Feature Selection using Multi-agent Reinforcement Learning

Harshit Jain

25 April 2019

1 Abstract

Feature selection is very important when it comes to solving machine learning problem. Present data-sets are inundated with many features, few of them are important for the problem, others unimportant. One of the major challenge in solving a machine learning problem is to find the features which can significantly enhance the classification performance or the accuracy of prediction. My experimental study involves breast cancer dataset and I aim to formulate the feature selection as a Multi-agent [2] coordination problem and perform feature selection using multi-agent reinforcement learning. I approach the feature selection problem by assigning one agent to each feature. The agent finds if the feature should be part of the model or not.

2 Introduction

Often we come across data-sets which has thousands of features available. However, most of them do not contribute significantly to the classification or offers less predictive power. Also we would always want the There are many traditional approaches for feature selection like Filters, Wrappers and Hybrid(Filter+Wrapper).

Filters use statistical techniques to find the most relevant feature. We can also say that the filters are the pre-processing step which is not dependent upon the machine learning algorithm. Common methods include variable ranking techniques. Wrappers on the other hand solely using machine learning algorithms to find the best feature selection. This has a disadvantage that we have to try all the combinations of features before we could find the best subset of features. A hybrid approach on the other hand combines both the approaches together to bring in the best of both.

My experimental study uses breast cancer dataset. It is a dataset with large number of irrelevant features. We aim to find the relevant set of features which can contribute to the best model. The following approach is wrapper method to find the subset of features. The idea is to assign a learning agent to each feature.

Each agent controls a single feature and learns whether it will be included in or excluded from the final feature subset.

3 Related Work

Filters [6] primarily uses statistical methods to determine the best set of features. Mutual information and Correlation[3] are two of them. Mutual information [6] measures a relationship between two random variables that are sampled simultaneously. In particular, it measures how much information is communicated, on average, in one random variable about another. The mutual information between two variables is 0 if and only if the two variables are statistically independent. The formal definition of the mutual information of two random variables X and Y, whose joint distribution is defined by P(X,Y) is given by

$$\sum_{x \in X} \sum_{y \in Y} P(x,y) log \frac{P(x,y)}{P(x)P(y)}$$

In this definition, P(X) and P(Y) are the marginal distributions of X and Y obtained through the marginalization process.

Similarly, in Pearson Correlation [4] several quantitative variables are measured on each member of a sample. If we consider a pair of such variables, it is frequently of interest to establish if there is a relationship between the two; i.e. to see if they are correlated. Pearson's correlation coefficient is a statistical measure of the strength of a linear relationship between paired data. In a sample it is denoted by r and is by design constrained as follows

$$-1 \leq r \leq 1$$

Furthermore: Positive values denote positive linear correlation; Negative values denote negative linear correlation; A value of 0 denotes no linear correlation; The closer the value is to 1 or -1, the stronger the linear correlation. While implementing a uni-variate filter we consider each of the feature and the target class and compute the Pearson correlation. Based on the square of this score the top-ranked features are selected; the square is considered so negatively correlated features are included. Example of this approach is uni-variate correlation-based feature selection (uCFS)[6]. When we select multiple features we call it as multivariate filter. A popular example of this is the minimal redundancy maximal-relevance (mRMR) method. mRMR searches for a subset S F that maximises relevance by maximising the mutual information between each individual feature and the class C, and at the same time minimises redundancy by minimising the mutual information between any two feature spaces within the subset. Another example of this category is the multivariate (mCFS).

Chi-square test measures dependence between stochastic variables, so using this test essentially weeds or removes out the features that are the most likely to be independent of class and therefore irrelevant for classification. It shares similarities with coefficient of determination, R. The chi-square score is given by :

$$\chi^2 = \frac{(Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency}$$

where, Observed frequency is number of observations of class and Expected frequency is the number of expected observations of class if there were no relationship between the features and the target.

