Hybrid Binary–Chain Multi-label Classifiers

Pablo Hernandez-Leal, Felipe Orihuela-Espina, L. Enrique Sucar and Eduardo F. Morales
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1, Sta. Ma. Tonantzintla, Puebla, Mexico
{pablohl,f.orihuela-espina,esucar,emorales}@inaoep.mx

Abstract

In multi-label classification the goal is to assign an instance to a set of different classes. Several approaches have been proposed to deal with multi-label classification problems, ranging from considering each class independently from the other (binary relevance methods) to considering all the possible combinations of values of the original classes into a single compound class (power-set approach). In between, other methods have been proposed to consider dependencies among classes whilst trying to keep computational complexity of the method low. In this paper, instead of finding probabilistic dependencies among classes, we focused on finding independencies among classes using a simple correlation approach. We first build a correlation matrix among classes and use it to build chain classifiers among correlated sub-sets of classes while learning independent classifiers for uncorrelated classes. It is experimentally shown that this simple hybrid approach exhibits very competitive predictive performance among state-of-the-art multi-label classifiers with lower time complexity.

1 Introduction

Multi-label classifiers (MLCs) have gained an increasing attention in recent years, as different important problems can be seen as multi-label classification (Zhang and Zhou, 2007; Vens et al., 2008), such as text classification (assigning a document to several topics), HIV drug selection (determining the optimal set of drugs), among others.

There are basically two types of approaches that have been proposed for solving an MLC problem with binary classes: *binary relevance* and *label power-set* (Tsoumakas and Katakis, 2007). In the binary relevance approach (Zhang and Zhou, 2007), an MLC problem is transformed into d binary classification problems, one for each class variable, $C_1, \ldots, C_d$. A classifier is independently learned for each class variable, and the results are combined to determine the predicted class set. The problem with this approach is that it is unable to capture the interactions among classes and, in general, the most likely class of each classifier will not match the most likely set of classes due to possible interactions among them.

The label power-set approach (Tsoumakas and Katakis, 2007) on the other hand transforms the multi-label problem into a single-class version by defining a new class variable whose possible values are all of the possible combinations of values of the original classes. In this case the interactions between the different classes are implicitly considered. One disadvantage of this approach is its computational complexity, as the size of the new class variable increases exponentially with the number of classes.

Recently, alternative approaches have been proposed to consider dependencies among classes without incurring in high computational costs. One of these approaches is chain classifiers (Read et al., 2011), that consist of d binary classifiers which are linked in a chain, such that each classifier incorporates the class predicted by the previous classifiers as additional attributes. This approach combines the computational advantages of binary-relevance meth-

ods while incorporating dependencies among classes considered in the chain. The chain order, however, is randomly selected and does not necessarily capture the actual dependencies among classes while incorporating much irrelevant information from independent classes. A related approach is presented in (Zaragoza et al., 2011), where the authors use a probabilistic approach to establish a dependency class structure to build chain classifiers.

In this work, instead of looking for dependencies among classes, we look for independencies between classes using a simple correlation analysis. We first build a matrix of pairwise correlations between classes. Since uncorrelated classes can be considered independent, we build independent classifiers for them. For each subset of correlated classes, we group them and induce a simple chain classifier among them. We show, experimentally, that by identifying independent classes, we are able to significantly reduce the computation time while achieving competitive predictive performance results against state-of-the art multi-label classifiers.

2 Multi-Label Classifiers

The *multi-dimensional classification* problem corresponds to searching for a function h that assigns to each instance represented by a vector of m features $\mathbf{x} = (x_1, \ldots, x_m)$ a vector of d class values $\mathbf{c} = (c_1, \ldots, c_d)$:

$$h : \Omega_{X_1} \times \cdots \times \Omega_{X_m} \rightarrow \Omega_{C_1} \times \cdots \times \Omega_{C_d}$$

$$(x_1, \ldots, x_m) \mapsto (c_1, \ldots, c_d)$$

where there are d class variables, $C_1, \ldots, C_d$. We assume that $C_{i|i=1,\ldots,d}$ and $X_{j|j=1,\ldots,m}$ are discrete, and that Ω_{C_i} and Ω_{X_j} respectively represent their sample spaces.

