LIA: A Label-Independent Algorithm for Feature Selection for Supervised Learning

Gail Gilboa Freedman[1], Alon Patelsky[1], and Tal Sheldon[1]

The Interdisciplinary Center Herzliya, Israel gail.gilboa@idc.ac.il
http://portal.idc.ac.il/faculty/en/pages/profile.aspx?username=ggilboa

Abstract. The current study considers an unconventional framework of unsupervised feature selection for supervised learning. We provide a new unsupervised algorithm, and call it *LIA*, for Label-Independent Algorithm. LIA combines information-theory and network-science techniques. In an empirical study, we compared LIA with a standard supervised algorithm (MRMR), that is similar to LIA for minimizing the redundancy among the selected features, but different as it has the advantage of being able to use the labels of the instances in the input data set for maximizing the relevance of the selected features. We used cross-validation to evaluate the effectiveness of selected features for generating different well-known classifiers for a variety of publicly available data sets. The results demonstrate that the classification accuracy of our proposed algorithm is very close to or in some cases better than that of MRMR. Thus, LIA has potential to be useful for development of a feature selection tool that supports a wide variety of applications where dimension reduction is needed and the instances in the data-set are unlabeled.

Keywords: Machine Learning, feature selection, symmetric uncertainty, supervised learning, classifiers, greedy algorithm, community detection, Louvain algorithm, performance evaluation

1 Introduction

Feature selection is the task of identifying the most informative features in any body of data. Traditionally, feature selection algorithms for supervised learning are designed to work with known classification problems, selecting the subset of features that promises to provide the most accurate classification for a given problem. Yet rapid advances in data storage and sharing technologies now enable the collection and warehousing of vast quantities of high-dimensional data, often before it is known precisely how those data will be used. For instance, patient data collected by hospitals (physical measurements, current diagnosis, clinical history, etc.) are routinely supplied to researchers for various applications, including generating predictive models aimed at (for example) optimizing medical procedures or personalizing treatments. Maintaining such databases may be costly, time-consuming, and reliant on compliance by medical staff. Given the inefficiencies of storing redundant data, and of re-selecting information for each

potential application, hospitals would like to minimize the number of features in the data they store while keeping it informative. This raises an interesting seldom-studied question of whether unsupervised methods for feature selection achieve performance comparable to a supervised feature selection methods.

It is natural to assume that such an algorithm – i.e., one specifying a different feature subset for each classification problem – would perform better than an algorithm which takes a unsupervised approach and selects the same feature subset for all problems. The present paper explores the extent of this advantage.

Our novel approach to feature selection assumes that the selection is unsupervised, but the application is in supervised learning. This framework is in accordance with many real-life situations, including the medical data example adduced above. As that example demonstrates, the challenge of reducing the dimensions of stored data often arises in cases where the classification problem is still unknown, but the unlabeled data set contains identifiable redundancies. We hypothesize that in many such cases, analysis of internal relations between predictive features (e.g., patients' "blood-pressure" and "weight" in our hospital example) may suffice for selecting a feature subset that would be efficient when integrated into a variety of classifiers.

We suggest an algorithm that analyzes the mutual relations between predictive features and exploits these relations to identify redundancies in the data set. We call our algorithm *LIA*, for Label-Independent Algorithm. LIA has two steps: partition and selection. In the first step, partition, the algorithm represents the data set as a large network, where each node or vertex stands for a feature. Pairs of nodes are connected by edges with weights that represent the mutual dependency between the corresponding features. Then, LIA detects communities in this feature network, using the familiar Louvain's algorithm [9]. In the second step, selection, the algorithm performs a greedy process that selects features based on the results of the partition. Specifically, it starts with a set consisting of one feature from each community, and adds new features in each iteration so as to minimize the mutual information between new features and the features selected thus far.

LIA is an unsupervised method and therefor is used prior to knowing which classifier uses the selected features. The effect of this scenario is double-sided. On the one hand, it improves many aspects of data maintenance, and avoids the need for selecting a different data set for each classification purpose. On the other hand, it reduces users' ability to identify the subset of features that is most relevant to any specific classification problem. Which effect is more dominant? The answer, of course, depends on the costs of each (e.g., the cost of storing the data vs. the cost of false classification). However, to begin with, it is interesting to examine the influence of using an unlabeled data set regardless of these costs. For this purpose, we conducted an empirical study, comparing LIA with the standard label-dependent feature selection algorithm, MRMR.