The F-statistic on the other hand is a ratio of two variances. Variances are a measure of dispersion. In other words how far is the data are scattered from the mean. Larger values of f-statistic imply greater dispersion.

$$F = \frac{(RSS_{NH} - RSS_{AH})/(df_{NH} - df_{AH})}{RSS_{AH}/df_{AH}}$$

Wrappers[6] uses the predictive algorithm and algorithm's performance as the objective function in order to evaluate feature subsets. Such a problem is indeed a NP-hard problem as we have to evaluate all the 2^n possible feature subsets. For the same reason wrappers are computationally intensive and slow but can also outperform filters because they take into account the feedback from the classifier. Wrapper methods can be classified into Sequential Selection Algorithms and Heuristic Search Algorithms. The sequential selection algorithms start with an empty set and we go on adding features until the maximum objective function is obtained. To speed up the selection, a criteria is chosen which incrementally increases the objective function until the maximum is reached with the minimum number of features. The heuristic search algorithms evaluate different subsets to optimize the objective function. Different subsets are generated either by searching around in a search space or by generating solutions to the optimization problem. Examples of wrapper include hill climbing algorithms such as forward selection and backward elimination, neural networks and genetic algorithms, best-first search, sequential floating forward selection and Monte Carlo tree search, .

Embedded methods incorporate the feature selection as the part of the training process, i.e. the feature selection process is "embedded" in the classifier. Here, the objective function is designed such that choosing a feature will maximize the MI between the feature and the class output and also minimize the MI between the selected feature and the subset of the so far selected features. This is given as:

$$I(Y, f) - \beta \sum_{s \in S} I(f; S)$$

where Y is the output, f is the current selected feature, s is the feature in the already selected subset S and β controls the importance of the MI between the current feature f and the features in the subset S. The output subset is applied to a Neural Network classifier. The above equation will select better subset

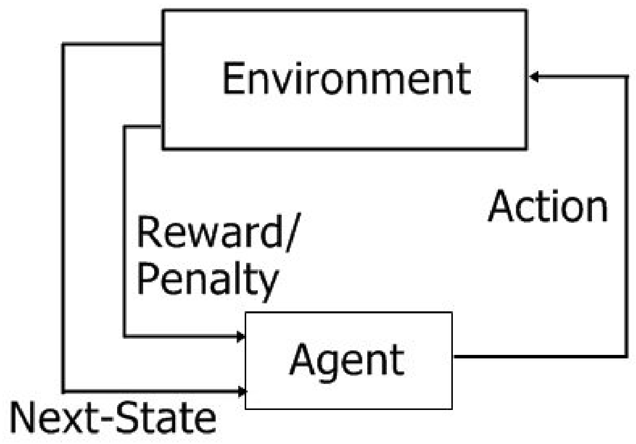


Figure 1: Reinforcement learning.

since the inter-feature MI is used in the calculation to select the non-redundant features.

4 Multi-agent Reinforcement learning

The figure 1. shows the block diagram of how reinforcement learning works. The agent preforms some action or interacts with the environment. In return the interaction with the environment gives back a reward/punishment depending on the action which was taken. The agent learns from this feedback signal or reward to modify its actions.

Reinforcement Learning [7] different from other type of learning algorithms that we study in machine learning. In Supervised Learning we learn or train our algorithm from the given outputs of the sample. In unsupervised learning we only use data to predict the samples. However in reinforcement learning we make decisions and compare actions to maximise a cumulative reward and not make predictions. Since a agent is involved in learning, there is trade-off problem between exploration and exploitation where the agent needs to decide between potentially finding better strategies at the expense of a getting a lower reward or if it should exploit what it already knows. With ϵ-greedy, we generate a random number, if the random number is less than the probability i.e. epsilon value, we choose to explore or the agent selects an action randomly. If the generated value is greater than the epsilon, we exploit the knowledge we already have, i.e. the higher q-value and choose the best possible action[8]. We follow the approach of multiple information learners where each agent considers other agents in the system as the part of the environment. The environment in such scenario is therefore dynamic.