Under a $0-1$ loss function, the h function should assign to each instance $\mathbf{x}$ the most likely combination of classes, that is:

$$h(x) = \underset{c_1,\ldots,c_d}{\operatorname{argmax}}\, p(C_1 = c_1, \ldots, C_d = c_d|\mathbf{x})$$

This assignment amounts to solving a total abduction inference problem and corresponds to the search for the most probable explanation (MPE), a problem that has been proved to be an NP-hard problem for Bayesian networks (Shimony, 1994).

3 Related Work

3.1 Multi-label Classification

In multi-label classification domains each instance is associated with a subset of labels from a set of d labels. This multi-label classification problem can be seen as a particular case of a multidimensional classification problem where all class variables are binary, that is $|\Omega_{C_i}| = 2$ for $i = 1, \ldots, d$.

An overview of multi-label classification is given in (Tsoumakas and Katakis, 2007). Two main approaches are distinguished: (a) problem transformation methods, which transform the multi-label classification problem into either one or more single-label classification problems and (b) algorithm adaptation methods, which extend specific learning algorithms to handle multi-label data directly.

Other related approaches are multidimensional Bayesian network classifiers (MBCs). A MBC is a Bayesian network $B = (\mathcal{G}, \Theta)$, where $\mathcal{G}$ is an acyclic directed graph with vertexes Z_i and Θ is a set of parameters $\theta_{z|\mathbf{pa}(z)} = p(z|\mathbf{pa}(z))$, where $\mathbf{pa}(z)$ is a value for the set $\mathbf{Pa}(Z)$, parents variables of Z in $\mathcal{G}$.

The set of vertexes $\mathcal{V}$ is partitioned into two sets $\mathcal{V}_\mathcal{C} = \{C_1, \ldots, C_d\}$, $d \geq 1$, of class variables and $\mathcal{V}_\mathcal{X} = \{X_1, \ldots, X_m\}$, $m \geq 1$, of feature variables. The set $\mathcal{A}$ of arcs is also partitioned into three sets, $\mathcal{A}_\mathcal{C}$, $\mathcal{A}_\mathcal{X}$, $\mathcal{A}_{\mathcal{C}\mathcal{X}}$, such that $\mathcal{A}_\mathcal{C} \subseteq \mathcal{V}_\mathcal{C} \times \mathcal{V}_\mathcal{C}$ is composed of the arcs between the class variables, $\mathcal{A}_\mathcal{X} \subseteq \mathcal{V}_\mathcal{X} \times \mathcal{V}_\mathcal{X}$ is composed of the arcs between the feature variables and finally, $\mathcal{A}_{\mathcal{C}\mathcal{X}} \subseteq \mathcal{V}_\mathcal{C} \times \mathcal{V}_\mathcal{X}$ is composed of the arcs from the class variables to the feature variables. The corresponding induced subgraphs are $\mathcal{G}_\mathcal{C} = (\mathcal{V}_\mathcal{C}, \mathcal{A}_\mathcal{C})$, $\mathcal{G}_\mathcal{X} = (\mathcal{V}_\mathcal{X}, \mathcal{A}_\mathcal{X})$ and $\mathcal{G}_{\mathcal{C}\mathcal{X}} = (\mathcal{V}, \mathcal{A}_{\mathcal{C}\mathcal{X}})$, called respectively class, feature and bridge subgraphs.

Different graphical structures for the class and feature subgraphs lead to different families of MBCs. For instance, a simple approach is to

learn trees for both subgraphs (van der Gaag and de Waal, 2006). Another work is (Qazi et al., 2007) in which the authors use a directed acyclic graph for the class subgraph, an empty graph for the features, and a bridge subgraph where features receive arcs from some class variables, without sharing any of them. (Bielza et al., 2011) present the most general models since any Bayesian network structure is allowed in the three subgraphs. Moreover they use all the possibilities for learning from data: wrapper, filter and hybrid score+search strategies.

