Journal of Universal Computer Science, vol. 19, no. 4 (2013), 502-520
submitted: 3/8/12, accepted: 20/2/13, appeared: 28/2/13

Boosting-based Multi-label Classification

Tomasz Kajdanowicz
(Wroclaw University of Technology, Poland
tomasz.kajdanowicz@pwr.wroc.pl)

Przemyslaw Kazienko
(Wroclaw University of Technology, Poland
przemyslaw.kazienko@pwr.wroc.pl)

Abstract: Multi-label classification is a machine learning task that assumes that a data instance may be assigned with multiple number of class labels at the same time. Modelling of this problem has become an important research topic recently. This paper revokes AdaBoostSeq multi-label classification algorithm and examines it in order to check its robustness properties. It can be stated that AdaBoostSeq is able to result with quite stable Hamming Loss evaluation measure regardless of the size of input and output space.
Key Words: multi-label classification, boosting, AdaBoostSeq, machine learning
Category: I.2, I.2.6, I.2.8

1 Introduction

In traditional classification the main objective is to learn a function f that maps an input instances $x \in \mathcal{X}$ to an output space composed of binary variable. The goal of mapping is typically focused on binary classification $\mathcal{C} = \{c_1, c_2\}$, often coded with {-1,1} numbers, or multi-class classification $\mathcal{C} = \{c_1, \ldots, c_m\}$. A classical example of binary classification is a problem of predicting whether the next day will or will not be rainy on the basis of historical weather data.

More sophisticated techniques allow solving classification problems with more complex outputs than only single label. A particular complex classification problem is multi-label classification. By multi-label problem it is understood the situation when particular data instance may be assigned with multiple number of class labels at the same time and these labels may be correlated. Two approaches that can deal with such problem can be distinguished, algorithm adaptation methods and problem transformation methods. Whereas the former adapt existing algorithms providing binary and multi-class classification, the latter transform the multi-label problem to multiple binary problems and then solve them using traditional approaches.

The growing interest in learning algorithms operating over multiple labels has arisen especially in last few years. Basic examples of multi-label problems abound in learning to automated tagging, medical diagnosis, document categorization, scene categorization and much more.

This paper addresses the problem of multi-label sequence classification, that traditionally has been identified with sequence labelling. This approach corresponds to the classification of a label sequence associated to observed input. By means of sequence labelling the problem of multi-label classification will be presented and considered in the following Sections. In particular, the paper provides related work specific to the topic (Section 2) and a nature of the sequential labelling problem. After that, details and formalism of the boosting algorithm for multi-label classification based on sequence-loss balancing function is revoked according to proposal in [Kajdanowicz and Kazienko, 2011] (Section 3). In Section 3.2 computational complexity of the method is assessed. The results of experiments on the real data from multi-label classification task are presented in Section 4.

2 Related Work

In general sequence labelling aims at solving the problem of learning a classification function $f : \mathcal{X} \rightarrow \mathcal{Y}$ with a potentially high output space $\mathcal{Y}$. For instance considering sequence labelling particular $y \in \mathcal{Y}$ is the sequence of labels associated to the data instance $x \in \mathcal{X}$.

According to the proposal in [Daume et al., 2009], it is assumed that a sequence labelling problem considered in this paper is a cost-sensitive classification problem (eg. [Lee et al., 2011]), where classification results y have the structure of a vector; y may be a sequence of l values (a l-length sequence): y_i=(y_i^1, y_i^2, ..., y_i^l), $\forall$ (μ=1,2,...,l) $y_i^\mu \in \{-1, 1\}$. The meaning of particular y_i^μ value ($\{-1, 1\}$) is simple and denotes whether the μth label (λ_μ) is absent or present in the label-set of ith data instance, respectively.

Additionally, the algorithms accomplishing sequence labelling can also make use of chaining idea and extend input space in generalization of consecutive labels from sequence. It means that while computing a given item value y_i^μ, $1 < \mu \leq l$, from the l-length sequence y_i=(y_i^1, y_i^2, ..., y_i^l), the algorithms may utilize the input data both from the original input $x_i \in X$ and from the partially produced output $y_i^{P_\mu}$, where $y_i^{P_\mu}$ is a part of the final y_i obtained so far, e.g. $y_i^{P_\mu}$=(y_i^1, y_i^2, ..., $y_i^{\mu-1}$) [Kajdanowicz and Kazienko, 2009b], [Kajdanowicz and Kazienko, 2010]. This composition of x_i and $y_i^{P_\mu}$, i.e. $(x_i, y_i^{P_\mu})$ remains an input vector, but in opposite to the single x_i it also depends on the output $y_i^{P_\mu}$ achieved so far. This concept makes use of the typical nature of the sequential data, in which a given item value y_i^μ may depend somehow on the values of the previous items in the sequence, namely $y_i^1, y_i^2, ..., y_i^{\mu-1}$. For the purpose of the paper the notation $x_i^{P_\mu}$ will denote the composed feature vector build as $(x_i, y_i^{P_\mu})$.

For instance, in the domain of debt recovery sequence prediction (in such case, a sequence consist of repayment indicators for the consecutive months) [Kajdanowicz and Kazienko, 2009b, Kajdanowicz and Kazienko, 2009a, Kajdanowicz and Kazienko, 2010], the input variables $(x_i, y_i^{P_\mu})$ may be the combination of business input information of the particular debt x_i, i.e. x_i's profile, together with the prediction output y^{P_μ} for the preceding periods.

Overall, sequence labelling is a research problem that emerges in many other application domains, among others in protein function classification [Zhang and Zhou, 2006], semantic classification of images [Boutell et al., 2004] or text categorization [Schapire and Singer, 2000].

The main goal of the paper is to present a new boosting method for multi-label classification modelled as sequential output classification, in which the boosting concept would be applied to particular sequence items. In the standard binary classification, boosting is general machine learning concept that relies on the improvement of the final learning result by iteratively training a set of base classifiers on a weighted training set of data. Learning process results in a weighted linear combination of classifiers, trained at each iteration, forming the final meta classifier [Freund and Schapire, 1997, Punyakanok and Roth, 2000].

In order to adapt boosting scheme, it can be modified in two ways: the base classifiers will be able to predict the whole structure (thus the base classifiers need to be multi-label classifiers) or the boosting scheme will be able to transform the problem and generalize it for each label using binary classifiers [Altun et al., 2002].