Our main finding is that LIA not only reduces the costs of handling high-dimensional data (our primary motivation for presenting this algorithm), but remains competitive in terms of the accuracy of classifiers that use its output

features. This result is worth noting, because label-dependent algorithms are based on knowledge about the relevance of different features to different classes, and therefore should be expected to perform significantly better than LIA. The results point to the huge potential of using LIA, or other unsupervised algorithms, in cases where it is costly to maintain large data sets that are not yet specified to a certain classification problem.

2 Literature review

The feature selection problem has attracted substantial attention in the computer science literature (see [13] and [32] for reviews). The problem is often formalized as an initial step in building machine-learning models, in both supervised and unsupervised learning (see reviews in [15,**?**,7,**?**] and [18], respectively). Solutions proposed for feature selection have proven effective for many real-world applications [25] where stored data is potentially highly redundant, for example in the realms of text categorization [46], network intrusion detection [42], face recognition [45] and gene expression [17].

Feature selection algorithms for supervised learning aim to chose the subset of features that is optimal for building an effective classifier. By classifiers, we refer to classification models such as those widely used in machine learning [28], whose purpose is to predict a class value from particular predictive features. Thus, theses algorithms aim to minimize redundancy while maximizing the relevance of the selected features [25,**?**,33], all in a time-efficient manner. These algorithms usually fall into one of four categories: filter, wrapper, embedded, and hybrid (see [43] for a review and [22] for a comparison of these categories). The proposed LIA falls into the filter category as it does not employ a classifier [12,**?**], but it is does not employ the classification problem itself.

Our motivation for studying the efficiency of an unsupervised learning algorithm in the context of supervised learning lies at the challenges of data maintenance in today's age of modern commercial informatics and Big Data. It is very often that feature selection must take place already during the pre-processing phase [14] to enable effective ETL (Extract, Transform, and Load) processes. However, the data at this phase are often unlabeled, and therefore supervised feature selection algorithms cannot be applied.

Notably, our proposed algorithm differs from "class-independent feature selection" algorithms, so-called because they select the same feature subsets for different classes. While they select the same subset for each label-value, these algorithms are not label-independent like ours. For example, the RELIEF algorithm [26] and its multi-class extension ReliefF [27] select features in order to separate instances from different classes.

The proposed LIA algorithm follows the information-based approach prevalent in the feature selection literature. Many real-world data sets include pair-wise dependencies between features [20,**?**], and it is common to quantify these mutual dependencies using information measures (see [31] for a review). Prominent feature selection algorithms that employ the mutual information measure include

MIFS [8], MIFS-U [29], MRMR [37], TMI-FX [36], and MIFS-ND [23]. LIA uses *symmetric uncertainty* (see [41]), which is a normalization of the mutual information measure to the *entropy* (see [39] for definition) of the features. We selected this measure because it is well-established in the feature selection literature and has proven to be effective [10,20,**?**,41].

LIA's employs network analysis and represents the features in the input data set as nodes in a network. A similar approach has been shown to be effective for feature selection in previous work [16] [41]. However, in contrast with previous applications of clustering analysis, LIA quantify feature–feature relations in a manner that is independent of their relevance to any given class. LIA partitions this network, using Louvain's algorithm for partitioning this network [9], which is based on modularity optimization. The maximization of modularity is an NP-complete problem [11], but Louvain's algorithm achieves good partitions reasonably quickly by taking a heuristic greedy approach. Some eminent examples of other community detection algorithms include Newman-Girvan [21], InfoMap [40], stochastic block models [6], and label propagation algorithms [34]. The curious reader is referred to the comparative analyses of community detection algorithms in [35] and [30].

LIA follows a greedy approach that is similar to the greedy approach used in other feature selection algorithms like MRMR [37] or MIFS-ND [23], but is also different for having criterion that uses only feature–feature computations.