In our approach we consider a global reward as our feedback signal[5]. This global reward is calculated by each agent. However this reward contains noise from exploratory actions which are taken by the other agents. Also, the agent is not able to distinguish if this noise has come from the environment or from other agents. CLEAN [1] rewards were used to overcome this problem. Clean rewards were defined as the difference between the global reward the agent would have got if it had made the exploratory action whereas all the agents have made the greedy action and the global reward if each of the action would have made the greedy action which can be given as formula[5] as:

$$C_i = G(a - a_i + c_i) - G(a)$$

Here $G(a)$ is the global reward when all agents made the greedy choice. $G(a - a_i + c_i)$ is the counter or the offline reward, agent i would have got if it had it made the counter action c_i instead of action a_i, while the rest of the agents have made the greedy choice. Each agent then updates it Q-value using:

$$Q(c_i) \leftarrow Q(c_i) + \alpha(C_i - Q(c_i))$$

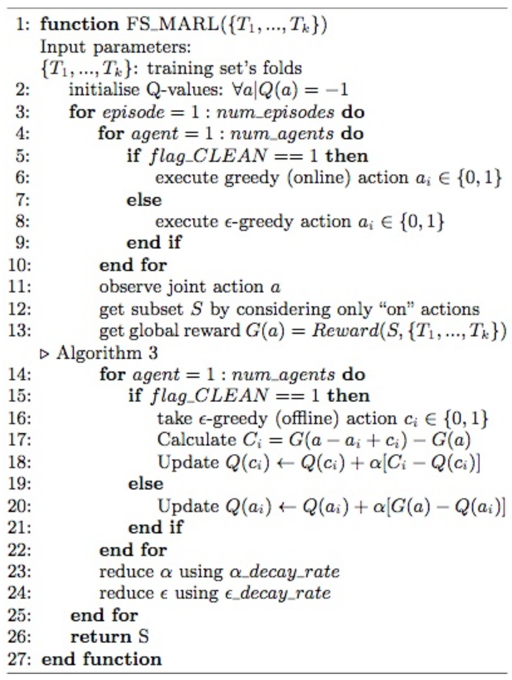
In my study I work on feature selection as a multi agent coordination problem[5] and implement the same. The central idea of the approach is to "assign" a learning agent to each feature. Thus, the total number of learning agents is equal to the number of features. The purpose of feature selection is to discard irrelevant, noisy and redundant features, by only proposing a subset of the features. Each reinforcement learning agent is therefore allowed to control a single feature; in other words, to decide whether a feature will be included in or excluded from the final feature subset.

There are two actions for each agent; "0" and "1". Action "0" indicates that the feature is "disabled" and will not be included in the feature subset. Similarly the, action "1" indicates that the feature is "enabled" and will not be included in the feature subset. Therefore, all the actions with 1's will be part of our feature subset. Let us now describe the reward function used. Our feature selection constitutes a multi-objective problem i.e. we will have to reduce the size of feature subset and increase the classification's performance at the same time which would be the accuracy or F-score. It should be emphasised that these two goals can be conflicting since reducing the number of features, valuable information can be lost.

Given a feature subset size as s, an upper boundary k for the subset's size and its corresponding classification performance P, we propose the reward function:

$$Reward = \left\{ \begin{array}{ll} P, & \text{if,} s \leq k \\ Pk/s, & \text{if,} s > k \end{array} \right\}$$

where k denotes the maximum number of features allowed and s denotes the subset size. Here we impose a penalty if the number of features in our subset is greater than the specified number of features i.e. k. The algorithm [5] is given as:



```
function FS_MARL($\{T_1, ..., T_k\}$)
Input parameters:
$\{T_1, ..., T_k\}$: training set's folds
    initialise Q-values: $\forall a | Q(a) = -1$
    for $episode = 1 : num\_episodes$ do
        for $agent = 1 : num\_agents$ do
            if $flag\_CLEAN == 1$ then
                execute greedy (online) action $a_i \in \{0, 1\}$
            else
                execute $\epsilon$-greedy action $a_i \in \{0, 1\}$
            end if
        end for
        observe joint action $a$
        get subset $S$ by considering only "on" actions
        get global reward $G(a) = Reward(S, \{T_1, ..., T_k\})$
▷ Algorithm 3
        for $agent = 1 : num\_agents$ do
            if $flag\_CLEAN == 1$ then
                take $\epsilon$-greedy (offline) action $c_i \in \{0, 1\}$
                Calculate $C_i = G(a - a_i + c_i) - G(a)$
                Update $Q(c_i) \leftarrow Q(c_i) + \alpha[C_i - Q(c_i)]$
            else
                Update $Q(a_i) \leftarrow Q(a_i) + \alpha[G(a) - Q(a_i)]$
            end if
        end for
        reduce $\alpha$ using $\alpha\_decay\_rate$
        reduce $\epsilon$ using $\epsilon\_decay\_rate$
    end for
    return S
end function
```