Sequential output prediction is a particular type of structured prediction because it is likely to be the simplest non-trivial structure. Formally, learning label sequences is a kind of generalization to discover the discriminant function, mapping instances from $\mathcal{X}$ to label sequences $\mathcal{Y}$ [Daume et al., 2009]. Considering the case, that an output vector $y_i = (y_i^1, y_i^2, \dots, y_i^l)$ is a sequence and can be produced by predicting y's all l labels, the boosting scheme may be iteratively revoked for each of l items independently, i.e. each label in the sequence is classified autonomously using the typical boosting algorithm. An extension, introduced in this paper assumes that the input space for the μth sequence item $(2 \leq \mu \leq l)$ is extended with the output of classifications obtained so far for the previous items (from 1 to $\mu - 1$) [Kajdanowicz and Kazienko, 2011]. This is a similar approach to classifier chains considered in the literature.

However another innovative idea is proposed in this paper. The boosting concept is adapted to classify label sequences allowing dependent classification. By treating classification in sequential manner rather than independently for each label [Punyakanok and Roth, 2000], a new algorithm is able to focus on error minimization in prediction of large spans of the output. This is done by the appropriate adjustment of the cost function in boosting. A modified AdaBoost scheme with a sequence-loss balancing cost function, called AdaBoostSeq, is introduced and formally presented in the next Section of the paper.

3 AdaBoostSeq Method for Multi-label Classification

3.1 Description of AdaBoostSeq Method

Based on the most popular boosting algorithm, AdaBoost [Freund and Schapire, 1997], a modification of the cost function has been introduced[1]. It is assumed that there is a binary sequence classification problem with $y_i^\mu \in \{-1, 1\}$, for $i = 1, 2, \ldots, n$ and $\mu = 1, 2, \ldots, l$, where n is the number of instances, l is the length of the sequence. The general goal is to construct l optimally designed linear combinations of K base classifiers of the form:

$$\forall \mu = 1, 2, \ldots, l \quad F^\mu(x^{P_\mu}) = \sum_{k=1}^{K} \alpha_k^\mu \Phi(x^{P_\mu}, \Theta_k^\mu) \tag{1}$$

where: $F^\mu(x^{P_\mu})$ is the combined, final meta classifier for the μth label in sequence; $\Phi(x^{P_\mu}, \Theta_k^\mu)$ represents the kth base classifier, performing according to its Θ_k^μ parameters and returning a binary class label for each instance x used for learning; α_k^μ is the weight associated to the kth classifier.

Values of the unknown parameters result from optimization done for each classified label in the sequence, as follows:

$$\arg \min_{\alpha_k^\mu, \Theta_k^\mu, k:1,K} \sum_{i=1}^{N} \exp(-y_i^\mu F^\mu(x_i^{P_\mu})) \tag{2}$$

As a direct optimization of Eq. 2 is highly complex, a stage-wise suboptimal method is performed. At each step optimization is carried out with respect to a new parameter, leaving unchanged the previously optimized one. Therefore, let us define the result of the partial sum up to m terms (the mth partial sum):

$$F_m^\mu(x^{P_\mu}) = \sum_{k=1}^{m} \alpha_k^\mu \Phi(x^{P_\mu}, \Theta_k^\mu), m = 1, 2, \ldots, K \tag{3}$$

According to the definition from Eq. 3, the following recursion is a natural consequence:

$$F_m^\mu(x^{P_\mu}) = F_{m-1}^\mu(x^{P_\mu}) + \alpha_m \Phi(x^{P_\mu}, \Theta_m^\mu) \tag{4}$$

Due to assumption, that before calculating $F_m^\mu(x^{P_\mu})$, the value of $F_{m-1}^\mu(x^{P_\mu})$ has already been optimized in the previous step, the problem at step m is to compute:

$$(\alpha_m^\mu, \Theta_m^\mu) = \arg \min_{\alpha^\mu, \Theta^\mu} J(\alpha^\mu, \Theta^\mu) \tag{5}$$

[1] The following description is based on authors' previous work [Kajdanowicz and Kazienko, 2011].

where the newly proposed sequence-loss cost balancing function J is defined as:

$$J(\alpha^\mu, \Theta^\mu) = \sum_{i=1}^{N} \exp(-y_i^\mu(\xi F_{m-1}^\mu(x_i^{P_\mu}) + (1-\xi)y_i^\mu \hat{R}_m^\mu(x_i^{P_\mu}) + \alpha^\mu \Phi(x_i^{P_\mu}, \Theta^\mu))) \tag{6}$$

where: $\hat{R}_m^\mu(x_i^{P_\mu})$ is an impact function denoting the influence on classification according to the quality of preceding sequence labels predictions; ξ is the parameter that allows controlling the influence of impact function in weights composition, $\xi \in \langle 0, 1 \rangle$.

$\hat{R}_m^\mu(x_i^{P_\mu})$ is applied in computation for current sequence position, as follows:

$$\hat{R}_m^\mu(x_i^{P_\mu}) = \sum_{i=1}^{m-1} \alpha_i^\mu R^\mu(x_i^{P_\mu}) \tag{7}$$

$$R^\mu(x_i^{P_\mu}) = \frac{\sum_{j=1}^{\mu-1} y_i^j \frac{F_j(x_i^{P_\mu})}{\sum_{k=1}^{K} \alpha_k^l}}{\mu} \tag{8}$$

where: $R^\mu(x_i^{P_\mu})$ is the auxiliary function that denotes the average coincidence between classification result $F_j(x_i^{P_\mu})$ and the actual value y_i^j weighted with the weights α_k^j associated to the kth base classifier with respect to value of μ.