Finally, another algorithm for multi-label classification is named RAKEL (Random K-Labelsets) (Tsoumakas and Katakis, 2007). RAKEL obtains m random subsets of size k and for each one a power set is constructed. A voting scheme with a user defined threshold determines the classification. One disadvantage of this approach is the random process used for obtaining the subsets, since no information is guiding the process.

3.2 Chain Classifiers

(Read et al., 2011) introduce chain classifiers as an alternative method for multi-label classification that incorporates class dependencies, while it tries to keep the computational efficiency of the binary relevance approach. Chain classifiers consist of d binary classifiers which are linked in a chain, such that each classifier incorporates the classes predicted by the previous classifiers as additional attributes. Thus, the feature vector for each binary classifier, L_i, is extended with the labels (0/1) of all previous classifiers in the chain. Each classifier in the chain is trained to learn the association of label l_i given the features augmented with all previous binary predictions in the chain, $l_1, l_2, \ldots, l_{i-1}$. For classification, it starts at L_1, and propagates along the chain such that for $i \in \mathcal{L}$ (where $\mathcal{L} = \{l_1, l_2, \ldots, l_d\}$) it predicts $p(l_i \mid \mathbf{x}, l_1, l_2, \ldots, l_{i-1})$. As in the binary relevance approach, the class vector is determined by combining the outputs of all the binary classifiers in the chain. They combine several chain classifiers by changing the order for the labels, building an ensemble of chain classifiers. The final label vector is obtained using a voting scheme; each label l_i receives a number of votes from the m chain classifiers, and a threshold is used to determine the final predicted multi-label set. They used support vector machines as the base binary classifier.

3.3 Bayesian Chain Classifiers

Given a multi-label classification problem with d classes, a Bayesian chain classifier (BCC)(Zaragoza et al., 2011) uses d classifiers, one per class, linked in a chain. The objective of this problem can be posed as finding a joint distribution of the classes $\mathbf{C} = (C_1, C_2, \ldots, C_d)$ given the attributes $\mathbf{x} = (x_1, x_2, \ldots, x_l)$: $p(\mathbf{C}|\mathbf{x}) = \prod_{i=1}^{d} p(C_i|\mathbf{pa}(C_i), \mathbf{x})$ where $\mathbf{pa}(C_i)$ represents the parents of class C_i. In this setting, a chain classifier can be constructed by inducing first the classifiers that do not depend on any other class and then proceed with their sons, according to the dependence structure which can be represented as a Bayesian network.

(Zaragoza et al., 2011) build a tree–structured dependency model. They simplify the problem by considering the marginal dependencies between classes, using (Chow and Liu, 1968) algorithm, that is, a *maximum weight undirected spanning tree* ($MWST$). Then they take a class (node) as root of a tree and assign directions to the arcs starting from this root node to build a directed tree. The chaining order of the classifiers is given by traversing the tree following an ancestral ordering. Based on this order they construct chain classifiers in which each base classifier incorporates one additional attribute, its parent class in the tree.

3.4 Other approaches

In (Kang et al., 2006) the authors describe a method to explicitly model correlations between class labels. The multi-label problem is posed as an optimization problem that considers how similar the training samples are with the testing samples considering subsets of classes at the same time. It needs to assign a weight to the classes given by their frequency. The method follows a greedy strategy that depends

on the relative order of the weights of the classes which are given in reverse order of the class frequency. In (Ji et al., 2010) the authors describe a method to extract shared structures (subspaces) in multi-label classification problems to capture the correlation information among classes. In this framework, a subspace is assumed to be shared among multiple labels, and a linear transformation is computed to discover this subspace. The shared structure is obtained by solving an eigenvalue problem using a regularization term to reduce the complexity.

In (Zhang and Zhang, 2010) the authors use a Bayesian network to encode the conditional dependencies of the labels as well as the feature set, with the feature set as the common parent of all labels. Their approach first constructs classifiers for all the labels independently. This produces some errors. They then learn a Bayesian network structure guided by these errors. Finally, they construct a new classifier for each label incorporating their parents in the Bayesian network as additional features.