Figure 2: Multi-agent Reinforcement learning Algorithm

5 Experiment Design and results

5.1 Breast Cancer dataset

I used the breast cancer dataset from sklearn for my study. The data has 569 examples with 39 features. The data has 45 percent positive examples. The original dataset wasn't accessible and therefore I used a similar dataset for my study. The various features of the data had different scale, therefore required scaling else they might behave badly if the individual features do not more or less look like standard normally distributed data. I used standardScaler to scale the features. I implemented standardisation according to the following formula.

$$x_i^j = (x_i^j - \mu_i)/\sigma_i$$

Further, we split the data into training and test data set with test size to be 20 percent of our total data.

5.2 Comparing Methods

I compared the feature selection using reinforcement learning with various other state of the art methods from statistics. Univariate feature selection works by selecting the best features based on univariate statistical tests. Chi-square test, F-test ANOVA and mutual information are few of the methods available for feature selection which I have used in my study.

Each method was run for different values for k, i.e. the number of features we wish to include in the model. The value of upper boundary on the number of features to be included in the model is varied from 1 to 8. Further for our method using reinforcement learning we fixed the parameters α=0.2, ϵ=0.5, $\alpha decayrate$=0.995 and $\epsilon decayrate$=0.995. The number of episodes we train our model was kept to 1000.

5.3 Classification model and Evaluation metric

The classification method that I used is Logistic regression. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.I have set the split percentage for the test set to be 0.2. Four popular evaluation metrics are used for classification tasks are, accuracy, precision, recall and F-score (also known as F1-score). I've primarily used F-score for my study. F-score is a trade-off between precision and recall. If tp and tn be the number of true positives and true negatives respectively, and if fp and fn be the number of false positives and false negatives respectively the $F-score$ is given by

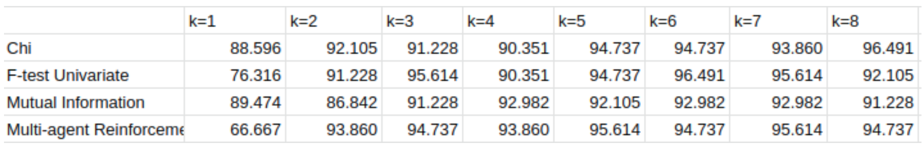
$$F-score = \frac{Precision(P)}{Recall(R)}$$

$$where, Precision(P) = \frac{tp}{tp+fp} Recall(R) = \frac{tp}{tp+fn}$$

There were other metrics which could have been chosen for our study but F-score seemed to be the best choice for the study because while using other metrics they resulted in poor F-score whereas while choosing an optimal F-score, it showed good results in accuracy, precision and recall. And since we are interested in obtaining an overall good performance, F-score seemed to be the best choice.

6 Analysis of results

I computed the F-scores for each of the feature selection method and plotted them on the graph as shown in figure 3. and found out few interesting results. The F-test and Multi agent reinforcement learning with CLEAN rewards performed fairly poor when the number of features were selected to be very less i.e. 1 in our case. On increasing the value of upper boundary on the number of features to be included to 2 mutual information performed poor in comparison to other methods. Multi-agent reinforcement learning outperformed other methods. F-test and chi square performed fairly well. For more values of k, multi-agent was fairly stable and provided the better performance than other feature selection methods. F-test was the most unstable in providing the results



	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8
Chi	88.596	92.105	91.228	90.351	94.737	94.737	93.860	96.491
F-test Univariate	76.316	91.228	95.614	90.351	94.737	96.491	95.614	92.105
Mutual Information	89.474	86.842	91.228	92.982	92.105	92.982	92.982	91.228
Multi-agent Reinforceme	66.667	93.860	94.737	93.860	95.614	94.737	95.614	94.737