The impact function $\hat{R}_m^\mu(x_i^{P_\mu})$, introduced in Eq. 7 and 8, measures the correctness of classification for all preceding labels in the sequence for each instance. This function is utilized in the cost function and by minimizing it is able to provide smaller error deviation for the whole sequence. The greater compliance between prediction and the real value, the higher the function value. Considering α^μ as a fixed constant (Eq. 5 and 6), the cost function J may be optimized with respect to the base classifier $\Phi(x_i^{P_\mu}, \Theta^\mu)$ that is simplified to:

$$\Theta^\mu = \arg\min_{\Theta^\mu} \sum_{i=1}^{N} w_{i(m)}^\mu \exp(-y_i^\mu \alpha^\mu \Phi(x_i^{P_\mu}, \Theta^\mu)) \tag{9}$$

where:

$$w_{i(m)}^\mu = \exp(-y_i^\mu(\xi F_{m-1}^\mu(x_i^{P_\mu}) + (1-\xi)y_i^\mu \hat{R}_m^\mu(x_i^{P_\mu}))) \tag{10}$$

As $w_{i(m)}^\mu$ depends on neither α^μ nor $\Phi(x_i^{P_\mu}, \Theta^\mu)$ for each $x_i^{P_\mu}$, $w_{i(m)}^\mu$ can be treated as a weight of sample instance $x_i^{P_\mu}$. Due to binary nature of the base classifier, minimization of Θ^μ is equivalent to:

$$\Theta^\mu = \arg\min_{\Theta^\mu} \left\{ P_m^\mu = \sum_{i=1}^{N} w_{i(m)}^\mu I(1 - y_i^\mu \Phi(x_i^{P_\mu}, \Theta^\mu)) \right\} \tag{11}$$

where

$$I(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{12}$$

For the base classifier computed at step m, we have:

$$\sum_{y_i^\mu \Phi(x_i^{P_\mu}, \Theta_m^\mu) < 0} w_{i(m)}^\mu = P_m^\mu \tag{13}$$

$$\sum_{y_i^\mu \Phi(x_i^{P_\mu}, \Theta_m^\mu) > 0} w_{i(m)}^\mu = 1 - P_m^\mu \tag{14}$$

and the optimum value of α_m results from:

$$\alpha_m^\mu = \arg\min_{\alpha^\mu} \{\exp(-\alpha^\mu)(1 - P_m^\mu) + \exp(\alpha^\mu) P_m^\mu\} \tag{15}$$

The derivation of Eq. 15 with respect to α equaled to zero, results in:

$$\alpha_m^\mu = \frac{1}{2} \ln \frac{1 - P_m^\mu}{P_m^\mu} \tag{16}$$

Once the base classifier $\Phi(x_i^{P_\mu}, \Theta^\mu)$ and α_m^μ are computed, the weights for step $m + 1$ may be calculated:

$$\begin{aligned} w_{i(m+1)}^\mu &= \frac{\exp\left(-y_i^\mu \left(\xi^{P_\mu} F_m^\mu(x_i^{P_\mu}) + (1-\xi) y_i^\mu \hat{R}_m^\mu(x^{P_\mu})\right)\right)}{Z_m} = \\ &= \frac{w_{i(m)} \exp\left(-y_i^\mu \xi \alpha_m^\mu \Phi(x_i^{P_\mu}, \Theta_m^\mu) - (1-\xi)\alpha_m^\mu R^\mu(x^{P_\mu})\right)}{Z_m} \end{aligned} \tag{17}$$

where Z_m is the normalizing factor, as follows:

$$Z_m = \sum_{i=1}^{N} w_{i(m)}^\mu \exp\left(-y_i^\mu \xi \alpha_m^\mu \Phi(x_i^{P_\mu}, \Theta_m^\mu) - (1-\xi)\alpha_m^\mu R^\mu(x_i^{P_\mu})\right) \tag{18}$$

It is worth mentioning that the weight of the particular instance $w_{i(m+1)}^\mu$ changes with respect to its value at the previous iteration step m. Simultaneously, the value of the impact function $\hat{R}_m^\mu(x^{P_\mu})$, which denotes the correctness in classification for the preceding labels in the sequence, changes as well. The pseudo code of AdaBoostSeq for multi-label classification is presented in Algorithm 1.

The typical termination criterion used in the above algorithm is $k \leq K$, where K denotes the number of base classifiers. The final prediction after learning according to Algorithm 1 is performed by calculation of Eq. 19.

$$\forall \mu = 1, 2, \ldots, l \quad F^\mu(.) = sign(\sum_{k=1}^{K} \alpha_k^\mu \Phi(., \Theta_k^\mu)) \tag{19}$$

Algorithm 1 The pseudo code of learning AdaBoostSeq algorithm for multi-label sequential classification.

Input: K base classifiers Φ, data set $(\mathcal{X}, \mathcal{Y})$
Output: learned combined classifier F

```
for each label in sequence (μ = 1 to l) do
    Initialize w^μ_{i(1)} = 1/N, m = 1;
    while termination criterion is not met do
        Compute Θ^μ_m (parameters of Φ^μ_m) and P^μ_m (Eq. 11);
        Compute α^μ_m = 1/2 ln ((1 − P^μ_m) / P^μ_m);
        Set Z_m = 0;
        for each instance (i = 1 to N) do
            Compute w^μ_{i(m+1)} = w^μ_{i(m)} ·
                · exp(−y^μ_i ξ α^μ_m Φ(x_i^{P_μ}, Θ^μ_m) − (1 − ξ) α_m R^μ(x_i^{P_μ}));
            Sum up Z_m = Z_m + w^μ_{i(m+1)};
        end for
        for each instance (i = 1 to N) do
            Normalize w^μ_{i(m+1)} = w^μ_{i(m+1)} / Z_m;
        end for
        Set k = m, m = m + 1;
    end while
end for
```

Concluding, the general idea behind AdaBoostSeq algorithm is to perform for each label in sequence a loop of four algorithmic steps done for all base classifiers, see Fig. 1. Firstly the parameters of a base classifier are obtained Θ^μ_m(considering weights assigned to instances) as well as the sum of weights of incorrectly classified instances P^μ_m. Secondly the weight α^μ_m for the base classifier is computed. In the next step the weights w^μ_m assigned to instances are recalculated and in final step - normalized. This four step process is realized for all base classifiers. The number of base classifiers is a parameter of the algorithm.

3.2 Complexity of AdaBoostSeq Method

The computational complexity of AdaBoostSeq method depends outright on the size of label set $|\mathcal{L}|$ (modelled here as a sequence). It can be quantified as $O(|\mathcal{L}| \times K \times f(d + |\mathcal{L}|, N))$, where $f(d + |\mathcal{L}|, N)$ denotes the complexity of underlying base classifier performing on data set with d attributes and N instances. According to the nature of AdaBoostSeq method each label is generalized by K base classifiers.

