# Investigation of Ensemble Feature Selection for classification

Mathematical Information Group M1 Wang Youbin

November 16, 2021

**Abstract**

With proliferation of extremely high-dimensional data, feature selection has been a mandatory preprocessing process of machine learning. Ensemble learning, which combines the output of multiple models, is a prolific field in machine learning for its good performance and has been commonly used for classification and regression. Ensemble method can also be implemented in feature selection and is expected to get more robust feature subset and solve the problem of selecting appropriate feature selection algorithm for different scenarios.

## 1 Introduction

In some areas of supervised classification, such as gene micro-array analysis and text classification,it is common to deal with datasets containing large number of feature with small ratio between number of samples and number of features, which implies that part of features are irrelevant to class concept and redundant in presence of other features. Feature selection is a process of removing irrelevant features and redundant features from original features for dimension reduction. Feature selection can improve learning accuracy by reducing overfitting, speeding up learning process, leading to better understand of model, simplifying the learning model, saving the effort on data collection and guiding the following knowledge discovery by domain experts.

### 1.1 Categories of feature selection

Feature selection can be classified into individual evaluation and subset evaluation in terms of the output of feature selection algorithm. Individual evaluation's output is rankings of the features in terms of degree of relevance to class concept, while the output of subset evaluation is feature subset of original features based on certain search strategy. Note that the weight of the features generated by the feature selection algorithm can be turned to rankings by ranking the weight, and rankings can be turned to feature subset by selecting part of top ranked features. Besides, feature selection can be divided into filter method, wrapper method and embedded method in terms of feature selection's relation with learning process.[4]

- filter method: feature selection is implemented independently of the learning process and the features are assigned weight in terms of their relevance to class concept based on statistical characteristics of data. These methods have the advantage of being fast and independent of classification model, but it can't detect redundant features and often provide inferior results.
- wrapper method: Feature subset are selected guided by the classification performance of predetermined model. They often have better results than filter methods but at cost of high computation time.
- embedded method: Feature selection is embedded in the process of training and are usually specific to given classification method. Currently, most embedded methods are designed by adding regularization part of loss function of learning model to achieve a sparse solution. They often provide trade-off between performance and computation cost.

## 1.2 Evaluation metric of feature selection

Feature selection method can be evaluated in terms of performance of learning method used afterward, the robustness of selected feature subset and ability of selecting relevant features when the knowledge of relevant features is available.

### 1.2.1 stability

Stability of feature selection is defined as the sensitivity of feature selection algorithm's output to variations or small change in training set(eg. by removing or adding samples or by adding noise to features). The stability of feature selection is measured by computing similarity of outputs of feature selectors on different subsamples of training set. For description of method, $S(.,.)$ represents similarity function of two feature selection output. $f_i$ is a vector of length D representing the output of feature selection on $i_{th}$ subsample of data,where D is the number of features. $f_i^j$ represent the rank of feature j in the case feature ranking and $f_i^j = 1$ if $j_{th}$ feature is presented in the subset, and 0 otherwise in the case of feature subset selection.

For **feature ranking**, the Spearman rank coefficient is used[9], where $S_W$ takes value in [-1,1]; value of 1 means rankings are identical, a value of 0 means that there is no correlation between rankings, and a value of -1 means that rankings have inverse order.

$$S_R(f_i, f_j) = 1 - 6\sum_{l=1}^{D} \frac{(f_i^l - f_j^l)^2}{D(D^2-1)} \tag{1}$$

For **feature subsets**, Jaccard index is used[9], where $S_S$ takes value in [0,1] with 0 meaning that there is no overlap between two sets and 1 that two sets are identical.

$$S_S(f_i, f_j) = \frac{|f_i \cap f_j|}{|f_i \cup f_j|} = \frac{\sum_{l=1}^{D} I(f_i^l = f_j^l = 1)}{\sum_{l=1}^{D} I(f_i^l + f_j^l > 0)} \tag{2}$$

The overall stability is defined as average over all pairwise similarity comparisons between different feature selection result on all subsamples, where k is the number of subsamples of dataset.

$$S_{tot} = \frac{2\sum_{i=1}^{k}\sum_{j=i+1}^{k} S_S(f_i, f_j)}{k(k-1)} \tag{3}$$

### 1.2.2 performance of learning model

The classification performance of learning model trained on the selected features are considered since the objective of feature selection is to improve the model performance. The performance can be measured by precision, recall, F1 score and AUC, etc.

