Pattern Recognition Letters 40 (2014) 56–65

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Multi-label learning under feature extraction budgets ☆

Pekka Naula *, Antti Airola, Tapio Salakoski, Tapio Pahikkala

Department of Information Technology, University of Turku, 20014, Finland

ARTICLE INFO

Article history:
Received 23 October 2012
Available online 24 December 2013

Keywords:
Feature selection
Greedy forward selection
Multi-label learning
Regularized least-squares

ABSTRACT

We consider the problem of learning sparse linear models for multi-label prediction tasks under a hard constraint on the number of features. Such budget constraints are important in domains where the acquisition of the feature values is costly. We propose a greedy multi-label regularized least-squares algorithm that solves this problem by combining greedy forward selection search with a cross-validation based selection criterion in order to choose, which features to include in the model. We present a highly efficient algorithm for implementing this procedure with linear time and space complexities. This is achieved through the use of matrix update formulas for speeding up feature addition and cross-validation computations. Experimentally, we demonstrate that the approach allows finding sparse accurate predictors on a wide range of benchmark problems, typically outperforming the multi-task lasso baseline method when the budget is small.

1. Introduction

Multi-label learning [1] concerns the problem of learning to make predictions about the association between data points and a set of candidate labels. In multi-label classification, one aims to predict which of the available labels are relevant with respect to the data point of interest, and which are not. In label ranking (see e.g. [2]) one rather predicts the ordering over the set of labels, where the labels best matching the data point appear at the top of the ordering. The applications of multi-label learning are varied, since in almost any domain of interest there are usually several interesting properties that can be simultaneously used to describe an object. For example, an image often has several objects appearing in it, a piece of music or a movie represents multiple genres, or a newspaper article may belong to several topic categories.

Multi-label methods are often divided into two categories: problem transformation methods and algorithm adaptation methods [3]. The former aim at dividing the original problem into one or more single-label classification or regression problems whereas the latter are based on extending existing single-task approaches to multi-label learning. There are a rich family of different approaches for both categories.

Two of the most common problem transformation methods are binary relevance method (BR) and label power-set method (LP). While BR divides the multi-label problem into binary single-task problems, one task per label, LP creates a binary single-label problem for every possible label combination. Compared to BR, LP has the advantage of being able to model the correlation between the labels, but this comes at a steep computational price as the number of possible label combinations grows exponentially with respect to the size of the label set. More advanced transformation methods such as RAKEL [4] have been developed to overcome this problem. Examples of single-task classifiers adapted to make use of label correlations include the ML-kNN [5] algorithm, that extends the K-nearest neighbors algorithm to multi-label classification, and the ML-C4.5 [6] multi-label decision tree method. For a comprehensive overview and experimental comparison of multi-label methods, we refer to Madjarov et al. [7].

In this work we consider the BR type of setting, where for each label one constructs a linear predictor, that produces scorings from which the classifications or rankings are derived. In many applications *sparsity*, meaning that for a significant number of features the corresponding coefficients in the models are set to zero, is a desirable property. The three most common motivations for learning sparse models are the following. Enforcing sparsity has a regularizing effect which may help to prevent overfitting, models depending only on a few variables are easier to understand and explain by human experts, and sparse models are cheaper to predict with than dense ones. The focus of this paper is especially on the third point of view.

As pointed out by Xu et al. [8], the prediction cost can, in turn, be divided into the times required for evaluating the models and for extracting the feature values. For linear models, the evaluation time is proportional to the number of nonzero model entries, totaled over all models multi-label learning. In contrast, the feature extraction time is proportional to the set of unique features

☆ This paper has been recommended for acceptance by S. Sarkar.

* Corresponding author at: Department of Information Technology, 20014, University of Turku, Finland. Tel.: +358 405022708; fax: +358 22410154.

E-mail addresses: pekka.naula@utu.fi (P. Naula), antti.airola@utu.fi (A. Airola), tapio.salakoski@utu.fi (T. Salakoski), tapio.pahikkala@utu.fi (T. Pahikkala).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.12.009

used for prediction. The feature value is extracted only once for a single data point, while the value can be used to predict several labels. The difference between the two types of sparsity is illustrated in the following example, where two linear models have the same model evaluation cost, but different feature extraction cost. Let

$$\mathbf{W}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \end{pmatrix}, \quad \mathbf{W}_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 2 & 3 & -1 & 2 \\ 0 & 0 & 0 & 0 \\ 3 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

denote the matrices determining two sparse linear models. The rows and columns of both matrices correspond to features and tasks, respectively. Both matrices have the same number of non-zero coefficients, but $\mathbf{W}_1$ requires all the features for prediction, whereas $\mathbf{W}_2$ requires only two of them.

The feature extraction costs are dominant to the model evaluation costs in many real-world tasks, and hence the focus of this paper is the minimization of the extraction cost. Our problem definition is quite similar to that of *budgeted learning* considered recently by Cesa-Bianchi et al. [9] and Hazan and Koren [10], the difference being that our work considers multi-label instead of single-task learning, and we do not consider settings where different features may be selected for different data points.