In contrast to binary relevance classification approach AdaBoostSeq method does not manifest ability to be easily parallelized in the training phase. This is

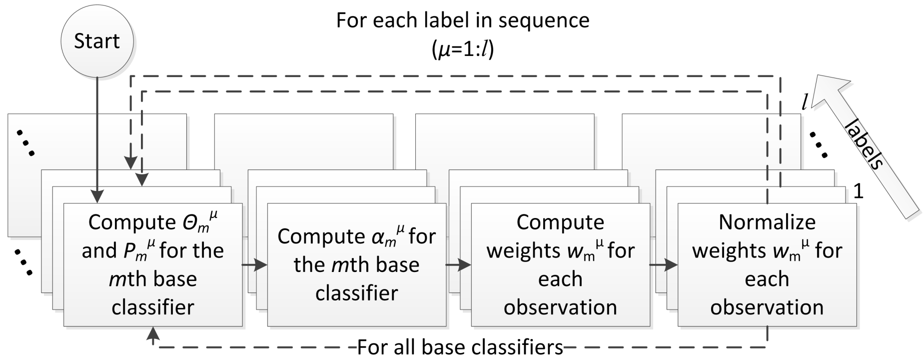

Figure 1: General ilustration of the steps performed by AdaBoostSeq algorithm.

due to the requirement of calculating the weights based on the error obtained in generalization of previous labels in the sequence.

4 Experimental Study of AdaBoostSeq Algorithm

The AdaBoostSeq algorithm was examined in order to check its properties in two groups of experiments. First group of experiments was accomplished in order to examine the robustness of the method by changing the amount of information given in the training. Namely, there were conducted experiments in scenarios where learning was based on 10%, 20%, ..., 100% number of attributes from the original input space. This kind of tests were carried out in order to observe whether the proposed approach is very sensitive to incompleteness of input data. The second group of experiments was performed to figure out how the approach behaves for distinct size of output space. There were conducted experiments examining the algorithm for distinct number of labels from label-space, from 10% of original number of labels in the original problem, up to 100% of labels.

4.1 Experimental Setup

The experiments were carried out on six distinct data sets from four diverse application domains. Detailed description of data sets can be found in next Section. The main objective of the performed experiments was to evaluate the accuracy and efficiency of the Classifier Chain method in distinct scenarios of input and output space. All scenarios were examining Hamming Loss (HL) [Schapire and Singer, 2000], Classification Accuracy (CA) [Ghamrawi and McCallum, 2005], Accuracy (ACC) [Godbole and Sarawagi, 2004], F-macro, F-micro and computation time spent on model training ($TIME$), separately for six distinct data sets.

What needs to be explained are macro and micro label-based evaluation measures.Instead of being calculated separately for each instance, they are evaluated separately for each label and averaged across all labels. Label based evaluation measures can be divided into macro-averaging measures and micro-averaging measures. The former measures are calculated as an average measure obtained for each of labels separately. Using the notation where *tp* denotes *true-positive*, *fp* - *false-positive*, *tn true-negative* and *fn* - *false-negative* [Bishop, 2006] each macro averaging measure M_{macro} can be calculated as in Eq. 20.

$$M_{macro} = \frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} M(tp_i, fp_i, tn_i, fn_i) \tag{20}$$

It is assumed that the particular measure $M(tp, fp, tn, fn)$ is a binary evaluation measure calculated on the contingency table. It means that macro-averaging measures ordinary average a binary measure.

On the other hand, the latter, micro-averaging measures, are calculated for all labels jointly. We can observe that in micro-averaging measures labels are treated as different instances of the same global label. It is expressed by the the summation term for each of *tp*, *fp*, *tn* and *fn* counts, see Eq. 21.

$$M_{micro} = M(\sum_{i=1}^{|\mathcal{L}|} tp_i, \sum_{i=1}^{|\mathcal{L}|} fp_i, \sum_{i=1}^{|\mathcal{L}|} tn_i, \sum_{i=1}^{|\mathcal{L}|} fn_i) \tag{21}$$

An example of evaluation measure that can be utilized to assess the multi-label classification in its macro and micro version is F-measure [Bishop, 2006].

The performance of the analysed methods was evaluated on original, taken from data source, training–test split of data sets. It was assumed that training and testing data instances were obtained from carefully designed data gathering process. Preserving the original data split enabled verification of obtained results with independent research carried out by other scientists.

AdaBoostSeq as a ensemble algorithm [Smetek and Trawinski, 2011,Wozniak and Krawczyk, 2012] required binary base learner. For that purpose three types of classifiers were examined: the k-Nearest Neighbour (kNN) with $k = 3$, the compound Random Forest classifier ($RAND$) with 200 trees and 20% of features selected randomly at each tree stage, and Support Vector Machine SVM with RBF kernel used with sigma and C parameters selected automatically based on grid search. The experiments were implemented in the Matlab environment.

4.2 Data sets

In order to evaluate and compare all proposed approaches, the experiments were carried out on six distinct data sets from four diverse application domains: semantic scene analysis, bioinformatics, music categorization and text processing.

The image data set *scene* [Boutell et al., 2004] semantically indexes still scenes. The biological data set *yeast* [Elisseeff and Weston, 2001] concerns micro-array expressions and phylogenetic profiles for genes classification. The music data set *emotions* [Trohidis et al., 2008], in turn, contains data about songs categorized into one or more classes of emotions. The *medical* [Pestian et al., 2007] data set is based on the Computational Medicine Center's 2007 Medical Natural Language Processing Challenge and contains clinical free text reports labelled with disease codes. Another data set, *enron*, is based on annotated email messages exchanged between Enron Corporation employees. The last data set, *genbase* [Diplaris et al., 2005] refers to protein classification.

4.3 Evaluation of AdaBoostSeq for Distinct Size of Input Space

In order to evaluate the robustness on completeness of input data used for training the experiments examined accuracy of the method for distinct amount of information given in the training. There were conducted experiments where learning was based on 10%, 20%, . . . , 100% number of attributes from the original input space. Selection of attributes was based on *forward selection* method [Guyon and Elisseeff, 2003]. In general this method selects stated number of best attributes from all available attributes. The decision is based on discrimination abilities obtained in generalization using attributes selected in forward manner [Guyon and Elisseeff, 2003]. For instance, when the learning phase was based on 10% of attributes it meant that it used 10% of the best attributes provided from forward selection method.