4 Hybrid Binary–Chain Classifiers

Constructing chain classifiers with a random class order, as in (Read et al., 2011), can introduce irrelevant information and unnecessary computation, as many of the classes used as additional attributes in the chain may be independent of the current class. On the other hand, trying to find the probabilistic dependencies among classes given a set of attributes, can be computationally expensive. In this paper we find the independent groups of classes, using a simple correlation analysis, and use that information to significantly reduce the computation time while achieving competitive predictive performance results.

4.1 Correlation and Class Dependency

Correlation and statistical independence are two basic concepts denoting a relation among events. While both concepts express a notion of the link among processes they are fundamentally different (Mari and Kotz, 2001). It is a common mistake to deduce statistical independency from a lack of correlation and even worse, to infer statistical dependency from a strong correlation. The latter is not true and the former is valid as long as the relationship is linear, although zero correlation can be obtained when non-linear relations exist. In this paper, we assume a linear relationship among classes to identify independencies. Of course, other measures can be used instead such as mutual information for instance.

4.2 HBCC

To build a Hybrid Binary–Chain Classifier (HBCC) we start by obtaining the pairwise linear correlation coefficients between each pair of classes. Uncorrelated classes are considered independent and consequently, independent classifiers can be built from them. Correlated classes are grouped together and for each group a chain classifier is built. Groups with common members are merged into a single group. In the worst case, all the classes are grouped together and therefore this would degenerate in a chain using all the classes as proposed in (Read et al., 2011). By grouping only the correlated classes, we substantially reduce the computation time and eliminate irrelevant information from independent classes that is normally included in chain classifiers. Also, by focusing on identifying independent classes, we can apply a simpler approach based on correlation, rather than a more sophisticated approach such as structure learning of Bayesian networks. Algorithm 1 provides a description of the HBCC algorithm. The algorithm requires a parameter, λ, which defines the threshold used to consider two variables (classes) independent. If this threshold is too high the algorithm reduces to the binary relevance approach, if it is too low the result is a chain classifier (Read et al., 2011).

5 Experiments and Results

The experiments were performed on 9 different benchmark multi-label datasets[1]; each of them with different dimensions ranging from 6 to 983 labels, and from about 600 examples to more

[1]The data sets can be found at `mulan.sourceforge.net/datasets.html` and at `www.cs.waikato.ac.nz~/jmr30/\#datasets`.

Algorithm 1 Learning a Hybrid Binary-Chain Classifier.

Input: a multi-label dataset (BD), a threshold value (λ)
Obtain a correlation matrix (CM) among all classes in BD
for each $e \in CM$ **do**
  if $e < \lambda$ **then**
    $e \leftarrow 0$
  else
    $e \leftarrow 1$
  end if
end for
Obtain G groups of correlated classes
for each $g \in G$ **do**
  if $size(g) = 1$ **then**
    Create a Binary classifier for class g
  else
    Create a Chain Classifier using the classes in g.
  end if
end for

than 40,000. All class variables of the datasets are binary, however, in some of the datasets the feature variables are numeric. In these cases we used a static, global, supervised and top-down discretization algorithm (Cheng-Jung et al., 2008). The details of the datasets are summarized in Table 1.

For the purpose of comparison we used four different multi-label precision measures (Bielza et al., 2011; Read et al., 2011):

1. *Mean accuracy* over the d class variables (accuracy per label):

$$\text{M-Acc} = \frac{1}{d}\sum_{j=1}^{d} Acc_j = \frac{1}{d}\sum_{j=1}^{d}\frac{1}{N}\sum_{i=1}^{N}\delta(c'_{ij}, c_{ij}) \tag{1}$$

   where $\delta(c'_{ij}, c_{ij}) = 1$ if $c'_{ij} = c_{ij}$ and 0 otherwise. Note that c'_{ij} denotes the C_j class value outputted by the model for case i and c_{ij} is its true value.