### 1.2.3 ability of selecting relevant features

When the information of relevant features are available, feature selection algorithm's ability to locate relevant features can be evaluated. For descriptions of the methods, note that **feat_sel** represents the subset of selected features, **feat_rel** is subset of relevant features and **feats** is the originial feature set.

For **feature subset**, **F1 score** is used [3]

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4}$$

$$precision = \frac{feat\_rel \cap feat\_sel}{feat\_sel} \tag{5}$$

$$recall = \frac{feat\_rel \cap feat\_sel}{feat\_rel} \tag{6}$$

For **feature rankings**, it can be evaluated by establishing a threshold to transform the rankings to feature subset and use the metric above to evaluate, where the value is affected by threshold. And it can also be evaluated by methods specially defined to evaluate rankings, such as Ranking loss[3]

which evaluates the number of irrelevant features that are better ranked than relevant ones, where **pos** stands for the position of the last relevant feature in the ranking. The fewer irrelevant features are on the top of the ranking, the bigger the value is.

$$R = \frac{pos - \#feat\_rel}{\#feats - \#feat\_rel} \tag{7}$$

However, the set of relevant features are not known unless we are using artificial data set.

### 1.3 Focused problem of feature selection

**Robustness**:One of the most important application of feature selection is knowledge discovery, where domain experts will invest and research on the selected features to gain more insight into the problem modeled. For knowledge discovery, robustness of selected features is a desirable characteristics, especially if subsequent analyses are costly. Domain experts tend to have less confidence in feature sets that change radically with slight variations in training set. However, it is reported single feature selection algorithm is quite unstable when faced changing data characteristics.[6]

**Selection of feature selection in different scenario**:There are large number of available feature selection algorithms, however, choosing the adequate feature selection method is not a easy question and it has been found by experiment that feature selection algorithm may have significant difference on different dataset(nonlinearity of data, noise in inputs and in the target). It is said that there is not a so-called "best" method.[3]

**Interaction between features**:It is analyzed in [4] that individually redundant features and irrelevant features may not be redundant or irrelevant when interacted with other features and can even provide significant performance improvement. And the situation is more complicated in real world data. As a result, multivariate feature selection which considers the feature interaction is better than univariate feature selection which evaluates each feature independently.

## 2 Ensemble Feature Selection

Inspired by success of ensemble learning in classification and regression, ensemble techniques are also applied to feature selection and are expected to improve the robustness and circumvent the problem of selecting appropriate feature selection algorithm for different scenario,under the assumptions that combining different feature selectors' results obtain better result than output of any individual ones. In [9], it is discussed that, by combining different feature selection, ensemble feature selection may help in reducing the risk of choosing an unstable subset, giving better approximation to the optimal subset or ranking of features, and expanding search space of feature subset space.

There are three essential steps in creating a feature selection ensemble. The first step is to create a set of different feature selectors while the second step aggregates the results of single feature selector, and finally a practical subset is selected according to particular threshold in third step.

### 2.1 Generation of different feature selection

The ensemble feature selection can be categorized into homogeneous ensemble and heterogeneous ensemble in terms of the way to generate different feature selectors.[10]

**Homogeneous**: N selections using the same feature selection algorithms, with each selection using different training data. When the size of training data is huge, the diversity in training set is achieved by partitioning the data into N nodes, while for small dataset, bootstrap is implemented to generate N data set just like bagging.

**Heterogeneous**: N selections using different feature selection algorithms that use the same training data.

Both approaches aims at improving the stability and classification performance by adding diversity. The first approach serves for particular feature selection algorithm by including diversity of data and could also improve the computation time by processing data in parallel nodes by partitioning the data when the dataset's size is huge. The second approach includes the diversity by combining different feature selection algorithms and is expected to perform well in all scenarios, so as to take advantage of the strengths of the individual selectors and overcome their weakness.

### 2.2 feature selection aggregation

To avoid computation cost, feature selection algorithms in ensemble feature selectors are often filter method and wrapper method, whose output is or could be transformed to the rankings of features. After getting feature rankings from different feature selectors, the feature rankings can be combined in many combination methods to produce a final output. Frequently used combination methods are simple strategy-choosing minimum, maximum, median, mean rankings among rankings of different feature selections as the feature's final output.

### 2.3 Threshold

After getting the final output, it is necessary to set a threshold to obtain a practical subset of features. Frequently used methods are selecting different percentages of features or using cross validation strategy with classification performance of classification model as metric to decide how many features to select.