The results obtained from experiments are presented in Figures 2, 3, 4, 5, 6 and 7.

According to obtained results it can be observed that for examined data sets collected majority of accuracy measures tend to be related to the amount of input description provided to AdaBoostSeq. In general while increasing the number of attributes used in training, the method obtains higher Classification Accuracy (CA), F-micro and F-macro results. However, it is worth noting that the method is not that much sensitive to input space description resulting with almoust constant Hamming Loss (HL) measure in *emotions*, *scene*, *yeast*, *medical* and *enron* data sets. This property is an advantage of the method.

The best results across all evaluation measures were obtained with AdaBoostSeq algorithm that utilized Random Forest compound classifier as a base classifier. Other two base classifiers, kNN and SVM obtained worse results then Random Forest. It was surprising that SVM was performing the worst, providing inferior result while increasing number of attributes used for training.

Considering F-macro and F-micro measures it can be concluded that for one part of data sets (*emotions*, *yeast* and *enron*) they seem to be weakly dependent on amount of input attributes, whereas for other (*scene*, *medical* and

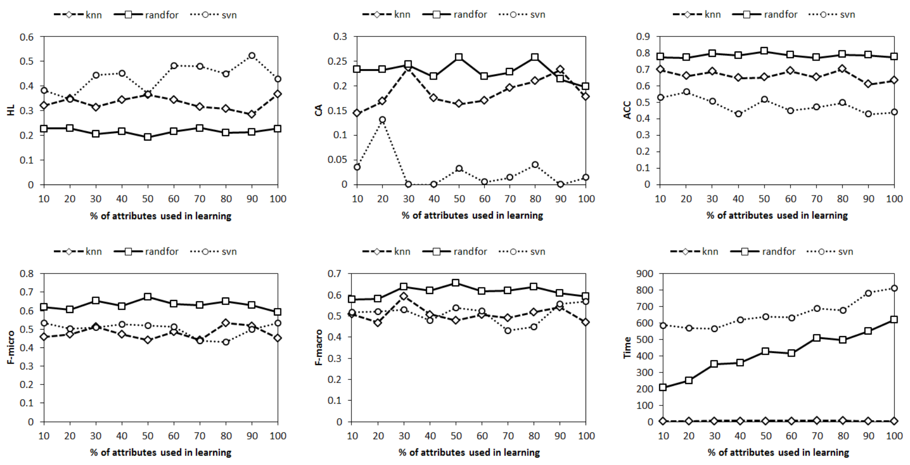

Figure 2: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of input space for *emotions* data set.

genbase) increasing the number of input attributes improves them significantly. This statement is valid for all three base classifiers.

Comparing all three examined base classifiers it needs to be emphasized that all of them preserved linear scalability of computational cost measured by execution time. SVM was the most demanding base classifier while kNN outperformed other two classifiers requiring from one to two orders of magnitude less time.

Summarizing, AdaBoostSeq method that utilizes Random Forest as base classifier is the best choice from all three examined base classifiers. It provides the best Hamming Loss, Classification Accuracy and F-measures among all data sets. Even though it is quite demanding in computational resources in comparison to kNN, it can be treated as the best and robust solution that is resistant to incompleteness of input attributes to some extent.

4.4 Evaluation of AdaBoostSeq for Distinct Size of Output Space

The second group of experiments was performed in order to figure out how the proposed method behaves for distinct size of output space. There were conducted experiments examining the AdaBoostSeq method for distinct number of labels from label-space, from 10% of original number of labels in the original problem, up to 100% of labels. Each part of labels was obtained from random draw from whole label-space.

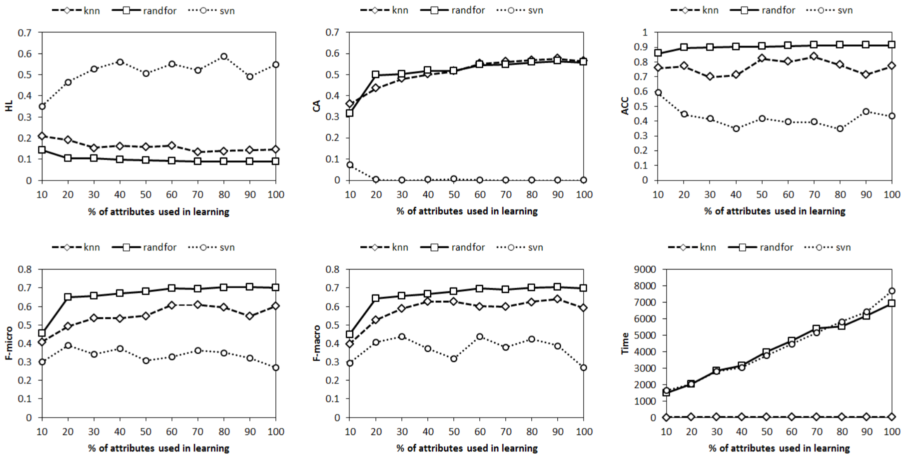

Figure 3: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of input space for *scene* data set.

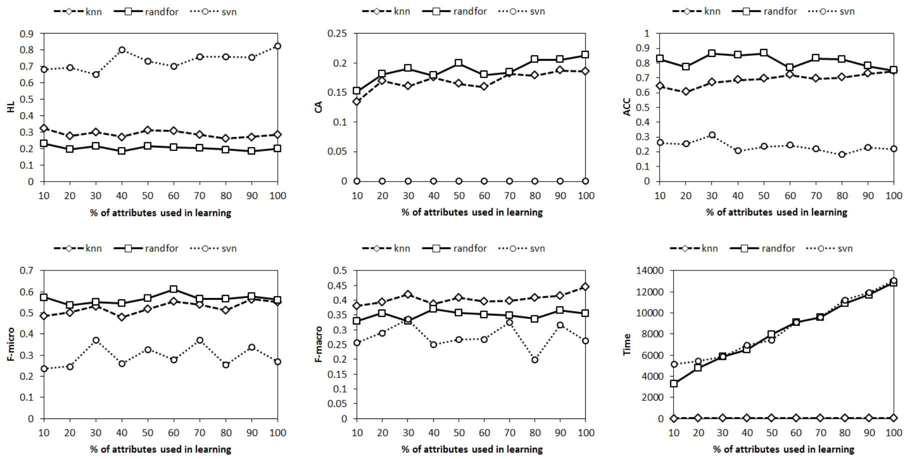

Figure 4: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of input space for *yeast* data set.