2. *Global accuracy* over the d-dimensional class variable (accuracy per example):

$$\text{G-Acc} = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{c}'_i, \mathbf{c}_i) \tag{2}$$

   where $\delta(\mathbf{c}'_i, \mathbf{c}_i) = 1$ if $\mathbf{c}'_i = \mathbf{c}_i$ and 0 otherwise. Therefore, we call for a total coincidence on all of the components of the vector of predicted classes and the vector of real classes.

3. *Multilabel accuracy* as defined in (Tsoumakas and Katakis, 2007):

$$\text{ML-Acc} = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{c}_i \wedge \mathbf{c}'_i}{\mathbf{c}_i \vee \mathbf{c}'_i} \tag{3}$$

   In this measure, accuracy is micro-averaged across all examples.

4. *F measure* is the harmonic mean between precision and recall.

$$\text{F-measure} = \frac{1}{d}\sum_{i=1}^{d}\frac{2 \times p_j \times r_j}{(p_j + r_j)} \tag{4}$$

   where p_j and r_j are the precision and recall for all C_j. Here, the accuracy is calculated per label and then averaged.

Table 1: Multi-Label datasets used in the experiments. N is the size of the dataset, d is the number of binary classes or labels, m is the number of features. $*$ indicates numeric attributes.

No.	*Dataset*	*N*	*d*	*m*	*Type*
1	Emotions	593	6	72$*$	Music
2	Scene	2407	6	294$*$	Vision
3	Yeast	2417	14	103$*$	Biology
4	Medical	978	45	1449	Text
5	Enron	1702	53	1001	Text
6	TMC2007	28596	22	500	Text
7	Bibtex	7395	159	1836	Text
8	MediaMill	43907	101	120$*$	Media
9	Delicious	16105	983	500	Text

We estimated the performance measures using 10-fold cross-validation[2]. We used the Naive

[2]In a Core 2 Duo at 2.4 GHz with 8 GB of RAM.

Bayes, J48 and SVM implementations of Weka (Witten et al., 2011) software.

We compared HBCC against *binary relevance* (all the classes are considered to be independent of each other) and chain classifiers (as proposed in (Read et al., 2011)). We applied the Kruskal-Wallis test for testing statistical significance, which is an extension of the Wilcoxon rank sum test for more than two groups ($\alpha = 5\%$).

We first performed experiments with different values for λ (threshold value for building the correlation matrix) ranging from $\lambda = 0.1$ with almost all classes grouped together ($\approx$ Chain-NB) to $\lambda = 0.9$ with almost all classes independent ($\approx$ BR). We only show the average results for all the datasets and for each performance metric (see Table 2).

As can be seen from the experiments, considering some form of correlation improves the performance over the binary relevance method and is very competitive with chain classifiers.

For the next experiments we set $\lambda = 0.6$, which is the value that obtained the best results on average, for all the datasets.

Table 3 summarizes the accuracy results for HBCC, Chain-NB classifiers (Read et al., 2011), and binary relevance (BR) methods. Since dataset Delicious did not produce an output using Chain-NB after 48 hrs, the results of this dataset are not used for obtaining the average of the three approaches. From the results we can observe that HBCC obtained the best score in Mean and Global measures. Also there are some results in which HBCC obtained statistical significant difference with respect to the other two approaches.

We also quantified how much saving, in terms of attributes and in terms of processing time, are obtained with HBCC in comparison with a chain classifier.

Table 4 shows the number of extra attributes added in all the classifiers constructed for each dataset and the average size of the groups obtained with HBCC. From this table we can see that our proposed method greatly reduces the number of added attributes when compared to Chain-NB approach.

Finally, Table 5 indicates the processing time of a HBCC, including the time to evaluate the correlation matrix using Matlab; which is compared to the time obtained by the Chain-NB and binary relevance approaches. The savings in time obtained by the proposed algorithm are notable and becomes more pronounced with larger datasets. In particular, for the largest dataset, our approach is at least 5 times faster in comparison to Read's approach.

