Neurocomputing 169 (2015) 187–195

Contents lists available at ScienceDirect

Neurocomputing

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

Kernel methods for heterogeneous feature selection

Jérôme Paul*, Roberto D'Ambrosio, Pierre Dupont

Université catholique de Louvain – ICTEAM/Machine Learning Group[1], Place Sainte Barbe 2 bte L5.02.01, B-1348 Louvain-la-Neuve, Belgium

ARTICLE INFO

Article history:
Received 30 June 2014
Received in revised form
12 December 2014
Accepted 29 December 2014
Available online 16 April 2015

Keywords:
Heterogeneous feature selection
Kernel methods
Mixed data
Multiple kernel learning
Support vector machine
Recursive feature elimination

ABSTRACT

This paper introduces two feature selection methods to deal with heterogeneous data that include continuous and categorical variables. We propose to plug a dedicated kernel that handles both kinds of variables into a Recursive Feature Elimination procedure using either a non-linear SVM or Multiple Kernel Learning. These methods are shown to offer state-of-the-art performances on a variety of high-dimensional classification tasks.

1. Introduction

Feature selection is an important preprocessing step in machine learning and data mining as increasingly more data are available and problems with hundreds or thousands of features have become common. Those high dimensional data appear in many areas such as gene expression array analysis, text processing of internet documents, and economic forecasting. Feature selection allows domain experts to interpret a decision model by reducing the number of variables to analyze. It also reduces training and classification times as well as measurement and storage requirements.

To the best of our knowledge, little effort has been dedicated to develop feature selection methods tailored for datasets with both categorical and numerical values. Such heterogeneous data are found in several applications. For instance, in the medical domain, high dimensional continuous feature sets (e.g. gene expression data) are typically considered along with a few clinical features. These features can be continuous (e.g. blood pressure) or categorical (e.g. sex, smoker vs non-smoker). To highlight important variables, a naive approach would transform heterogeneous data into either fully continuous or categorical variables before applying any standard feature selection algorithm. To get a continuous dataset, categorical variables can be encoded as numerical values. The specific choice of such numerical values is however arbitrary. It introduces an artificial order between the feature values and can lead to largely different distance measures between instances [1].

A standard approach relies on a multivariate numerical encoding, such as the disjunctive encoding, to represent categorical variables. For instance, a feature having 3 categories as possible values could be encoded by considering 3 new features instead: $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. However, they need specific approaches, such as group lasso [2], to correctly handle feature selection at the granularity of the original features.

The discretization of continuous features is a common alternative to represent categorical and numerical features in a similar space. Such approach comes at the price of making the selection highly sensitive to the specific discretization [1].

A natural alternative would consider tree ensemble methods such as Random Forests (RF), since they can be grown from both types of variables and these methods perform an embedded selection. RF were however shown to bias the selection towards variables with many values [3]. The cForest method has been introduced to correct this bias [3] but its computational time is drastically increased and becomes prohibitive when dealing with thousands of features.[2]

In this paper we propose two kernel based methods for feature selection. They are conceptually similar to disjunctive encoding while keeping original features throughout the whole selection

* Corresponding author.
E-mail addresses: jerome.paul@uclouvain.be (J. Paul), roberto.dambrosio@uclouvain.be (R. D'Ambrosio), pierre.dupont@uclouvain.be (P. Dupont).

[1] http://www.ucl.ac.be/mlg/.

[2] In each node of each tree of the forest, a conditional independence permutation test needs to be performed to select the best variable instead of a simple Gini evaluation.

http://dx.doi.org/10.1016/j.neucom.2014.12.098
0925-2312/

process. In both approaches, the selection is performed by the Recursive Feature Elimination (RFE) [4] mechanism that iteratively ranks variables according to their importances. We propose to extract those feature importances from two different kernel methods: the Support Vector Machine (SVM) and the Multiple Kernel Learning (MKL), with a dedicated heterogeneous kernel. We use the clinical kernel [5] that handles both kinds of features in classification tasks.

The remainder of this document is organized as follows. Section 2 describes the two proposed methods. Section 3 briefly presents competing approaches we compare to in our experiments. The experimental setting is presented in Section 4. Results are discussed in Section 5. Finally, Section 6 concludes this work.

2. Material and methods

This section presents the different building blocks that compose our two heterogeneous feature selection methods. Recursive Feature Elimination (RFE), the main feature selection mechanism, is presented in Section 2.1. It internally uses a global variable ranking for both continuous and categorical features. This ranking is extracted from two kernel methods (Support Vector Machine and Multiple Kernel Learning) that use a dedicated heterogeneous kernel called the *clinical kernel* (Section 2.2). Section 2.3 details how to obtain a feature ranking from a non-linear SVM. Finally, Section 2.4 sketches Multiple Kernel Learning, which offers an alternative way to rank variables with the clinical kernel.