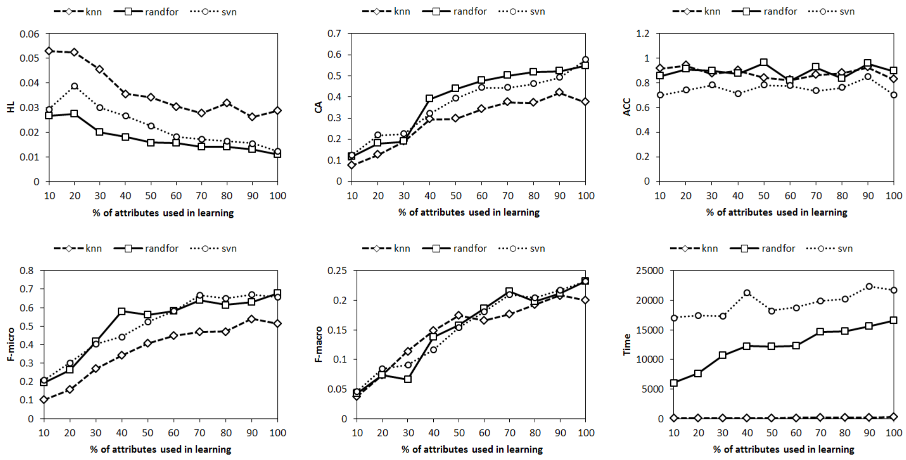

Figure 5: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of input space for *medical* data set.

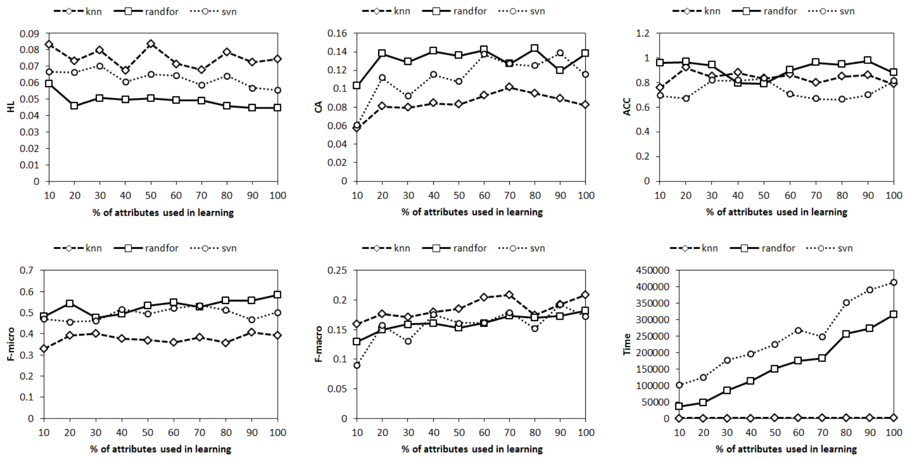

Figure 6: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of input space for *enron* data set.

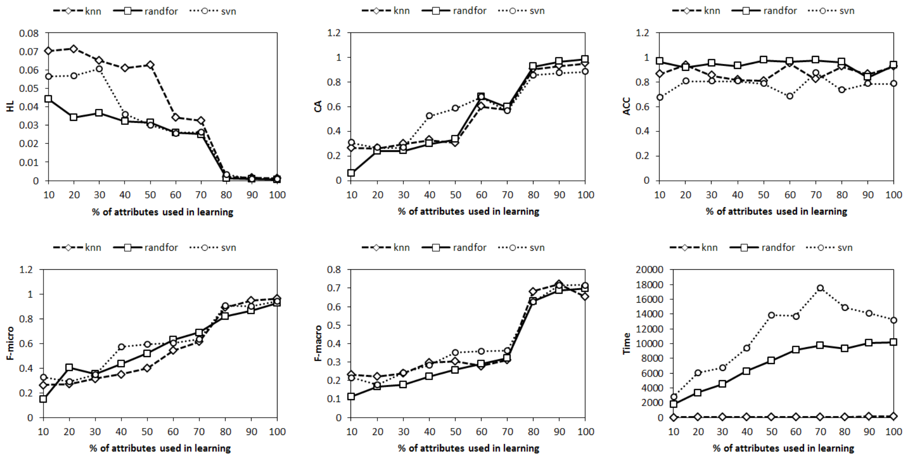

Figure 7: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of input space for *genbase* data set.

Following the experimental results it can be noticed that for all evaluation measures and for all data sets Random Forest compound classifier outperformed both, kNN and SVM methods. Again it was less time demanding than SVM but much more complex than simple kNN. The results obtained from experiments are presented in Figures 8, 9, 10, 11, 12 and 13.

It can be observed in all the considered data sets that HL is generally constant regardless of the number of labels the method generalizes. It can be noticed that only in *enron* and *genbase* data sets it slightly rises. On the other hand CA evaluation measure tends to be related to the size of generalized label-space. While increasing the number of labels to be learnt it falls significantly, except in *scene* data set.

Considering F-macro and F-micro evaluation measures it can be observed that only for *enron* data set when increasing the number of labels being learnt, AdaBoostSeq provides higher value of measures. For all the rest of data sets measures decrease with more labels generalized.

Comparing all three examined base classifiers it needs to be emphasized that all of them preserved linear scalability of computational cost measured by execution time while increasing the number of generalized labels. However, again the Random Forest outperformed kNN and SVM in all evaluation measures.