Figure 3: Performance Evaluation: Results

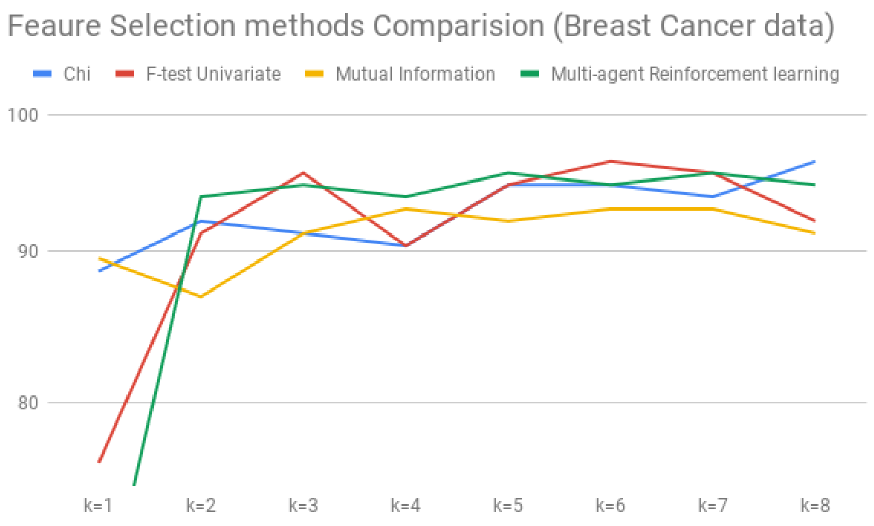


Figure 4: Performance Evaluation: Feature Selection using various methods

with multiple ups and downs. Chi-test gave poor performance for less features whereas mutual information gave poor performance for all values of k.

7 Summary

In my study I've implemented feature selection as multi-agent reinforcement learning problem and studied its performance with respect to the other traditional state of the art methods. My approach mainly focused on assigning an agent to the each feature. The algorithm learns to find out if the feature should be part of our model or not using the global and counterfactual rewards. I have also accommodated the CLEAN rewards that removes the exploratory noise created by other learning agents. My experimental study was conducted on breast cancer dataset and feature selection was done using Chi-square, F-test, Mutual Information and multi-agent reinforcement learning. Multi-agent reinforcement learning performed fairly well except for very low value of k or features. Mutual Information on the other performed poorly. F-test was the most unstable for my study.

Future Work

The following algorithm can be tested on high dimensional datasets as well and the performance can be evaluated and compared with other traditional methods. Moreover, many hybrid approaches can be tested as well which incorporate one or two methods to find the best features for the given dataset.

Acknowledgement

I'm really thankful to Professor James Park for helping me with the essential concepts which were needed for the success of the study. His lectures of Artificial Intelligence II were very insightful. His teaching style made the advanced topics seemed easy to understand.

References

[1] A. K. Agogino C. Holmesparker, M. E. Taylor and K. Tumer. *Clean rewards to improve coordination by removing exploratory action noise.* In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03, 2014.

[2] C. Claus and C. Boutilier. *The dynamics of reinforcement learning in cooperative multiagent systems.* In Proceedings of the AAAI Conference on Artificial Intelligence, 1998.

[3] I. Guyon and A. Elisseeff. *An introduction to variable and feature selection.* The Journal of Machine Learning Research, 2003.

[4] M. A. Hall. *Correlation-based feature selection for machine learning.* PhD thesis, The University of Waikato, 1999.

[5] Gary Brooks George Frangou Kleanthis Malialis, Jun Wang. Feature selection as a multiagent coordination problem. *arxiv*, 2016.

[6] G. Chandr lashekar and F. Sahin. *A survey on feature selection methods*, volume 40(1):16–28. Computers Electrical Engineering, 2014.

[7] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning.* MIT Press Cambridge, MA, USA, 1998.

[8] C. J. Watkins and P. Dayan. *Q-learning.* Machine learning, 8(3-4):279–292, 1992.