An Evaluation Of Feature Selection Algorithms In Machine Learning

R Ravi kumar, M Babu Reddy P Praveen

Abstract: In the outline of statistical pattern recognition, feature selection is an vital step aim to dig out the most significant inequitable in order for classification and accumulate it succinctly into a pattern vector of a lower dimensionality. The impetus for applying feature selection is assorted. At first place, features can be expensive to obtain. The cost includes quantity attainment, data preprocessing, storage and transfer, reasons behind computation, etc. Furthermore, high-dimensional problems should more samples for training to accomplish a good overview potential of a classifier (i.e., the curse of dimensionality). Reduced dimensionality of the pattern vector can also help to gain enhanced perceptive of a given problem in applications like medicine or genetic engineering. The main goal of this paper is to design a classifier independent (filter-based) feature selection method that would allow the merit of individual features to be assessed from one-dimensional projections of the data. However, we have encountered several other problems during our research which had to be addressed first.

Index Terms: Classification, Clustering, Data Mining, Feature Selection, Pattern reorganization.

1. INTRODUCTION

MANY tasks in statistical pattern recognition are characterized by high dimensional data which have to be processed and analyzed using statistical tools. A pattern (data sample) is a vector formed generally by many measurements or observations (features) of different physical or other quantities. There are often tens to several hundreds or even thousands of features composing an individual pattern vector. Examples of such data are measurements arising in character, text, and face recognition from digitized images, spam email identification, diagnostics tasks in medicine and genetic engineering, recognition tasks in biology, economics, astronomy, etc. The recognition/classification of a given pattern is characterized by one of the two following tasks. Supervised classification is a problem of establishing decision regions between patterns and assigning an unknown input pattern into one of the predefined classes. In unsupervised classification, classes are learned based on the similarity of patterns. The recognition system operates in two modes – learning (training) from a given set of examples and classification (testing), see Figure 1.1. preprocessing module serves for data normalization, noise removal, segmentation of patterns of interest from background, etc. In the training phase, the feature selection/extraction module finds suitable features which describe the input patterns, and subsequently a classifier is trained to partition the feature space. The feedback allows to optimize the preprocessing and feature selection/extraction strategies with respect to a designed classifier. In the classification phase, the system makes automatic decision about unknown input patterns.Learning from a small sample statistic results in inaccurate parameter estimates and consequently, a poor generalization is achieved on unknown data. The determination of an arbitrarily complicated decision boundary therefore requires a large number of training samples. Moreover, the number of the necessary samples grows exponentially with the feature space dimensionality [14]

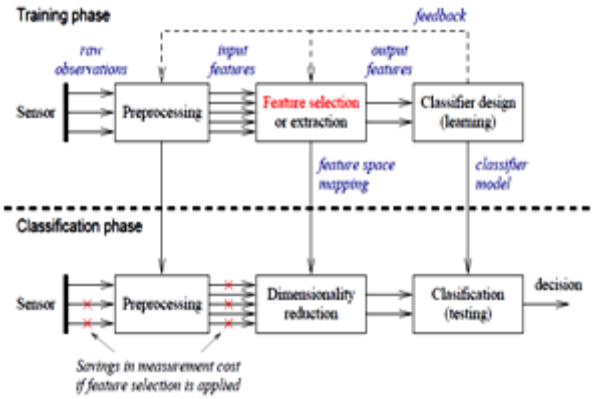

Figure 1.1: A block diagram of a pattern recognition system.

and huge data storage and computation resources are required. This phenomenon is known as the curse of dimensionality [15]. A common solution is to reduce the dimensionality of the input feature space and to employ only those features that are most relevant to the given problem. A fewer features allow to design less complex classifiers with better parameter estimates, and perhaps with improved prediction accuracy. Notice that even dimensionality reduction can become very difficult if the discrepancy between the original dimensionality and the number of training samples is too large. There are two approaches leading to dimensionality reduction. They are referred to as the feature selection and the feature extraction, see [13, 6, 4] for instance. It is important to distinguish between both notions. While feature selection methods output a subset of the original features without any further change, feature extraction algorithms transform the input features into a completely different space.