## 3 Experiment

The experiment on high dimension dataset is implemented to compare homogeneous ensemble feature selection and heterogeneous feature selection with single feature selectors in terms of classification performance and robustness. And the ability to select relevant features are also evaluated in artificial dataset. The experiment setting refers to [2].

### 3.1 Data set

The datasets chosen all have small ration of features number over samples size and have different fields of knowledge, image data, micro-array data, mass-spectrometric data, and text classification data which may imply different real world data scenarios. The datasets are described briefly in table 1.

**Table 1:** Datasets Description.

| name | feature number | samples size | class 1 | class 2 | Description |
|---|---|---|---|---|---|
| Arcene[11] | 10000 | 200 | 112 | 88 | mass-spectrometric data to distinguish cancer versus normal pattern |
| Dexter[11] | 20000 | 600 | 300 | 300 | text classification task to classify which article are about corporate acquisition |
| Gisette[11] | 5000 | 1000 | 500 | 500 | handwritten digit recognition problem to distiinguish highly confusing digit '4' and '9' |
| colon[1] | 2000 | 62 | 40 | 22 | micro-array data to distinguish normal and tumor tissue |
| Artificial | 10000 | 300 | 150 | 150 | described below |

Artificial data set with labels of relevant features are also generated to inspect the ability to select relevant features. It consists of 100 relevant features and 9900 irrelevant features. Relevant features consists of 50 correlated features and 50 uncorrelated features, which generated by zero mean Gaussian distribution while the correlation between correlated features is randomly generated. The relevant features are generated by uniform distribution. 300 samples are generated. To generate balanced labels, the relevant features are first summed, then the median of sum over all samples is subtracted, finally the sign is taken, which provides 150 positive labels and 150 negative labels.

$$\begin{gathered}\text{correlated relevant features} \sim N(0,\Sigma) \text{ with } \Sigma_{i,j} \sim u(0,1) \text{ and } \Sigma_{i,j} = \Sigma_{j,i} \\ \text{uncorrelated features} \sim N(0,I) \\ \text{relevant features} \sim u(0,1)\end{gathered}$$

### 3.2 Feature selection algorithms used

The feature selection algorithms chosen in experiment are two filter method and one embedded method. For the filter method, Relief-F and Symmetrical Uncertainty are chosen. For embedded method, SVM-RFE are chosen. Each feature selection algorithm is briefly introduced below. The reason for choosing

these feature selection algorithm is that they are based on different metrics and so ensure diversity. The implementation of these method. The algorithms implementation refers to scikit-feature repository [8].

#### 3.2.1 Relief-F

ReliefF[7] is a multivariate filter method taking into account feature interaction. The idea of Relief is that relevant feature's value should be different from near different class instance and be same for near same class instance. For each sample $x_i$, the k nearest neighboring samples, using L1 norm as metric, belonging to the same class $\{Nearhit_{i,j}\}_{j=1,\dots,k}$ and the ones of opposite class $\{Nearmiss_{i,j}\}_{j=1,\dots,k}$ are selected. The difference is taken to update the weight vector W representing how well each feature distinguishes samples of two classes. Hyper-parameter k is set to 5 in this experiment.

$$W := W - \frac{1}{k}\sum_{j=1}^{k}|x_i - Nearhit_{i,j}| + \frac{1}{k}\sum_{j=1}^{k}|x_i - Nearmiss_{i,j}|$$

#### 3.2.2 Symmetrical Uncertainty

Symmetrical Uncertainty[9] is a univariate filter method considering each feature separately. For every feature, the symmetrical uncertainty, which is normalized mutual information between features and class, is computed as feature's ability to discriminate different class.

$$SU(F,C) = 2 \times \frac{H(F) - H(F|C)}{H(F) + H(C))}$$

where $F$ and $C$ are random variables standing for feature and class, and function H computes the entropy.

#### 3.2.3 SVM-RFE

Feature importance can be considered as the sensitivity of objective function for the absence of feature. For a linear SVM, it can be derived from the weight vector of hyperplane. For SVM-RFE[5], the linear SVM is trained on the present feature set, and C-parameter is tuned using an internal cross-validation of training set. Features are ranked according to corresponding absolute value of weight in the weight vector of the hyperplane, and the k% worst features are discarded. The above procedure is repeated until the feature set containing the desired number is reached. The output of SVM-RFE is the rankings of features where the features discarded at same time are assigned the same rank. In the experiment, hyper-parameter k is set to 10% and 1% features is selected after the recursion procedure.