Finally, the concept of unsupervised feature selection has been widely addressed in the literature, but mostly in the context of unsupervised learning. In this context, features are selected during the pre-processing phase as part of the construction of *clustering models*, which are widely used in machine learning (see [19] and [24] for reviews). It is common in the balk of literature on unsupervised feature selection, to evaluate algorithms by comparing their performance when employed in different unsupervised models (see [19] and [24] for reviews). Moreover, unsupervised algorithms often considers these performances as part of their process (for examples see [18]). It is not a common approach to compare an unsupervised algorithm with a supervised one ([44]). The current study takes this uncommon approach.

3 Formalizing the Problem

In many real-world situations there are advantages to reducing the dimensions of collected data, at an early stage prior to labeling the data, even to the point of compromising on the accuracy of the classifiers that might use these data in the future (see the hospital example described in the introduction). Motivated by theses situations, we formulate the problem as follows:

The unsupervised feature selection problem for supervised learning: Given a data set D of dimension n with a feature set $F = \{f_1, f_2, \ldots, f_n\}$, and a fixed integer K, what is the optimal subset of features $S = \{s_1, s_2, \ldots, s_K\}$ that are informative but non-redundant?

4 The Label-Independent Algorithm (LIA)

The proposed LIA uses a combination of information theory and network science techniques. Inspired by the principle of identifying features that demonstrate minimal redundancy, LIA computes the feature–feature *symmetric uncertainty* measure to quantify feature–feature similarity. The symmetric uncertainty of two random variables measures the degree to which knowing the value of either random variable reduces uncertainty regarding the value of the other [38]. It is derived by normalizing the *information gain* to the *entropies* of the random variables (see [39] for definitions of these measures). Normalization induces values in the range [0,1] and assures symmetry. A value of zero means that the two variables are independent. Thus, higher (vs. lower) values mean that knowing the value of either variable is more (vs. less) useful for predicting the value of the other. For mormal definition of the symmetric uncertainty of two features X, Y see [41]. LIA has two steps, described next.

4.1 Step 1: Network-Based Partition

Given a data set D of dimension n with a feature set $F = \{f_1, f_2, , f_n\}$, we represent it by a network (complete graph) $G = (V, E)$, in which the vertices (nodes) $V = \{f_1, f_2, , f_n\}$ represent the features and the edges are $E = \{(f_i, f_j)|i, j \in [1, n]\}$. In an adjacency matrix A, all the edges are weighted to reflect how informative its vertices (i.e., pairs of features) are in relation to one another, such that $A_{i,j}$ is the symmetric uncertainty between the i^{th} and j^{th} feature. Then, we partition the network G into m parts or communities $P = \{P_1, P_2, , P_m\}$, applying the Louvain algorithm [9]. Each community includes a subset of features such that the features within any given community are more informative toward one another than toward features in any other community.

4.2 Step 2: Greedy Selection

In the second step, we use the feature–feature symmetric uncertainty values and the partition P (all computed in step 1) to select a subset of K features. We first calculate the *internal average symmetric uncertainty (IASU)* for each feature f_i. We compute this as the average of the symmetric uncertainty values between each feature and the other features in its community. We denote the set of selected features by S, and initialize it by inserting the feature from each community with the highest $IASU$ value. Next, for each non-selected feature, we calculate the *average symmetric uncertainty (ASU)* as the average of the symmetric uncertainty values between a given feature and the features in S. We then add $K - 1$ features to S, one by one, in a greedy manner. In each iteration, we add the feature with the minimal ASU to S, and update the ASU values of all non-selected features accordingly. The process ends when S includes K features.