Table 4: Number of attributes added when classifying a dataset using Correlated-Based chain classifier (HBCC) and Chain-NB approach. Also the average size of the chains obtained by HBCC are presented.

Data set	*Attributes added*		Average chain size
	HBCC	Chain-NB	HBCC
Emotions	1	15	1.16
Scene	0	15	1
Yeast	12	91	1.85
Medical	0	990	1
Enron	12	1378	1.22
TMC2007	1	231	1.04
Bibtex	26	12561	1.16
MediaMill	59	5050	2.71
Delicious	4568	482653	5.64
Average	519.88	55887.11	1.86

Table 6 shows the performance of the proposed approach with two other base classifiers, C4.5 and SVM (implementations taken from Weka). We only show the average results for all the datasets and for each of the performance metrics. With C4.5 classifier our approach obtained the best results for all the measures. With SVM classifier HBCC obtained the best results in two measures and competitive results in the rest.

6 Conclusions and Future Work

In this paper we have introduced a Hybrid Binary-Chain Classifier for multi-label classification. The proposed approach is simple and easy to implement, and yet is highly competitive in terms of classification performance against a chain classifiers, and more efficient. We consider that this approach provides a practical and powerful alternative for building multidimensional classifiers.

Table 2: Average results with different threshold values for all the datasets and each performance metric.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	BR	Chain-NB
Mean	0.859	0.859	0.860	0.859	0.860	**0.860**	0.860	0.859	0.858	0.858	0.859
Global	0.210	**0.210**	0.209	0.207	0.208	0.208	0.208	0.207	0.207	0.207	0.200
ML	0.411	0.412	0.412	0.410	0.411	**0.412**	0.411	0.411	0.410	0.410	**0.413**
F-measure	0.388	0.388	0.389	0.389	**0.390**	0.389	0.390	0.388	0.387	0.387	**0.392**

Table 3: Classification accuracies for HBCC, (Read et al., 2011) approach (using Naive Bayes as base classifier) and binary relevance (BR) method. A "†" indicates statistically significant differences between HBCC and Chain-NB and a "*" indicates statistically significant differences between HBCC and BR.

Data set	*Mean Accuracy*			*Global Accuracy*		
	HBCC	BR	Chain-NB	HBCC	BR	Chain-NB
Emotions	0.8454	**0.8460**	0.8449	0.3879	**0.3895**	0.3879
Scene	**0.9058**	**0.9058**	0.9055	**0.5343**	**0.5343**	0.5301
Yeast	**0.8727***	0.8641	0.8673	**0.2768†**	0.2702	0.2586
Medical	**0.9746**	**0.9746**	0.9739	**0.2648**	**0.2648**	0.2587
Enron	**0.7811**	**0.7811**	0.7762	**0.0012†**	0.0006	0.0000
TMC2007	**0.8888**	**0.8888**	0.8803	**0.1435†**	**0.1435**	0.1136
Bibtex	**0.9130**	0.9126	0.9107	**0.0594**	0.0592	0.0549
MediaMill	0.7003	0.6963	**0.7168†**	**0.0003**	**0.0003**	0.0001
Delicious	**0.8937**	0.8871	-	0.0000	0.0000	-
Average	**0.8602**	0.8587	0.8594	**0.2085**	0.2078	0.2005
Data set	*Multi-Label Accuracy*			*F-measure*		
	HBCC	BR	Chain-NB	HBCC	BR	Chain-NB
Emotions	0.6689	**0.6695**	0.6679	0.7655	**0.7663**	0.7646
Scene	**0.7161**	**0.7161**	0.7157	0.7845	0.7845	**0.7848**
Yeast	**0.6733***	0.6571	0.6699	0.5935	0.5730	**0.6307†**
Medical	**0.3663**	**0.3663**	0.3596	**0.0865**	0.0865	0.0840
Enron	**0.1937**	0.1935	0.1869	**0.1460**	0.1459	**0.1472**
TMC2007	**0.4829**	**0.4829**	0.4333	**0.4717**	0.4717	0.4522
Bibtex	**0.1879†**	0.1872	0.1755	**0.1851**	0.1840	0.1817
MediaMill	0.0114	0.0110	**0.0982†**	0.0868	0.0847	**0.0941**
Delicious	**0.1352**	0.1249	-	**0.0702**	0.0620	-
Average	0.4126	0.4105	**0.4134**	0.3899	0.3871	**0.3924**

