc	0.8705	0.8915	0.8946	0.8958	0.8958	0.896	0.8958	0.8958
MLAcc	0.6834	0.7267	0.7348	0.7371	0.7362	0.7391	0.7377	0.7384
F_Measure	0.7475	0.7809	0.7888	0.7911	0.7899	0.7925	0.7917	0.7921

Figure 3: Convergence process of CCC over emotions dataset using NB as base classifier and then SNB as base classifier. 8 iterations are showed and the system reached a fixed value at iteration 2 for NB. In the case of SNB, CCC generated values close to a fixed point from iteration 4.

Dataset	class variables	ordering for NB	ordering for SNB
flags	1-Red, 2-Green, 3-Blue, 4-Yellow, 5-White, 6-Black, 7-Orange	2-3-6-4-7-5-1	2-7-6-4-3-5-1
emotions	1-amazed, 2-happy, 3-calm, 4-quiet, 5-sad, 6-angry	3-6-1-4-5-2	4-6-3-1-5-2

Table 2: Class variables ordering for each dataset and each base classifer.

Experiment 3: Evaluation of the class variables ordering in the CCC:

Permutation tests (Ojala and Garriga, 2010), $p < 0.05$, were performance and there were not significant differences with the results of random permutations of the class variables in each dataset.

As an example, four permutations of class variables ordering in the CCC are presented: $Perm_k$, $k \in \{1, 2, 3, 4\}$.

Table 4 summarizes the results for the different permutations of the example. The results are similar for the different class variables ordering between each metric, each dataset and each base classifier.

Dataset	$GAcc$	$MAcc$	$MLAcc$	$F\text{-}measure$
flags				
		base classifier: naïve Bayes		
BR	0.1288 ± 0.0661	0.7259 ± 0.0437	0.5518 ± 0.0603	0.6687 ± 0.0637
CC	0.1480 ± 0.0552	0.7302 ± 0.0369	0.5672 ± 0.0471	0.6830 ± 0.0498
CCC	**0.1691 ± 0.0652**	**0.7372 ± 0.0421**	**0.5831 ± 0.0535** $*$	**0.6963 ± 0.0517** $*$
		base classifier: Semi-Naïve Bayes		
BR	0.4707 ± 0.1073	0.8898 ± 0.0290	0.8071 ± 0.0471	0.8750 ± 0.0326
CC	0.4942 ± 0.0828	0.8906 ± 0.0181	0.8131 ± 0.0276	0.8788 ± 0.0197
CCC	**0.5741 ± 0.0869** ‡	**0.9058 ± 0.0263**	**0.8441 ± 0.0422**	**0.8986 ± 0.0303**
emotions				
		base classifier: naïve Bayes		
BR	0.2483 ± 0.0605	0.7873 ± 0.0221	0.5022 ± 0.0448	0.5840 ± 0.0400
CC	0.2568 ± 0.0636	0.7865 ± 0.0229	0.5049 ± 0.0483	0.5854 ± 0.0427
CCC	**0.2635 ± 0.0704** $*$	**0.7887 ± 0.0234**	**0.5123 ± 0.0495** $*$	**0.5931 ± 0.0424** $*$
		base classifier: Semi-Naïve Bayes		
BR	0.3848 ± 0.0617	0.8435 ± 0.0188	0.6267 ± 0.0409	0.7016 ± 0.0388
CC	0.4740 ± 0.0611	0.8705 ± 0.0187	0.6834 ± 0.0338	0.7475 ± 0.0297
CCC	**0.5583 ± 0.0676** ‡	**0.8958 ± 0.0154** ‡	**0.7384 ± 0.0351** ‡	**0.7921 ± 0.0293** ‡

Table 3: Performance comparisons between BR, CC and CCC ($mean \pm std.deviation$). CCC was run with 8 iterations. The best results for each metric, each dataset and each base classifier are highlighted in bold type. "*" means significant difference between the three multi-label classifiers (Friedman test, $p < 0.05$) but without significant difference by pairs, "‡" means significant difference between CCC and CC, and between CCC and BR (Friedman test, $p < 0.05$, post hoc analysis with Wilcoxon signed-rank tests with Bonferroni correction, $p < 0.017$).