2 RELATED WORK

Pudil et al. [3] show a historic development of traditional sequential subset search strategies. In 1963, Marill and Green [6] introduced a top-down technique known at present as the Sequential Backward Selection (SBS). It used the divergence [4] as a criterion function. Its bottom-up counterpart, termed as the Sequential Forward Selection (SFS)[10].

- *R. Ravi Kumar, Assistant Professor in the Dept of CSE, S R Engineering College, Warangal, TS & Research Scholar in Krishna University, Machilipatnam, A.P. E-mail: ravikumar.racha@gmail.com*
- *M. Babu Reddy, Assistant Professor, Department of CS, Krishna University, Machilipatnam, Andhra Pradesh . E-mail: m_babureddy@yahoo.com*
- *P Praveen, Associate Professor the Dept of CSE, S R Engineering College, Warangal, TS .India.E-mail:prawin1731@gmail.com*

2.1 Traditional subset search strategies

The so called nesting effect. It means that adding (subtracting) only the locally best (worst) features cannot be corrected in the later stage due to their greedy character. First attempt dealing with nested subsets of features was presented by Michael and Lin [6] . Stearns [9] redeveloped this idea,, into the Plus-l-Minus-r search method, called also (l,r) or LRS algorithm. Values l and r correspond to the number of forward and backward steps. However, this is also a suboptimal method as there is no way how to predict values l and r that would lead to the optimal subset of features. Kittler [4] generalized SBS, SFS, and LRS algorithms so that sets of features could be added or removed. Moreover, Kittler performed a comparative study of so far known traditional search strategies and showed that generalized methods perform better compared to ordinary algorithms but only at the expense of the computational time. Narendra and Fukunaga [6] adopted the Branch and Bound algorithm (BAB) for the feature selection. Jain [9] claims that techniques based on the Branch and Bound strategy are the only optimal methods without performing the exhaustive search. The optimality is guaranteed, however, for the use with monotonous search criteria functions only. Notice that the most commonly used criteria functions (e.g., the prediction accuracy) violate the monotonicity. The computational time of the BAB algorithm is enormous especially for high dimensional problems. Several improvements of the BAB algorithm and some basic information about the feature selection can be found in the Fukunaga's book [12]. A considerable speed up of the BAB algorithm was achieved by Somol et al. [9] in 2004. The principle consists in predicting criterion values based on statistics from the previous features discards. Probably one of the most effective suboptimal sequential search techniques, in terms of speed and optimality, are currently the Floating Search methods introduced by Pudil et al. [7] in 1994. The nesting effect is efficiently counteracted by applying a number of backtracking steps. The idea originates from the LRS algorithm. Nevertheless, the number of forward steps l and backward steps r is controlled dynamically within the search and thus no parameter setting is necessary. Depending on the direction of the search, algorithms are called the Sequential Forward Floating Search (SFFS) and the Sequential Backward Floating Search (SBFS). Floating methods. allow to use non-monotonous criteria functions. Adaptive versions of the Floating Search strategies were proposed by Somol et al. [9] in 1999. Algorithms incorporate generalized search methods into floating search strategies. showed that adaptive algorithms perform even better than floating ones, however, the running time is much higher. The latter type of a search strategy in this series is probably the Oscillating Search algorithm (OS) introduced by Somol and Pudil [9] in 2000. This method is independent of the search direction. In order to improve the criterion value, it explores subsets of features only within a certain interval of the target subset size. An initial subset of features of a certain cardinality is assumed to be given at the beginning of the search. Somol and Pudil claim that the solution is usually better than that found by the floating methods.

3 FILTER METHODS

Feature ranking Feature ranking methods are optimal only under the assumption of normally distributed data and statistically independent features (generally, neither of these are true). A weight is assigned to each feature according to its individual merit (e.g., information content, entropy, relevance) and only the top-ranking candidates are selected. Feature ranking methods are not able to remove redundant or replicated information and do not enforce a good complementarity between features.