### 3.3 Ensemble techniques

#### 3.3.1 Generation of different feature selectors

**homogeneous ensemble**: bootstrap strategy is used to generate 40 bags from the training set. For each of the bag, a separate feature selection with same feature selection algorithm described above was performed and the ensemble was formed by the combination strategy described in 3.3.2.

**heterogeneous ensemble** : On training set, separate feature selection with different feature selection algorithm described above was performed and the ensemble was formed by the combination strategy described in 3.2.2.

#### 3.3.2 Combination

To combine the feature selection from different feature selectors, the output of feature selection are turned into weights range from 0 to 1 by the following strategy. For the rankings output, the features with the same rank are permuted randomly and the rankings are normalized into 0 and 1, where bigger value represents better rankings. For weights output, the weights are normalized into 0 and 1. And the final features' weight was formed by simply averaging the weights over all feature selectors' weight output, where $w_i$ is the normalized weight vector of $i_{th}$ feature selector, S is the number of feature selectors and $W$ represents the final output.

$$W = \frac{1}{S}\sum_{i=1}^{S} w_i \tag{8}$$

### 3.4 Experiment flow

Every data set is divided by 10-Fold cross validation. Subsequently, feature selection is performed on each fold's training set.

**robustness**: To analyze the robustness, Jaccard index are computed on all pairwise combinations of feature selections' output for each fold and the overall stability are the average over number of combinations. And Jaccard index was analyzed for different subset sizes: the top 1%, top 5% and top 10% best features of the rankings. Table 1 summarizes the result of robustness analysis across different datasets.

| Dataset | | SU | RLF | SVM_RFE | Homo_SU | Homo_RLF | Homo_SVM | Heter |
|---|---|---|---|---|---|---|---|---|
| **Arcene** | JC1% | 0.44 | 0.75 | 0.25 | 0.43 | 0.66 | 0.36 | 0.61 |
| | JC5% | 0.65 | 0.64 | 0.24 | 0.68 | 0.66 | 0.54 | 0.58 |
| | JC10% | 0.73 | 0.72 | 0.53 | 0.79 | 0.75 | 0.59 | 0.65 |
| | mean | 0.60 | 0.70 | 0.34 | 0.63 | 0.69 | 0.50 | 0.61 |
| **Dexter** | JC1% | 0.85 | 0.63 | 0.65 | 0.89 | 0.72 | 0.67 | 0.74 |
| | JC5% | 0.88 | 0.57 | 0.30 | 0.87 | 0.74 | 0.61 | 0.74 |
| | JC10% | 0.84 | 0.26 | 0.69 | 0.83 | 0.69 | 0.66 | 0.73 |
| | mean | 0.86 | 0.49 | 0.55 | 0.86 | 0.72 | 0.65 | 0.74 |
| **Gisette** | JC1% | 0.78 | 0.77 | 0.30 | 0.76 | 0.74 | 0.54 | 0.74 |
| | JC5% | 0.71 | 0.69 | 0.25 | 0.67 | 0.66 | 0.62 | 0.62 |
| | JC10% | 0.69 | 0.66 | 0.54 | 0.69 | 0.69 | 0.62 | 0.63 |
| | mean | 0.72 | 0.70 | 0.36 | 0.71 | 0.69 | 0.59 | 0.66 |
| **colon** | JC1% | 0.20 | 0.84 | 0.47 | 0.20 | 0.67 | 0.41 | 0.64 |
| | JC5% | 0.05 | 0.74 | 0.27 | 0.06 | 0.86 | 0.49 | 0.56 |
| | JC10% | 0.07 | 0.73 | 0.56 | 0.07 | 0.85 | 0.59 | 0.55 |
| | mean | 0.11 | 0.77 | 0.43 | 0.11 | 0.8 | 0.50 | 0.59 |
| **Artificial** | JC1% | 0.00 | 0.59 | 0.50 | 0.00 | 0.97 | 0.49 | 0.54 |
| | JC5% | 0.02 | 0.42 | 0.24 | 0.02 | 0.44 | 0.43 | 0.40 |
| | JC10% | 0.05 | 0.46 | 0.45 | 0.05 | 0.45 | 0.44 | 0.45 |
| | mean | 0.03 | 0.49 | 0.40 | 0.03 | 0.62 | 0.45 | 0.46 |