Algorithm 1 LIA

```
0: Initialize G=NULL
0: for (f_i, f_j) in F do
0:     Add f_i and/or f_j to the set of the vertices V(G)
0:     Add (f_i, f_j) to the set of the edges E(G)
0:     A_{i,j} = SU(f_i, f_j) as the weight of the edge
0: end for
0: P = Louvain (G)
   //Step2: Greedy selection
0: for P_i in P do
0:     for f_j in P_i do
0:         IASU_j = Average of SU(f_j, f_l) for f_l ∈ P_i
0:     end for
0:     S = ∪s_i, where s_i = feature with highest IASU in P_i
0: end for
0: if |S| < K then
0:     F' = F\S
0:     for f_j in F' do
0:         ASU_j = Average of SU(f_j, f_l) for f_l ∈ S
0:     end for
0:     for k = |S| + 1 to K1 do
0:         s_k = feature with highest ASU
0:         Transform s_k from F' to S
0:         if k < K then
0:             Update ASU
0:         end if
0:     end for
0: end if
0: return S =0
```

4.3 Time Complexity Analysis

The time complexity of LIA depends on the number of features in the input dataset n. The most time-consuming stage is the first step of the algorithm. This includes the initial $O(n^2)$ computations of all the pair-wise symmetric uncertainty values, the construction of a feature network with time complexity $O(n^2)$, and Louvain's community detection, which is also $O(n^2)$. In the second step, LIA performs a greedy process with $O(n)$ computations in each iteration (for updating the average symmetric uncertainty of each non-selected feature with the features selected thus far). The number of iterations is lower than k, which is lower than n. In total the time complexity of the greedy process is $O(kn)$, which is bounded by $O(n^2)$. The overall time complexity of LIA is $O(n^2)$.

5 Empirical Study

Simulations were carried out on a workstation with a 2.3 GHz Intel Xeon E5 2686 v4 processor (Ubuntu 18.04.1 bionic). The algorithm was implemented using

Table 1. Data sets examined.

Domain	Data set	Instances	Features
Life Science	iris	150	4
Intrusion	wine	178	13
Social Science	congress voting	435	16
Life Science	spect	267	22
Security	phishing websites	11055	30
Games	connect-4	67557	42
Social Science	coil2000	9821	85
Biology	lung	73	325

Python 3.6.6 software, with packages from the data analysis library *Panda* [3] and with the network analysis packages *Networkx* [2] and *Community* [4].

We used eight data sets varying in size (number of instances) and dimension (number of features) in representative application domains, including biology, computer security and games. The data sets are described in Table 1.

5.1 Procedure

We evaluate the performance of LIA by comparing its results to those achieved using MRMR [37], a standard label-dependent feature selection algorithm that is similar to LIA in being both information-based and greedy. The implementation of MRMR was taken from the publicly available resource (see [1]). We use four common classifiers, namely the decision tree, K-nearest neighbor (KNN), random forest, and support vector machine (SVM)), all implemented in scikit-learn Python packages (see [5]). We compare the performance of the two algorithms, LIA and MRMR, using 10-fold cross validation and comparing their accuracy and F-score measures for all classifiers and data sets.

6 Results

In this section we present the results of our empirical study comparing LIA with MRMR for a variety of classification problems and classifiers. The study's main result is that in most cases LIA's accuracy is similar to that of MRMR. This is a non-intuitive result, considering that LIA ignores the labels in the input data set, while MRMR strongly relies on analyzing the relationships between the features and labels.

6.1 Performance Evaluation for Different Iterations of LIA

We first evaluate the efficiency of LIA along its iterations. It was made for each of the 8 datasets in our study to verify that LIA is competitive with a standard label-dependent algorithm for a variety of data sets and classifiers. The following graph demonstrates an example of such result.