2.1. Recursive feature elimination

RFE [4] is an embedded backward elimination strategy that iteratively builds a feature ranking by removing the least important features in a classification model at each step. Following [6], a fixed proportion of 20% of features is dropped at each iteration. The benefit of such a fixed proportion is that the actual number of features removed at each step gradually decreases till being rounded to 1, allowing a finer ranking for the most important features. This iterative process is pursued till all variables are ranked. The number of iterations automatically depends on the total number p of features to be ranked while following this strategy. RFE is most commonly used in combination with a linear SVM from which feature weights are extracted. However, it can be used with any classification model from which individual feature importance can be deduced. A general pseudo-code for RFE is given in Algorithm 1.

Algorithm 1. Recursive Feature Elimination.

```
R ← empty ranking
F ← set of all features
while F is not empty do
    train a classifier m using F
    extract variable importances from m
    remove the 20% least important features from F
    put those features on top of R
end
return R
```

2.2. Clinical kernel

The so-called clinical kernel proposed in [5] was shown to outperform a linear kernel for classifying heterogeneous data. It averages univariate subkernels [7] defined for each feature:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{p}\sum_{f=1}^{p} k_f(x_{if}, x_{jf}) \tag{1}$$

$$k_f(a,b) = \begin{cases} I(a=b) & \text{if } f \text{ is categorical} \\ \dfrac{(max_f - min_f) - |a-b|}{max_f - min_f} & \text{if } f \text{ is continuous} \end{cases} \tag{2}$$

where $\mathbf{x}_i$ is a data point in p dimensions, x_{if} is the value of $\mathbf{x}_i$ for feature f, I is the indicator function, a and b are scalars and max_f and min_f are the maximum and minimum values observed for feature f, respectively. One can note that summing kernels simply amounts to concatenating variables in the kernel induced space.

Given two data points, the subkernel values lie between 0, when the feature values are farthest apart, and 1 when they are identical, similar to the Gaussian kernel. The clinical kernel is basically an unweighted average of overlap kernels [8] for categorical features and triangular kernels [9,10] for continuous features. The overlap kernel can also be seen as a rescaled l_1-norm on a disjunctive encoding of the categorical variables. The clinical kernel assumes the same importance to each original variable. We show here the benefit of adapting this kernel for heterogeneous feature selection.

2.3. Feature importance from non-linear Support Vector Machines

The Support Vector Machine (SVM) [11] is a well-known algorithm that is widely used to solve classification problems. It looks for the largest margin hyperplane that distinguishes between samples of different classes. In the case of a linear SVM, one can measure the feature importances by looking at their respective weights in the hyperplane. When dealing with a non-linear SVM, we can instead look at the variation in margin size $1/\|\mathbf{w}\|$. Since the larger the margin, the lower the generalization error (at least in terms of bound), a feature that does not decrease much the margin size is not deemed important for generalization purposes. So, in order to measure feature importances with a non-linear SVM, one can look at the influence on the margin of removing a particular feature [12].

The margin is inversely proportional to

$$W^2(\alpha) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{w}\|^2 \tag{3}$$

where α_i and α_j are the dual variables of a SVM, y_i and y_j the labels of $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, out of n training examples, and k a kernel. Therefore, the importance of a particular feature f can be approximated without re-estimating α by the following formula:

$$J_{SVM}(f) = |W^2(\alpha) - W^2_{(-f)}(\alpha)| \tag{4}$$

$$W^2_{(-f)}(\alpha) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j k(\mathbf{x}_i^{-f}, \mathbf{x}_j^{-f}) \tag{5}$$

where $\mathbf{x}_i^{-f}$ is the ith training example without considering the feature f. In Eq. (5), the α's are kept identical to those in Eq. (3). This is a computationally efficient approximation originally proposed in [12]. The feature importance is thus evaluated with respect to the separating hyperplane in the current feature space and hence the current decision function.

Updating $k(\mathbf{x}_i, \mathbf{x}_j)$ to $k(\mathbf{x}_i^{-f}, \mathbf{x}_j^{-f})$ is pretty efficient and straightforward with the clinical kernel (Section 2.2). There is no need to recompute the sum of all subkernels but one only has to remove k_f (Eq. (2)) and normalize accordingly. Removing one such subkernel is equivalent to removing features in the projected space, which is similar to what is done with a linear kernel.