Table 5: Time (seconds) required to build a multi-label classifier using HBCC, Chain-NB and BR approaches. The first column shows the time to build a correlation matrix using Matlab, the third column shows the total time used for HBCC, the fourth shows the total time used for the Chain-NB approach and the fifth shows the total time for the BR approach.

Data set		*HBCC*		Chain-NB	BR
	Matrix	HBCC	Total	Total	Total
Emotions	0.01	0.27	0.28	0.44	0.27
Scene	0.01	1.34	1.35	2.46	1.30
Yeast	0.04	1.57	1.61	5.55	1.29
Medical	0.38	23.10	23.48	91.30	22.28
Enron	0.56	41.08	41.63	245.93	41.14
TMC2007	0.28	122.00	122.28	228.65	120.24
Bibtex	7.16	1064.58	1071.74	4770.54	1002.00
MediaMill	10.09	714.61	724.71	12120.53	615.646
Delicious	364.30	30638.98	31003.28	>172800.00	25200.02
Total	382.83	32607.52	32990.36	>190265.40	27004.19

Table 6: Performance with C4.5 and SVM

	C4.5			*SVM*		
	HBCC-0.6	BR	Chain	HBCC-0.6	BR	Chain
Mean Acc.	**0.9108**	0.9023	0.9087	0.9280	0.9230	**0.9283**
Global Acc.	**0.2239**	0.2178	0.2225	0.3509	0.3451	**0.3556**
ML Acc.	**0.4072**	0.3900	0.4031	**0.5760**	0.5658	0.5718
F-measure	**0.3618**	0.3246	0.3405	**0.5173**	0.5041	0.5125

As future work we plan to incorporate Bayesian chain classifiers for each group. Also we plan to use other clustering alternatives like spectral clustering or agglomerative clustering, as well as other measures such as Hilbert Schmidt independence criterion.

References

C. Bielza, G. Li, and P. Larrañaga. 2011. Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*.

T. Cheng-Jung, L. Chien-I, and Y. Wei-Pang. 2008. A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences*, (178):714–731.

C. Chow and C. Liu. 1968. Approximating discrete probability distributions with dependence trees. *Information Theory, IEEE Transactions on*, 14(3):462–467.

S. Ji, L. Tang, S. Yu, and J. Ye. 2010. A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(2):8.

F. Kang, R. Jin, and R. Sukthankar. 2006. Correlated label propagation with application to multi-label learning. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1719–1726. IEEE.

D.D. Mari and S. Kotz. 2001. *Correlation and dependence*. Imperial College Press.

M. Qazi, G. Fung, S. Krishnan, R. Rosales, H. Steck, R. Bharat Rao, D. Poldermans, and D. Chandrasekaran. 2007. Automated heart wall motion abnormality detection from ultrasound images using bayesian networks. *International Joint Conference on Artificial Intelligence*, pages 519–525.

J. Read, B. Pfahringer, G. Holmes, and E. Frank. 2011. Classifier chains for multi-label classification. *Machine Learning*, 85:333–359. 10.1007/s10994-011-5256-5.

S. E. Shimony. 1994. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399–410.

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

Linda C. van der Gaag and Peter R. de Waal. 2006. Multi-dimensional bayesian network classifiers. In *Third European Conference on Probabilistic Graphic Models*, pages 107–114.

C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel. 2008. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185–214.

I.H. Witten, E. Frank, and M.A. Hall. 2011. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

J.H. Zaragoza, L.E. Sucar, EF Morales, P. Larranaga, and C. Bielza. 2011. Bayesian chain classifiers for multidimensional classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

M.L. Zhang and K. Zhang. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 999–1008. ACM.

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