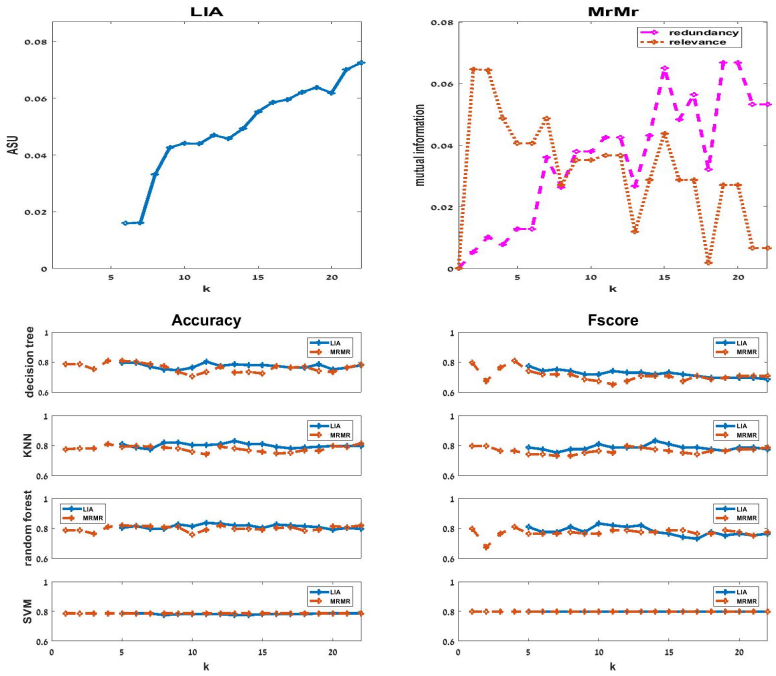

Fig. 1. Data set: SPECT.

6.2 Performance Evaluation for Specific Number K

The table in figure 2 shows the performance measures (accuracy and F-score) under a predefined stopping rule such that the data set includes $log(m)$ features (where m is the number of features in the original data set). Performance measures are shown for LIA in comparison with MRMR when it selects the same number of features, and when all features are selected.

As can be seen in the table, LIA's performance is similar to that of MRMR even when a stopping condition is predefined. This result leads to the practical recommendation that LIA may be used for dimension reduction of non-labeled data to a specific size that does not need to be chosen by the data collector.

7 Conclusions and Future Work

The current study sheds light on situations where the available data is still unlabeled, meaning that the benefits of dimension reduction for data maintenance are achievable only with a feature selection algorithm that is unsupervised. The performance evaluation of our proposed algorithm shows that our unsupervised approach is attractive in today's big data era, when data collectors must sift

		Performance measures with: K features selected by LIA K features selected by MRMR All features			
DT	**INPUT K**	**DT**	**KNN**	**RF**	**SVM**
IRIS	**2**	0.91(0.88) 0.65(0.72) 0.91(0.96)	0.95(0.96) 0.81(0.84) 0.97(0.98)	0.93(0.94) 0.73(0.82) 0.95(0.96)	0.95(0.98) 0.79(0.80) 0.97(0.98)
WINE	**4**	0.81(0.88) 0.86(0.70) 0.92(0.92)	0.64(0.56) 0.65(0.68) 0.67(0.66)	0.88(0.83) 0.89(0.71) 0.98(0.98)	0.68(0.61) 0.46(0.38) 0.44(0.44)
CONGRESS VOTING	**5**	0.94(0.94) 0.95(0.94) 0.94(0.93)	0.94(0.91) 0.96(0.93) 0.91(0.91)	0.93(0.94) 0.95(0.94) 0.93(0.93)	0.96(0.96) 0.96(0.96) 0.94(0.94)
SPECT	**5**	0.79(0.77) 0.80(0.74) 0.77(0.70)	0.80(0.78) 0.79(0.74) 0.79(0.77)	0.80(0.80) 0.81(0.76) 0.79(0.76)	0.78(0.79) 0.78(0.79) 0.89(0.79)
PHISHING WEBSITES	**5**	0.88(0.88) 0.61(0.61) 0.95(0.94)	0.88(0.87) 0.49(0.59) 0.93(0.91)	0.88(0.88) 0.61(0.61) 0.96(0.95)	0.88(0.88) 0.61(0.60) 0.94(0.93)
CONNECT4	**8**	0.65(0.64) 0.67(0.66) 0.69(0.69)	0.57(0.49) 0.59(0.61) 0.71(0.68)	0.65(0.64) 0.67(0.66) 0.78(0.76)	0.65(0.65) 0.66(0.66) 0.67(0.67)
COIL2000	**10**	0.93(0.92) 0.93(0.92) 0.89(0.88)	0.93(0.92) 0.93(0.93) 0.93(0.92)	0.93(0.93) 0.93(0.93) 0.92(0.92)	0.93(0.93) 0.93(0.93) 0.93(0.93)
LUNG	**9**	0.52(0.52) 0.54(0.28) 0.59(0.52)	0.56(0.44) 0.60(0.36) 0.69(0.44)	0.60(0.44) 0.58(0.44) 0.65(0.56)	0.58(0.48) 0.50(0.48) 0.54(0.48)

