Feature Selection

Damien François
Université catholique de Louvain, Belgium

INTRODUCTION

In many applications, like function approximation, pattern recognition, time series prediction, and data mining, one has to build a model relating some features describing the data to some response value. Often, the features that are relevant for building the model are not known in advance. Feature selection methods allow removing irrelevant and/or redundant features to only keep the feature subset that are most useful to build a prediction model. The model is simpler and easier to interpret, reducing the risks of overfitting, non-convergence, etc. By contrast with other dimensionality reduction techniques such as principal component analysis or more recent nonlinear projection techniques (Lee & Verleysen 2007), which build a new, smaller set of features, the features that are selected by feature selection methods preserve their initial meaning, potentially bringing extra information about the process being modeled (Guyon 2006).

Recently, the advent of high-dimensional data has raised new challenges for feature selection methods, both from the algorithmic point of view and the conceptual point of view (Liu & Motoda 2007). The problem of feature selection is exponential in nature, and many approximate algorithms are cubic with respect to the initial number of features, which may be intractable when the dimensionality of the data is large. Furthermore, high-dimensional data are often highly redundant, and two distinct subsets of features may have very similar predictive power, which can make it difficult to identify the best subset.

BACKGROUND

Feature selection methods are often categorized as 'filters,' 'wrappers,' or 'embedded' methods. Roughly stated, filters use statistical measures to 'filter out' unneeded features before building the model. Wrappers, by contrast, use the prediction error of the model to select the features. Embedded methods are actually prediction models that propose, as a by-product, a scoring, or even a selection of the features; like for instance decision trees (Breiman, 2001) and LASSO models (Efron, 2004).

This distinction between filters and wrappers, which has historical roots (Kohavi & John, 1997), is less and less relevant now because many new methods are very difficult to label with either name. More generally, all feature selection methods share the same structure; they need (1) a criterion that scores a feature or a set of features according to its (their) predictive power, and (2) a method to find the optimal subset of features according to the chosen criterion. This method comprises an exploration algorithm, which generates new subsets for evaluation, and a stopping criterion to help deciding when to stop the search.

Criteria for scoring a single feature include the well-known correlation, chi-squared measure, and many other statistical criteria. More powerful methods, like mutual information (Battiti, 1994), and the Gamma test (Stefansson *et al*, 1997) (for regression) and the RELIEF algorithm (Kira & Rendell, 1992) (for classification), allow scoring a whole subset of features. In a more wrapper-like approach, the performances of a prediction model can also be used to assess the relevance of a subset of features.

Algorithms for finding the optimal subset, which is a combinatorial problem, can be found in the Artificial Intelligence literature (Russel & Norvig, 2003). In the context of feature selection, greedy algorithms, which select or exclude one feature at a time, are very popular. Their reduced complexity still allows them to find near optimal subsets that are very satisfactory (Aha & Bankert, 1996).

MAIN FOCUS

This section discusses the concepts of relevance and redundancy, which are central to any feature selection method, and propose a step-by-step methodology along with some recommendations for feature selection on real, high-dimensional, and noisy data.

Relevance and Redundancy

What do we mean precisely by "relevant"? See Kohavi & John (1997); Guyon & Elisseeff (2003); Dash & Liu (1997); Blum & Langley (1997) for a total of nearly ten different definitions for the relevance of a feature subset. The definitions vary depending on whether a particular feature is relevant but not unique, etc. Counter-intuitively, a feature can be useful for prediction and at the same time irrelevant for the application. For example, consider a bias term. Conversely, a feature that is relevant for the application can be useless for prediction if the actual prediction model is not able to exploit it.

Is redundancy always evil for prediction? Surprisingly, the answer is no. First, redundant features can be averaged to filter out noise to a certain extent. Second, two correlated features may carry information about the variable to predict, precisely in their difference.

Can a feature be non relevant individually and still useful in conjunction with others? Yes. The typical example is the XOR problem (Y = X1 XOR X2), or the sine function over a large interval (Y = sin (2 π (X1 + X2)). Both features X1 and X2 are needed to predict Y, but each of them is useless alone; knowing X1 perfectly, for instance, does not allow deriving any piece of information about Y. In such case, only multivariate criteria (like mutual information) and exhaustive or randomized search procedures (like genetic algorithms) will provide relevant results.

A Proposed Methodology

Mutual information and Gamma test for feature subset scoring. The mutual information and the Gamma test are two powerful methods for evaluating the predictive power of a subset of features. They can be used even with high-dimensional data, and are theoretically able to detect any nonlinear relationship. The mutual information estimates the loss of entropy of the variable to predict, in the information-theoretical sense, when the features are known, and actually estimates the degree of independence between the variables. The mutual information, associated with a non-parametric test such as the permutation test, makes a powerful tool for excluding irrelevant features (François et al., 2007). The Gamma test produces an estimate of the variance of the noise that a nonlinear prediction model could reach using the given features. It is very efficient when totally irrelevant features have been discarded. Efficient implementations for both methods are available free for download (Kraskov et al, 2004; Stefansson et al, 1997).

Greedy subset space exploration. From the simple ranking (select the K features that have the most individual score), to more complex approaches like genetic algorithms (Yang & Honavar, 1998) or simulated annealing (Brooks et al, 2003), the potentially useful exploration techniques are numerous. Simple ranking is very efficient from a computational point of view, it is however not able to detect that two features are useful together while useless alone. Genetic algorithms, or simulated annealing, are able to find such features, at the cost of a very large computational burden.

Greedy algorithms are often a very suitable option. Greedy algorithms, such as Forward Feature Selection, work incrementally, adding (or removing) one feature at a time, and never questioning the choice of that feature afterwards. These algorithms, although being sub-optimal, often finds feature subsets that are very satisfactory, with acceptable computation times.

A step-by step methodology. Although there exists no ideal procedure that would work in all situations, the following practical recommendations can be formulated.

1. Exploit any domain knowledge to eliminate obviously useless features. Use an expert, either before selecting features to avoid processing obviously useless features. Removing two or three features a priori can make a huge difference for an exponential algorithm, but also for a cubic algorithm! You can also ask the expert after the selection process, to check whether the selected features make sense.
2. Perform simple ranking with the correlation coefficient. It is important to know if there is a strong linear link between features and the response value, because nonlinear models are seldom good at modeling linear mappings, and will certainly not outperform a linear model in such a case.