6. Discussion

Convergence of CCC occurred within few iterations: 3 when NB was the base classifier and 4 iterations when using SNB. According to these experiments it seems that CCC converges in few iterations, although a formal proof of convergence is left as future work. The results of CCC for the both datasets outperformed CC and BR, and in some cases, the results were significantly better than CC and BR. With respect to the base classifier, SNB outperformed NB, as it was expected; but the drawback of SNB is the combinatorial explosion when the number of attributes increases.

As it was explained in the introduction, different class variables ordering can potentially change the results when we are using CC; but we provided empirically evidence that CCC outcomes were robust to different class variables orderings, at least in the datasets used in this research. In CCC, the class variables ordering does not matter because the system involves a cyclic process in which all the base classifiers, in the chain, receive the class information from all the other ones, until the convergence is reached. Although there are other alternative approaches not so dependent on class order, such as ensemble methods (Read et al., 2009) or using a genetic algorithm (Gonçalves et al., 2013), the proposed CCC is simpler and computationally efficient.

Dataset	Permutation	$GAcc$	$MAcc$	$MLAcc$	$F\text{-}measure$
flags					
base classifier: naïve Bayes					
$Perm_1$	2-3-6-4-7-5-1	0.1691 ± 0.0652	0.7372 ± 0.0421	0.5831 ± 0.0535	0.6963 ± 0.0517
$Perm_2$	1-2-3-4-5-6-7	0.1691 ± 0.0652	0.7349 ± 0.0416	0.5811 ± 0.0515	0.6946 ± 0.0503
$Perm_3$	7-6-5-4-3-2-1	0.1691 ± 0.0652	0.7349 ± 0.0446	0.5799 ± 0.0579	0.6926 ± 0.0559
$Perm_4$	1-2-3-4-7-6-5	0.1691 ± 0.0652	0.7364 ± 0.0425	0.5832 ± 0.0532	0.6963 ± 0.0515
base classifier: Semi-Naïve Bayes					
$Perm_1$	2-7-6-4-3-5-1	0.5741 ± 0.0869	0.9058 ± 0.0263	0.8441 ± 0.0422	0.8986 ± 0.0303
$Perm_2$	1-2-3-4-5-6-7	0.5741 ± 0.0869	0.9073 ± 0.0235	0.8474 ± 0.0366	0.9005 ± 0.0265
$Perm_3$	7-6-5-4-3-2-1	0.5951 ± 0.0987	0.9103 ± 0.0242	0.8531 ± 0.0382	0.9037 ± 0.0270
$Perm_4$	1-2-3-4-7-6-5	0.5741 ± 0.0869	0.9073 ± 0.0235	0.8474 ± 0.0366	0.9005 ± 0.0265
emotions					
base classifier: naïve Bayes					
$Perm_1$	3-6-1-4-5-2	0.2635 ± 0.0704	0.7887 ± 0.0234	0.5123 ± 0.0495	0.5931 ± 0.0424
$Perm_2$	1-2-3-4-5-6	0.2685 ± 0.0750	0.7901 ± 0.0248	0.5157 ± 0.0538	0.5956 ± 0.0463
$Perm_3$	6-5-4-3-2-1	0.2635 ± 0.0704	0.7881 ± 0.0239	0.5117 ± 0.0497	0.5923 ± 0.0428
$Perm_4$	1-2-3-6-5-4	0.2652 ± 0.0717	0.7887 ± 0.0238	0.5129 ± 0.0519	0.5931 ± 0.0449
base classifier: Semi-Naïve Bayes					
$Perm_1$	4-6-3-1-5-2	0.5583 ± 0.0676	0.8958 ± 0.0154	0.7384 ± 0.0351	0.7921 ± 0.0293
$Perm_2$	1-2-3-4-5-6	0.5498 ± 0.0681	0.8913 ± 0.0166	0.7314 ± 0.0353	0.7864 ± 0.0277
$Perm_3$	6-5-4-3-2-1	0.5632 ± 0.0591	0.8977 ± 0.0140	0.7413 ± 0.0307	0.7948 ± 0.0269
$Perm_4$	1-2-3-6-5-4	0.5515 ± 0.0671	0.8941 ± 0.0159	0.7360 ± 0.0357	0.7912 ± 0.0286