Subset search

Subset search strategies are more exact as the criterion function evaluates the quality of subsets of features. The task is usually transformed to a tree exploration problem, where each node represents a different subset of features. The value of the criterion function is used to guide the search for the best subset of features. However, this approach might become computationally very expensive for high dimensional problems, because the size of the explored space corresponds to the number of all possible combinations of features. The criterion function guiding the search for the best features is usually some kind of separability measure between classes. It can be either classifier independent (i.e., filter approach) or classifier specific [4,8] (i.e., wrapper approach or embedded method). The terms filter and wrapper were introduced by John et al. [4] Filters are regarded more as a preprocessing step for a subsequent learning. They often employ only a heuristic approach where the criterion function is not directly linked to the performance of a particular classifier. Instead, the solution relies on intrinsic properties of the training data. Features are usually evaluated based on criteria like probabilistic distance measures, Pearson correlation, entropy, or other information-theoretic measures [13]. Thereby filters provide a general approach to feature selection making the solution suitable for a large family of classifiers. However, it is often hard to define a criterion function which is globally optimized with respect to the expected risk Filters execute quite fast and thus can become useful for high dimensional problems where other methods have no chance due to their computational complexity. Nevertheless, the optimality of the selected features does not necessarily guarantee the best possible performance for a classifier.

Wrappers

Wrappers evaluate feature subsets by estimating the prediction accuracy of a preselected learning algorithm and thereby approximate the expected risk The search strategy uses the prediction accuracy as a criterion function to guide the search for the best subset of features and tries to find features that maximize it. In fact, the learning algorithm acts as a black box which makes wrappers remarkably universal and simple. Features are of course optimized for the preselected learning algorithm and thus may not be optimal for another learning method. Wrappers are brute-force methods and require a massive amount of computations, because a large number of classifiers have to be designed during the search process. A speed up may be achieved when using efficient search strategies. The search becomes nevertheless infeasible with increasing dimensionality especially for computationally intensive learning methods.

Embedded methods

Embedded methods represent a quite recent approach to feature selection. Their idea is to minimize the expected risk directly and to perform feature selection implicitly as a part of a

classifier design. Such a builtin feature selection mechanism can be found, for instance, in algorithms like SVM [8], Adaboost [8], or CART [7]. The selected features are of course tuned again for a specific classifier. The big advantage is much more efficient use the available data, because samples do not have to be divided into smaller training and validation parts. Moreover, the solution is reached much faster than in wrapper approach by avoiding retraining a classifier from scratch for every examined feature subset.

4 COMPARISONS OF FEATURE SELECTION ALGORITHMS

Ground truth regarding the quality of the features is not available in practice. Feature selection algorithms are therefore assessed based on their execution time and on the size and quality of the selected feature subsets. Experimental methodology for the results evaluation is however not standardized in the literature and differs from paper to paper. Moreover, the results are often provided without information about their reliability. It is therefore very difficult to draw any conclusions or to make any comparison between feature selection methods proposed or examined over various research papers.

4.1 Performance evaluation of the selected feature subsets

Feature selection is typically a single purpose task performed in an off-line manner. It has been argued that the execution time is therefore not as critical property as the optimality the solution. While this is true for data of moderate dimensionality, several recent applications (e.g., document classification, genetic engineering) involve several thousands of features. In such cases, the computational requirements of feature selection methods may become very important. The most popular criterion is based on the error rate (or the prediction accuracy) by of a preselected learning algorithm which is designed employing a selected feature subset. The error rate has to be estimated using the available samples, because their probability distribution is unknown in general. A common way is to run ten-fold or leave-one-out cross-validation, or repeated holdout validation with data splits in ratios 1:1, 2:1, or 3:2. Obviously, the achieved results can vary a lot. draws also attention to the fact that some research papers show only a cross-validation error achieved throughout the design (i.e., a value which can be severely optimistically biased) and not the error rate on an independent test data. Many research papers do not describe applied experimental methodology in detail. For example, it is often unclear what data preprocessing was utilized, or how large were training and test data for the feature selection and for a classifier design and for its performance evaluation. In order to compare different feature selection methods, some research papers compare performance of more preselected learning algorithms which are designed using the selected subsets of features. This is however more like a comparison of various classifiers designed on the given data rather than comparing feature selection algorithms.