Concluding, AdaBoostSeq method that utilizes Random Forest as a base classifier is again the best choice from all three examined base classifiers. It

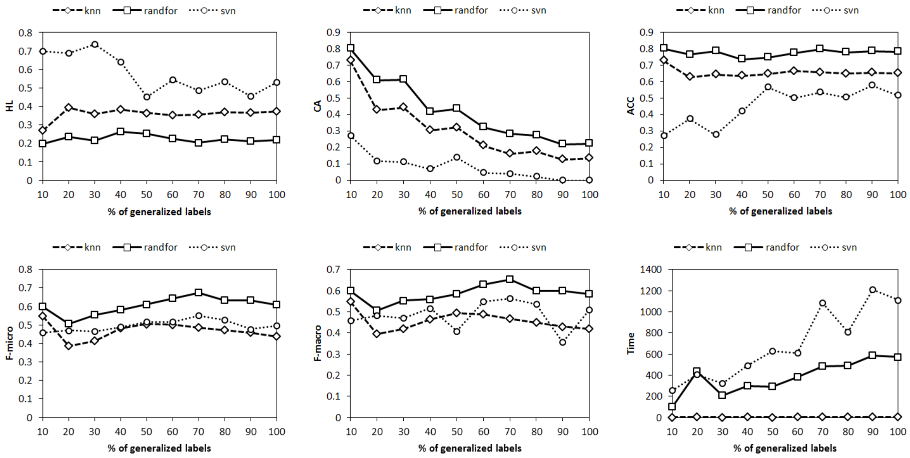

Figure 8: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of generalized output space for *emotions* data set.

provides the best Hamming Loss, Classification Accuracy and F-measures among all data sets for distinct parts of label-space. In general AdaBoostSeq algorithm may be considered as resistant to the size of output-space while HL is the criterion. Increasing the number of learnt labels unfortunately affects all other evaluation measures and decreases its predictive performance.

5 Conclusions

An approach to mulit-label classification of label sequences using a concept of boosting was examined in this paper. It requires a specific cost function to be utilized in the new algorithm AdaBoostSeq. This function respects both the classification error on the current sequence item and the average error on all other preceding items while performing boosting iterations. Following the consideration on the computational complexity as well as the examined predictive performance, AdaBoostSeq is a valuable proposal for multi-label classification. In general it can be stated that it is able to result with quite stable Hamming Loss regardless of the size of input and output space.

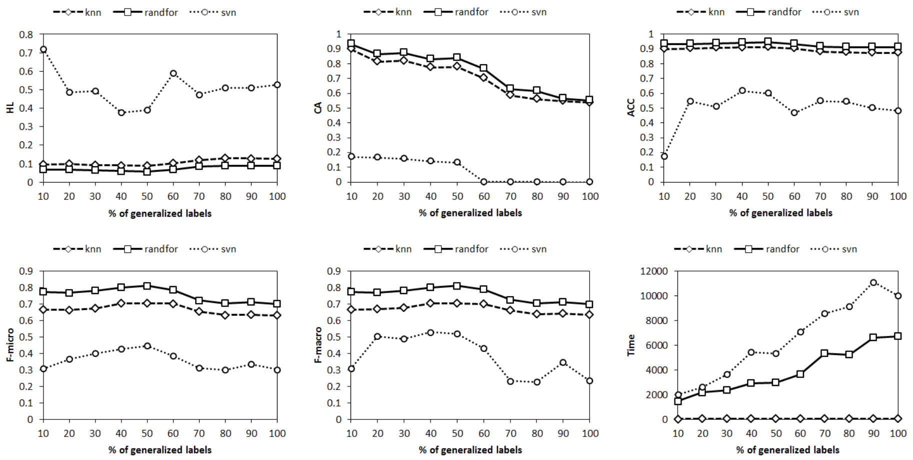

Figure 9: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of generalized output space for *scene* data set.

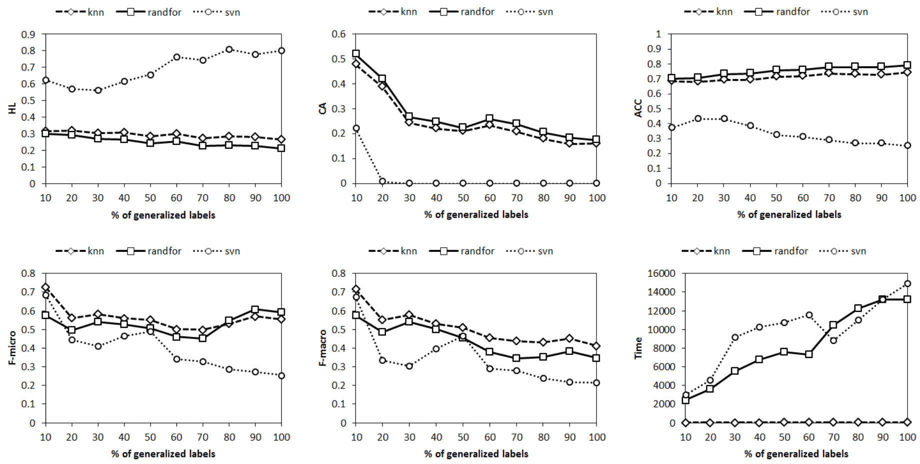

Figure 10: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of generalized output space for *yeast* data set.

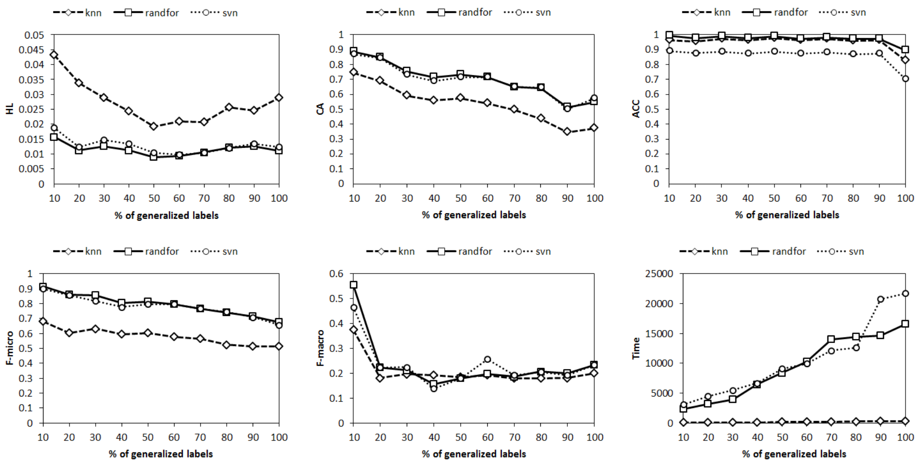

Figure 11: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of generalized output space for *medical* data set.