As a motivating example, consider an image recognition system that simultaneously predicts several properties of a given input image in real-time. Since each feature used for prediction is obtained from a possibly computationally expensive feature extractor, one must minimize the number of required features to ensure real-time recognition. A similar setting is commonly encountered in medical testing, where we want to perform as few tests as possible, yet make reliable diagnoses for a patient. To summarize, we consider the setting in which the number of features must be limited even if it decreases the prediction performance, because enforcing sparsity due to the high feature acquisition costs is necessary in numerous practical applications.

Two popular approaches for learning sparse models are the filter methods, that perform feature selection independently of the learning algorithm trained on the selected features, and wrapper or embedded methods where the selection process is optimized for the learning algorithm. The most prominent of the latter type of methods are the method known as lasso or basis pursuit, and the family of greedy search algorithms. There is empirical evidence in the literature favoring lasso over greedy methods [11] when the amount of selected features is large. Moreover, it has been shown that if the model underlying the data is truly sparse lasso converges to it [12]. However, in the setting considered in this work one must select only a small number of features even if the model is not truly sparse. Consequently, since the lasso methods are based on convex regularization, the smaller is the set of selected features, the worse will be the bias caused by the regularization on the learnt model [13]. This phenomenon does not concern the greedy methods, as they are based on a different selection principle.

In the recent years, techniques applicable to learning sparse models in the single-label setting have been extended to the multi-label setting. As a typical example of filter methods, Doquire and Verleysen [14] proposed a greedy method that combines a mutual information based selection criterion with a variant of the LP transformation method. Zhang et al. [15] proposed a naive Bayes multi-label method that applies as a first stage principal component analysis in order to reduce the feature set dimensionality followed by a genetic algorithm based feature selection phase. However, the reliance on PCA for dimensionality reduction makes this and similar methods unsuitable for the setting considered in this work, as they still need all the original features during prediction time.

Among the selection methods optimized for the learning algorithm, sparsity enforcing matrix norm-based regularization approaches, that extend the commonly used l_1-norm to the multi-task setting, have shown to be especially promising [16–19]. As a representative of the state-of-the art in this area, we consider the coordinate descent training approach for the $l_{1,\infty}$-regularization based multi-task lasso [17]. The optimization criterion for the method directly enforces such sparsity structure that leads to minimal number of features being used in the model (see matrix $\mathbf{W}_2$). Thus, the method provides a natural baseline for comparing our work.

We extend the greedy RLS approach [20,21], a greedy forward selection method for regularized least-squares proposed by some of the present authors, to multi-label setting. The work continues the work of Naula et al. [22,23], where a high-level description of the idea and some preliminary experimental results were presented. We prove that the resulting training algorithm has linear time and space complexities, making it computationally highly competitive for example with the most efficient known coordinate descent training algorithms proposed for the lasso-type of learning methods. In our experiments, we compare the predictive performance of the multi-label greedy RLS and multi-task lasso approaches over several real-world data sets, in order to determine which approach, if any, leads to higher predictive performance. The results suggest that whenever one wants to strongly enforce sparsity, the greedy approach is preferable, as on small feature subsets multi-label greedy RLS consistently outperforms multi-task lasso.

2. Methods

Here, we present the basic concepts and notations relevant for the following considerations. By $[n]$ we denote the index set $\{1 \dots n\}$. We use bold lowercase and uppercase letters for denoting vectors and matrices, respectively. Given a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ and index sets $\mathcal{R} \subseteq [m]$ and $\mathcal{S} \subseteq [n]$, we use $\mathbf{M}_{\mathcal{R},\mathcal{S}}$ for denoting the submatrix containing the rows and columns indexed by $\mathcal{R}$ and $\mathcal{S}$, respectively. Further, $\mathbf{M}_{\mathcal{R},}$, $\mathbf{M}_{,\mathcal{S}}$, and $\mathbf{M}_{i,j}$ are shorthands for, $\mathbf{M}_{\mathcal{R},[n]}$, $\mathbf{M}_{[m],\mathcal{S}}$, and $\mathbf{M}_{\{i\},\{j\}}$, respectively. We use analogous notations also for vectors.

Let

$$D = \{(\mathbf{x}^1, \mathbf{y}^1), \dots, (\mathbf{x}^n, \mathbf{y}^n)\},$$

be a training set of size n, where $\mathbf{x}^i \in \mathbb{R}^d$ and $\mathbf{y}^i \in \mathbb{R}^t$ are the feature and the label vectors of the ith instance, respectively, and d and t are the numbers of features and labels. The label vectors can be encoded so that $\mathbf{y}^i_j = 1$ if the ith instance is associated with the jth label and $\mathbf{y}^i_j = -1$ otherwise.

Our aim is to learn from D a real valued function

$$f_l : \mathbb{R}^d \to \mathbb{R}.$$