Fig. 2. Performance summary. We evaluate the performance of LIA when using a predefined stopping condition that significantly reduces the dimensions of the data set. Specifically, the number of selected features (k) is either the *log* of the number of features in the original data set, or the number of communities identified by LIA. The first line in each cell in the table demonstrates the accuracy (and f-score) using these K features, for a specific combination of data-set and classifier. Efficiency using features selected by LIA (first line in each cell) is compared with efficiency using features selected by MRMR (second line), and with efficiency using the complete set of all features (third line) as a benchmark.

through massive quantities of data features and decide which to keep and share – often before the (possibly many) applications of the data are known.

As the main result of this study, we show that our label-independent algorithm (LIA) is competitive with a standard label-dependent algorithm (MRMR) in terms of the accuracy of classifiers that use their outputs. This is despite the fact that LIA discounts the relevance of the features chosen to the (eventual) classification problem.

We recognize the importance of time efficiency in feature selection processes, and we are aware of the advantage of algorithms that do not need to perform the time-consuming analysis of the mutual relations between pairs of features required by LIA. However, in the wider context of big data, where dimension reduction is impossible with label-dependent algorithms, LIA induces efficiency.

Considering the variety of input data sets, classification problems, and classifiers that exist, the unsupervised approach taken in this article will not be

appropriate for all settings. Indeed, in some cases the performance of LIA may be poor in comparison with label-dependent algorithms, either demonstrating low accuracy or requiring more features to reach an optimal value. However, based on the variety of data sets, properties, and classifiers chosen for our empirical study, we believe that any such poor performance will likely arise from specific characteristics of the input data set that do not exist in the examples chosen for this study. More precisely, we believe that poor performance of LIA may be expected in rare cases where the input data sets are characterized by two properties: first, where the features are relatively independent of one another, such that there is no clear underlying pattern of relations between them; and second, where the variance of mutual relations is low. Together, these properties generate ambiguity in patterns of feature community.

In another study, we are also applying the techniques developed in the current paper to the feature selection problem for **un**supervised learning. Specifically, we apply the information-based partition of the features to the problem of feature selection for clustering models.

References

1. http://home.penglab.com/proj/mrmr/
2. https://networkx.github.io/
3. https://pypi.org/project/pandas/
4. https://pypi.org/project/python-louvain/
5. https://scikit-learn.org/stable/
6. Abbe, E., Sandon, C.: Community detection in general stochastic block models: Fundamental limits and efficient recovery algorithms. In: Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS). pp. 670 – 688 (2015)
7. Arauzo-Azofra, A., Aznarte, J.L., Bentez, J.M.: Empirical study of feature selection methods based on individual feature evaluation for classification problems. Expert Systems with Applications **38**(7), 8170 – 8177 (2011)
8. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks **5**(4), 537 – 550 (1994)
9. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment **10** (2008)
10. Blum, A.L., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence **97** (1997)
11. Brandes, U., Delling, D., Gaertler, M., Grke, R., H.M., Nikoloski, Z., Wagner, D.: Maximizing modularity is hard (2006)
12. Cadenas, J.M., Garrido, M.C., Martnez, R.: Feature subset selection filterwrapper based on low quality data. Expert Systems with Applications **40**(16), 6241 – 6252 (2013)
13. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Computers Electrical Engineering **40**(1), 16 – 28 (2014)
14. Chen, M., Mao, S., Liu, Y.: Big data: A survey. Mobile Networks and Applications **19**(2), 171 – 209 (2014)

15. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis **1**(3), 131 – 156 (1997)
16. Dhillon, I.S., Mallela, S., Kumar, R.: A divisive information theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research **3**, 1265 – 1287 (2003)
17. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology **3**(2), 185 – 205 (2005)
18. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. Journal of Machine Learning Research **4**, 845 – 889 (2004)
19. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis. Wiley, New York, isbn 9780340761199 edn. (2001)
20. Gaidhani, J.P., Natikar, S.B.: Implementation on feature subset selection using symmetric uncertainty measure. International Journal of Scientific and Engineering Research **5**(7), 1396 – 1399 (2014)
21. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences = **99**(12), 7821 – 7826 (2002)
22. Hall, M.A., Smith, L.A.: Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In: Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference. pp. 235 – 239 (1999)
23. Hoque, N., Bhattacharyya, D.K., Kalita, J.K.: Mifs-nd: A mutual information-based feature selection method. Expert Systems with Applications **41**(14), 6371 – 6385 (2014)
24. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, NJ, isbn 978-0130222787 edn. (1988)
25. Jovic, A., Brki, K., Bogunovi, N.: A review of feature selection methods with applications. In: Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 1200 – 1205 (2015)
26. Kira, K., Rendell, L.A.: The feature selection problem: Traditional methods and a new algorithm. In: Proceedings of the 10th National Conference on Artificial Intelligence. pp. 129 – 134. AAAI (1992)
27. Kononenko, I.: Estimating attributes: Analysis and extensions of relief. In: Bergadano, F., de Readt, L. (eds.) Proceedings of the 7th European Conference on Machine Learning (ECML 94). pp. 171 – 182. Springer-Verlag, Berlin (1994)
28. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: A review of classification techniques. In: Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering. pp. 3 – 24. IOS Press, Amsterdam, The Netherlands (2007)
29. Kwak, N., Choi, C.H.: Input feature selection for classification problems. IEEE Transactions on Neural Networks **13**(1), 143 – 159 (2002)
30. Lancichinetti, A., Fortunato, S.: Community detection algorithms: A comparative analysis. Physical Review E **80**(5), 056117 (2009)
31. Lin, L.: Divergence measures based on the shannon entropy. IEEE Transactions on Information theory **37**(1), 145 – 151 (1991)
32. Liu, H., Motoda, H.e.: Computational methods of feature selection. CRC Press (2007)
33. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering **17**(4), 491 – 502 (2005)

34. Liu, X., Murata, T.: Advanced modularity-specialized label propagation algorithm for detecting communities in networks. Physica A: Statistical Mechanics and its Applications **389**(7), 1493 – 1500 (2010)
35. Orman, G.K., Labatut, V., Cherifi, H.: Qualitative comparison of community detection algorithms. In: Proceedings of the International Conference on Digital Information and Communication Technology and its Applications. pp. 265 – 279. Springer, Berlin, Heidelberg (2011)
36. Oveisi, F., Oveisi, S., Efranian, A., Patras, I.: Tree-structured feature extraction using mutual information. IEEE Transactions on Neural Networks and Learning Systems **23**, 127 – 137 (2012)
37. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(8), 1226–1238 (2005)
38. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C. Cambridge University Press (1988)
39. Renyi, A.: On measures of entropy and information. In: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics. pp. 547 –561. University of California Press https://projecteuclid.org/euclid.bsmsp/1200512181, Berkeley, CA (1961)
40. Rosvall2007: Maps of information flow reveal community structure in complex networks. Proceedings of the National Academy of Sciences = **105**(4), 1118 – 1123 (2007)
41. Song, Q., Ni, J., Wang, G.: A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Transactions on Knowledge and Data Engineering **25**(1), 1 – 14 (2013)
42. Stein, G., Chen, B., Wu, A.S., Hua, K.A.: Decision tree classifier for network intrusion detection with ga-based feature selection. In: Proceedings of the 43rd Annual Southeast Regional Conference, Volume 2 (ACMSE 43). pp. 136 – 141. ACM (2005)
43. Wang, L., Zhou, N., Chu, F.: A general wrapper approach to selection of class-dependent features. IEEE Transactions on Neural Networks **19**(7), 1267 – 1278 (2008)
44. Wei, X., Xie, S., Cao, B., Philip, S.: Rethinking unsupervised feature selection: From pseudo labels to pseudo must-links. In: Springer, C. (ed.) Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 272 – 287 (2017)
45. Yang, A.Y., Wright, J., Ma, Y., Sastry, S.S.: Feature selection in face recognition: A sparse representation perspective. Tech. rep., EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-2007-99 (2007)
46. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proceedings of the 14th International Conference on Machine Learning (ICML 97). pp. 412 – 420. Morgan Kaufmann, San Francisco, CA (1997)