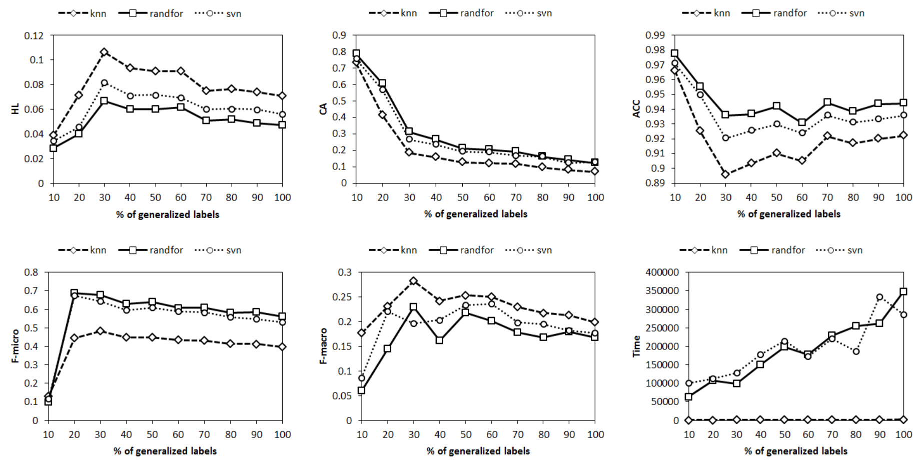

Figure 12: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of generalized output space for *enron* data set.

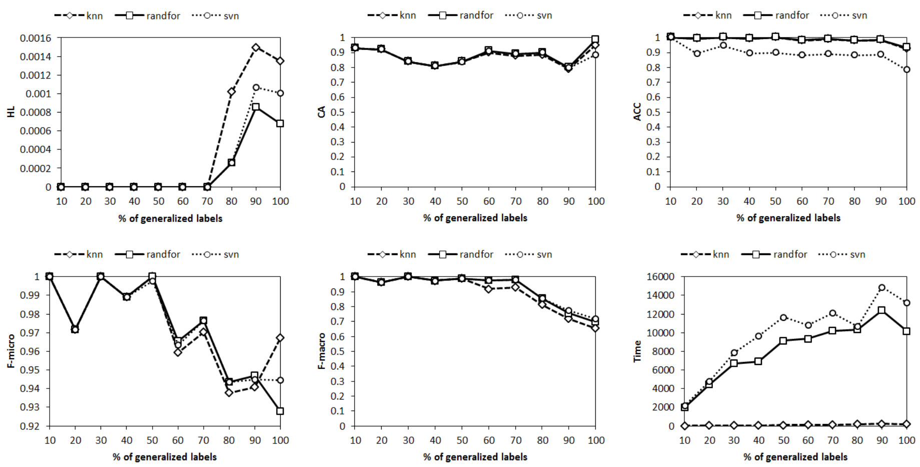

Figure 13: Hamming Loss (HL), Classification Accuracy (CA), Accuracy (ACC), F-micro, F-macro and computation time (Time) results obtained by AdaBoost-Seq for distinct size (percentage) of generalized output space for *genbase* data set.

References

[Altun et al., 2002] Altun, Y., Hofmann, T., and Johnson, M. (2002). Discriminative learning for label sequences via boosting. In *Advances in Neural Information Processing Systems*, Neural Information Processing Systems, pages 977–984. MIT Press.

[Bishop, 2006] Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

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

[Daume et al., 2009] Daume, H., Langford, J., and Marcu, D. (2009). Search-based structured prediction. *Machine Learning*, 75(3):297–325.

[Diplaris et al., 2005] Diplaris, S., Tsoumakas, G., Mitkas, P., and Vlahavas, I. (2005). Protein classification with multiple algorithms. In *Proceedings of 10 th Panhelllenic Conference on Informatics, PCI'2005*, volume 3746 of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 448–456. Spring-Verlag.

[Elisseeff and Weston, 2001] Elisseeff, A. and Weston, J. (2001). A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems*, Neural Information Processing Systems, pages 681–687. MIT Press.

[Freund and Schapire, 1997] Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Science*, 55(1):119–139.

[Ghamrawi and McCallum, 2005] Ghamrawi, N. and McCallum, A. (2005). Collective multi-label classification. In *Proceedings of International Conference on Information and Knowledge Management*, pages 195–200. ACM.

[Godbole and Sarawagi, 2004] Godbole, S. and Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD'2004*, pages 22–30.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182.

[Kajdanowicz and Kazienko, 2009a] Kajdanowicz, T. and Kazienko, P. (2009a). Hybrid repayment prediction for debt portfolio. volume 5796 of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 850–857. Springer Berlin / Heidelberg.

[Kajdanowicz and Kazienko, 2009b] Kajdanowicz, T. and Kazienko, P. (2009b). Prediction of sequential values for debt recovery. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 5856 of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 337–344. Springer Berlin / Heidelberg.

[Kajdanowicz and Kazienko, 2010] Kajdanowicz, T. and Kazienko, P. (2010). Incremental prediction for sequential data. volume 5991 of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 359–367. Springer Berlin/Heidelberg.

[Kajdanowicz and Kazienko, 2011] Kajdanowicz, T. and Kazienko, P. (2011). Boosting-based sequential output prediction. *New Generation Computing*, 29(3):293–307.

[Lee et al., 2011] Lee, S. M., Kim, D. S., and Park, J. S. (2011). Cost-sensitive spam detection using parameters optimization and feature selection. *Journal of Universal Computer Science*, 17(6):944–960.

[Pestian et al., 2007] Pestian, J., Brew, C., Matykiewicz, P., Hovermale, D., N., J., Bretonnel Cohen, K., and Duch, W. (2007). A shared task involving multi-label classification of clinical free text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*. Association of Computational Linguistics.

[Punyakanok and Roth, 2000] Punyakanok, V. and Roth, D. (2000). The use of classifiers in sequential inference. Technical report, Champaign, IL, USA.

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

[Smetek and Trawinski, 2011] Smetek, M. and Trawinski, B. (2011). Selection of heterogeneous fuzzy model ensembles using self-adaptive genetic algorithms. *New Generation Computing*, 29:309–327.

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

[Wozniak and Krawczyk, 2012] Wozniak, M. and Krawczyk, B. (2012). Combined classifier based on feature space partitioning. *International Journal of Applied Mathematics and Computer Science*, 22(4):855–866.

[Zhang and Zhou, 2006] Zhang, M. and Zhou, Z. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transaction on Knowledge and Data Engineering*, 18:1338–1351.