**Table 2:** Robustness of different feature selectors across different datasets. SU represents Symmetrical Uncertainty, RLF represents Relief-F, Homo_f represents the homogeneous ensemble of feature selection algorithm $f$ and Heter represents the Heterogeneous ensemble consisting of Relief-F, Symmetrical Uncertainty and SVM_FRE. Jaccard index on subset of 1%, 5% and 10% best features are denoted respectively by JC1%, JC5% and JC10%

**classification performance**: To analyze the feature selection algorithm in terms of the classification performance, for each fold, top 1% ranked features are selected based on feature selection's output. Linear SVM are trained on the training set containing selected features and C hyper-parameter is selected using 10-fold cross-validation strategy, then the trained model's accuracy on the test set was computed for each fold. The overall performance is evaluated by average of test set's accuracy over all folds. Table 2 summarizes the result of accuracy analysis for different datsets.

**Table 3:** Accuracy of different feature selectors accross different datasets

| accuracy / feature selector / dataset | SU | RLF | SVM_RFE | Homo_SU | Homo_RLF | Homo_SVM | Heter |
|---|---|---|---|---|---|---|---|
| **Arcene** | 0.76 | 0.80 | 0.55 | 0.71 | 0.76 | 0.75 | 0.81 |
| **Dexter** | 0.89 | 0.85 | 0.90 | 0.88 | 0.87 | 0.90 | 0.87 |
| **Gisette** | 0.87 | 0.84 | 0.88 | 0.87 | 0.85 | 0.90 | 0.85 |
| **colon** | 0.75 | 0.76 | 0.87 | 0.77 | 0.83 | 0.79 | 0.87 |
| **Artificial** | 0.58 | 0.96 | 0.97 | 0.59 | 0.97 | 0.97 | 0.97 |

In order to jointly evaluate the trade-off between robustness and classification performance for

feature selection algorithm, the harmonic mean of robustness and classification performance, which is named after RPT(robustness-performance trade-off) in [9].For robustness measure the mean of Jaccard index of top 1%, 5% and 10% ranked features is used, while for classification performance the accuracy of models trained on top 1% ranked features is used. The RPT result are summarized in table4. And the comparison between homogeneous ensemble, heterogeneous ensemble and single feature selectors are plotted with accuracy against robustness in figure 2. And the robustness and classification performance metrics are same as described above.

$$RPT = 2 \times \frac{robustness \times performance}{robustness + performance}$$

**Table 4:** RPT of different feature selectors accross different datasets

| RPT: feature selector / dataset | SU | RLF | SVM_RFE | Homo_SU | Homo_RLF | Homo_SVM | Heter |
|---|---|---|---|---|---|---|---|
| **Arcene** | 0.34 | 0.37 | 0.21 | 0.34 | 0.36 | 0.30 | 0.35 |
| **Dexter** | 0.44 | 0.31 | 0.34 | 0.44 | 0.39 | 0.38 | 0.40 |
| **Gisette** | 0.40 | 0.39 | 0.26 | 0.39 | 0.38 | 0.36 | 0.37 |
| **colon** | 0.09 | 0.38 | 0.29 | 0.10 | 0.37 | 0.31 | 0.35 |
| **Artificial** | 0.03 | 0.30 | 0.29 | 0.03 | 0.30 | 0.33 | 0.32 |

**ability to select relevant features**:For artificial data set, since the features label indicating whether the feature is relevant or not are available,it is able to evaluate the feature selection algorithm's ability to select relevant features. For each fold, the top $n\_relevant$, the number of relevant features, features are selected based on the features rankings and F1 score is computed. And the ranking loss are also computed for each folds. The result is illustrated in table 5.

**Table 5:** Analysis of ability to select relevant features for different features selectors on analysis dataset

| | SU | RLF | SVM_RFE | Homo_SU | Homo_RLF | Homo_SVM | Heter |
|---|---|---|---|---|---|---|---|
| **F1 Score** | 0.01 | 0.67 | 0.63 | 0.01 | 0.98 | 0.58 | 0.65 |
| **Ranking loss** | 0.99 | 1.00 | 0.91 | 0.99 | 0.12 | 0.21 | 0.90 |

## 3.5 Analysis

It can be observed form Figure 2 that the classifier with feature selection gets comparable classification performance as that without feature selection on most datasets unless dexter, which implies that there are irrelevant features that do not contribute to the classification and demonstrate the importance and necessary of feature selection. And it is expected to get better result to select more features to get better classification performance by selecting appropriate number of features by cross validation strategy.