Table 4: Performance comparison of CCC ($mean \pm std.deviation$) when the class variables are settled in different orderings. The results are similar within each metric, each dataset and each base classifier.

7. Conclusions and Future Work

The main contribution of this work is our proposal of extending chain classifiers to what we called "Circular Chain Classifiers" (CCC) which, according to the empirical evidence, do not depend on the class variables ordering of the chain, improving the performance of CC and maintaining the CC efficiency for the datasets used. The proposed extension is simple and easy to implement. The results are promising because CCC had fast convergence (just about 3 or 4 iterations), and outperformed CC and BR for the datasets used. Further experiments are necessary to have more conclusive evidence. In the future we will explore CCC performance with other datasets. A formal proof of convergence will be explored. Comparisons with ensemble of CC (Read et al., 2009) and with the genetic algorithms proposed by Gonçalves et al. (2013) will be done too.

Acknowledgments

We would like to acknowledge support for the Scholarship No. 434867 from the Mexican Research Council CONACYT.

References

C. Bielza, G. Li, and P. Larrañaga. Multi-dimensional classification with Bayesian networks. *International Journal of Approximate Reasoning*, 52(6):705–727, 2011.

K. Dembczynski, W. Cheng, and E. Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, volume 10, pages 279–286. IMLS, 2010.

H. J. Escalante, E. Villatoro-Tello, A. Juárez, M. Montes-y Gómez, and L. Villaseñor. Sexual predator detection in chats with chained classifiers. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 46–54, 2013.

S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

E. C. Gonçalves, A. Plastino, and A. A. Freitas. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In *Proceedings of the 25th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 469–476. IEEE, 2013.

M. Martínez-Arroyo and L. E. Sucar. Learning an optimal naive Bayes classifier. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 1236–1239. IAPR, 2006.

M. Ojala and G. C. Garriga. Permutation tests for studying classifier performance. *Journal of Machine Learning Research*, 11(Jun):1833–1863, 2010.

M. J. Pazzani. Searching for dependencies in Bayesian classifiers. In *Learning from Data*, pages 239–248. Springer, 1996.

J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2009.

M. S. Sorower. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 2010.

L. E. Sucar. *Probabilistic Graphical Models: Principles and Applications*. Springer, 2015.

L. E. Sucar, C. Bielza, E. F. Morales, P. Hernandez-Leal, J. H. Zaragoza, and P. Larrañaga. Multi-label classification with Bayesian network-based chain classifiers. *Pattern Recognition Letters*, 41:14–22, 2014.

K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas. Multi-label classification of music into emotions. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, volume 8, pages 325–330, 2008.

G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

L. C. Van Der Gaag and P. R. De Waal. Multi-dimensional Bayesian network classifiers. Technical Report UU-CS-2006-056, Department of Information and Computing Sciences, Utrecht University, 2006.

Y. Yang and G. I. Webb. Proportional k-interval discretization for naive-Bayes classifiers. In *Proceedings of the 12th European Conference on Machine Learning (ECML)*, pages 564–575. Springer, 2001.

J. H. Zaragoza, L. E. Sucar, E. F. Morales, C. Bielza, and P. Larrañaga. Bayesian chain classifiers for multidimensional classification. In *Proceedings of the 22th International Joint Conferences on Artificial Intelligence (IJCAI)*, volume 11, pages 2192–2197, 2011.

M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2014.

M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng. Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, 2(12):191–202, 2018.