4.2 Benchmark for Data Set

As UCI data sets suffer from many problems mentioned above, a very clean data repository called IDA [11] has been created. IDA repository is partly based on the UCI but all data are normalized and contain only two-class problems with predefined training and test data. The database also provides some simulation baseline results. Nevertheless, IDA is not commonly used for comparing feature selection algorithms. An alternative to the real-world problems are synthetic (or “toy”) data. The main motivation for examining algorithms employing synthetic data is known parameters of the generating probability distribution. Properties of analyzed feature selection techniques can be therefore investigated much more easily. We can tell, for instance, what results are expected, what type of the decision rule is the best for the data, which features should be selected, what is the theoretical performance of the selected features, etc. Also the number of samples can be controlled easily. Hence examined algorithms can be analyzed with respect to the sample size. Nevertheless, artificial data use only simplified models and thus can simulate only a very limited number of situations which may occur in reality. There is also a big danger that a proposed feature selection technique can be designed to solve a given artificial problem, i.e., knowledge of the data is embedded in the algorithm, which is not always desirable not work on the real-world data. The sample size of many real-world problems is typically much higher than the dimensionality. Using at least ten times as many training samples per class as the number of features is considered enough [9] to achieve good generalization properties of learning algorithms. Naturally, the ratio should be higher for more complex classifiers to avoid over-fitting.

5 CONCLUSION

In this paper, we organized the state-of-the-art in feature selection in order to provide a quick and compact overview for the reader. We discussed basic search strategies applied in feature selection (i.e., feature ranking and subset search algorithms) as well as methods which are independent or dependent on a classifier (i.e., filters, wrappers, and embedded methods). Next, the evaluation methodology applied in feature selection was briefly described including the most common benchmark data. Finally, short description and internet links to software toolboxes widely employed in statistical pattern recognition were provided.

REFERENCES

[1] J. Bi, K. P. Bennett, M. Embrechts, C. M. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. Journal of Machine Learning Research, 3:1229–1243, March 2003.

[2] C. M. Bishop. Neural Networks for Pattern Recognition. Oxford University Press, 1996.

[3] P. S. Bradley and O. L. Mangasarian. Feature selection via concave minimization and support vector machines. In Proceedings of the 15th International Conference on Machine Learning, pages 82–90, San Francisco, CA, USA, 1998. Morgan Kaufmann.

[4] R. Ravi Kumar, M. Babu Reddy and P. Praveen, "A review of feature subset selection on unsupervised learning," 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai,2017,pp.163-167.doi: 10.1109/AEEICB.2017.7972404.

[5] P. Praveen, C. J. Babu and B. Rama, "Big data environment for geospatial data analysis," 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 2016, pp. 1-6.doi: 10.1109/CESYS.2016.7889816

[6] R. P. W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, and D. M. J. Tax. PRTools4, A Matlab Toolbox for Pattern Recognition. Delft University of Technology, 2004.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification. John Wiley & Sons, New York, USA, 2001.
[8] F. Fleuret. Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research, 5:1531–1555, November 2004.
[9] V.FrancandV.Hlav´a˘c. Statistical pattern recognition toolbox for Matlab. Research Report CTU–CMP–2004–08, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, June 2004.
[10] C. Manning, P. Raghavan, and H. Schu¨tze, Introduction to Information Retrieval. Cambridge University Press, 2008
[11] R. Shapire and Y. Freund, Boosting: Foundations and Algorithms. The MIT Press, 2012.
[12] T. Ru¨ckstieβ, C. Osendorfer, and P. van der Smagt, “Minimizing data consumption with sequential online feature selection,” International Journal of Machine Learning and Cybernetics, vol. 4, no. 3, pp. 235–243, 2013.
[13] P.Praveen, B.Rama, “An Efficient Smart Search Using R Tree on Spatial Data”, Journal of Advanced Research in Dynamical and Control Systems, Issue 4,ISSN:1943-023x
[14] A. Farahat, A. Elgohary, A. Ghodsi, and M. Kamel, “Greedy column subset selection for large-scale data sets,” Knowledge and Information Systems, 2014.
[15] R. Polikar, “Ensemble based systems in decision making,” IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, 2006.