For homogeneous ensemble, it can be observed from first column of figure 2 that the robustness of selected features can be improved compared with the single feature selection, such as SVM-RFE in artificial, arcene, dexter dataset or relief in dexter dataset and the classification performance is also significantly improved for SVM-RFE in arcene dataet. However, it can be also observed that the homogeneous ensemble improves the robustness slightly with the degradation of classification performance such as the SVM-RFE in colon dataset. And there are situations where both robustness and classification performance of homogeneous ensemble feature selector slightly degrades compared with single feature selector, such as relief in arcene dataset.

For heterogeneous ensemble, it can be observed from the second column of figure 2 that, in every dataset, heterogeneous ensemble gets comparatively good results while single feature selection method achieve better result on some datasets but failed on others. However, unlike ensemble learning in classification and regression, heterogeneous ensemble's performance in terms of accuracy and classification performance are always better than the worst feature selector but worse than the outperformed single one.

As for the ability to select relevant features, from table 5, it can be observed that the ability to select relevant features is improved significantly for Relief's homogeneous ensemble compared with

the single one in terms of both of F1 score and Ranking loss. And SVM-RFE's Ranking Loss is also significantly improved by homogeneous ensemble which means that the relevant features are ranked better than irrelevant features. And the weights of features of the above two feature selectors are also plotted in figure 1. It can be seen that by including the diversity in data, the weight of relevant uncorrelated features are increased by homogeneous ensemble compared with single one. As for Symmetrical Uncertainty, there are no improvement for their failure to detect feature interaction. And heterogeneous ensemble, same as that in classification performance and robustness analysis, performs much better than the worst single one while worse than the best single one.

In general, it can be observed that homogeneous feature selection can provide more robust feature subsets than a single feature selection algorithm, however, the improvement depends on dataset and feature selection algorithm. Heterogeneous ensemble by simply averaging the result of feature selectors have comparatively good result in any scenarios but always worse than the best single one.

# 4 Future Work

The future work is to study on the **chooses of feature selection algorithm**, the **combination methods** and the **automatic threshold method** for ensemble feature selection. There are lots of feature selection method available, each of which has pro and cons, the methods selected should guarantee diversity while increasing the regularity, so as to take advantage of them to boost performance. It can be seen from the experiment, simple average combination could not boost the performance of feature selection and more sophisticated method is expected to be studied to make the result better than any of the single feature selectors like ensemble in classification and regression. Simply select the top ranked part of features may result in missing important features while the cross validation strategy suffers from high time computation. An automatic threshold based on the statistical characters are expected to be studied to get the final feature subset efficiently and correctly.

# References


[1] Uri Alon, Naama Barkai, Daniel A Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[2] Alexander Goscinski Benjamin Rabier. Ensemble methods for feature selection. https://github.com/agoscinski/EnsembleMethodsForFeatureSelection, 2018.

[3] Verónica Bolón-Canedo and Amparo Alonso-Betanzos. Ensembles for feature selection: A review and future trends. *Information Fusion*, 52:1–12, 2019.

[4] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[5] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[6] Alexandros Kalousis, Julien Prados, and Melanie Hilario. Stability of feature selection algorithms. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

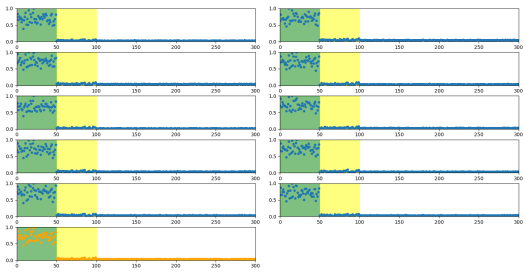
[7] Igor Kononenko. Estimating attributes: Analysis and extensions of relief. In *European conference on machine learning*, pages 171–182. Springer, 1994.

[8] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2018.

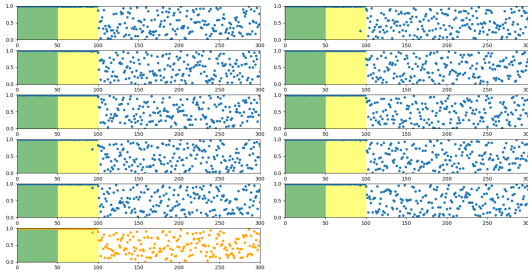
[9] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 313–325. Springer, 2008.

[10] Borja Seijo-Pardo, Iago Porto-Díaz, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. Ensemble feature selection: homogeneous and heterogeneous approaches. *Knowledge-Based Systems*, 118:124–139, 2017.

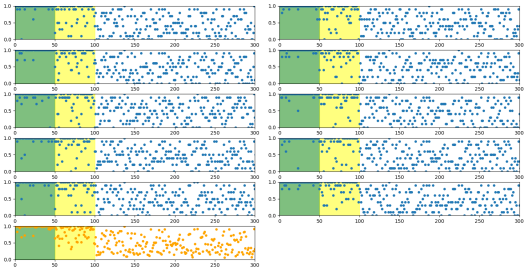
[11] CA Whistler, British Columbia. Nips 2003 workshop on feature extraction. http://clopinet.com/isabelle/Projects/NIPS2003/, 2003.



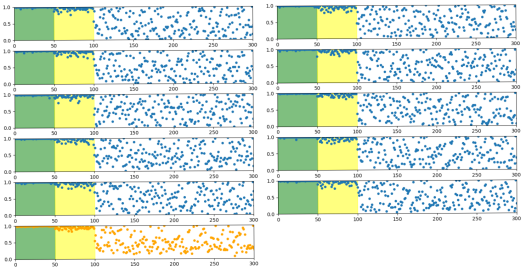
(a) weight of features for Relief



(b) weight of features for Relief's Homogeneous Ensemble

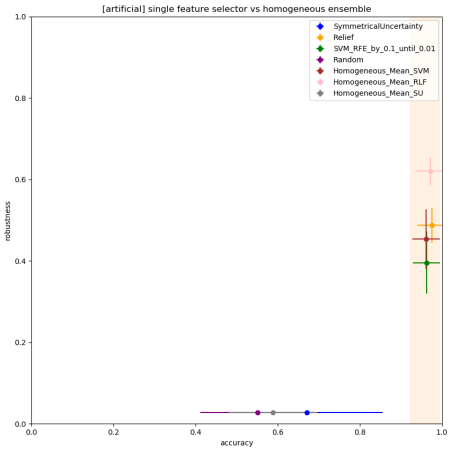


(c) weights of features for SVM-RFE

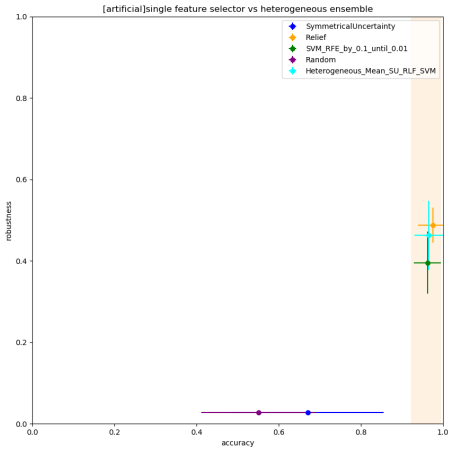


(d) weights of features for SVM-RFE's Homogeneous Ensemble

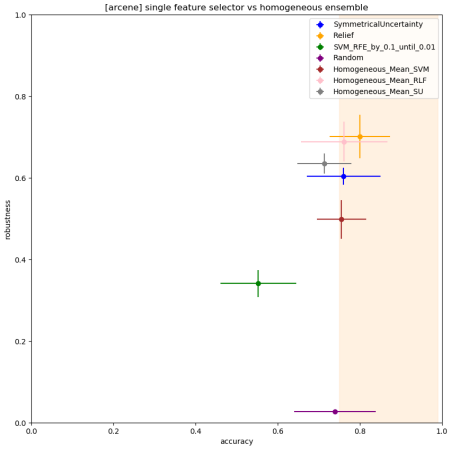
**Figure 1:** Features' weight of artificial dataset generated by single feature selector and Homogeneous Ensemble of Relief and SVM-RFE. There are eleven weights plot for each feature selector, the first 10 plots represent the weights generated by feature selector on each fold's training set and the last plot is the features' mean weight over all folds. The x-axis represents the index of features and y-axis represents weight. The correlated relevant features' indices are from 0 to 49(green area) and the uncorrelated relevant features indices are from 49 to 99(yellow area). The irrelevant features' indices are from 99 to 9999, for the ease of observation only part of irrelevant features' weights are plotted.



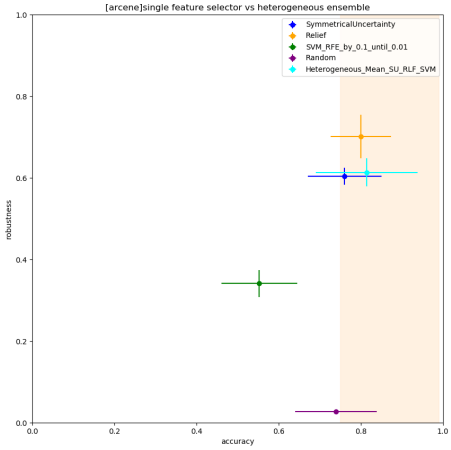

(a) homogeneous ensemble vs single feature selector on artificial data set



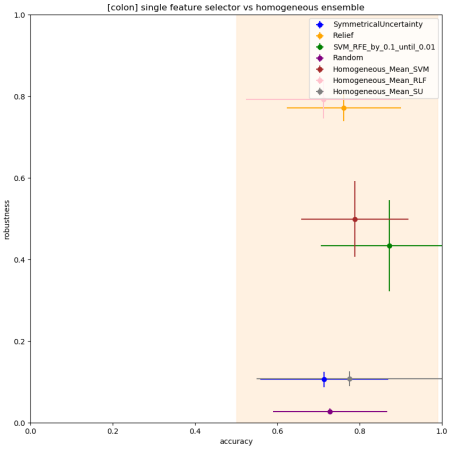

(b) heterogeneous ensemble vs single feature selector on artificial data set



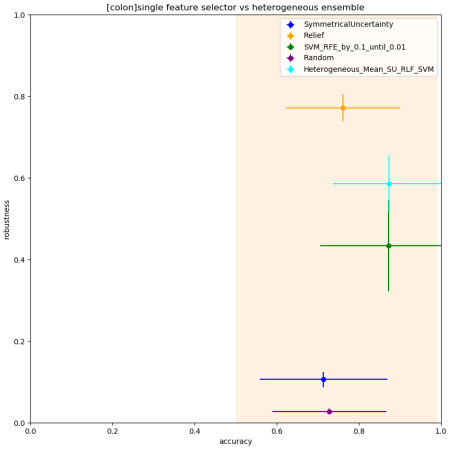

(c) homogeneous ensemble vs single feature selector on arcene dataset



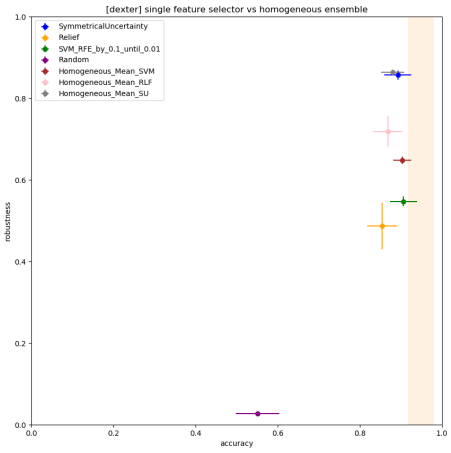

(d) heterogeneous ensemble vs single feature selector on arcene data set



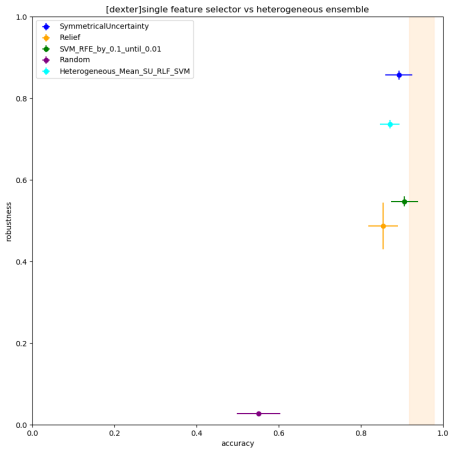

(e) homogeneous ensemble vs single feature selector on colon data set



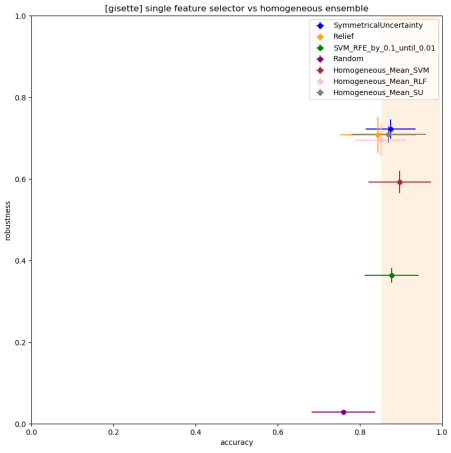

(f) heterogeneous ensemble vs single feature selector on colon data set



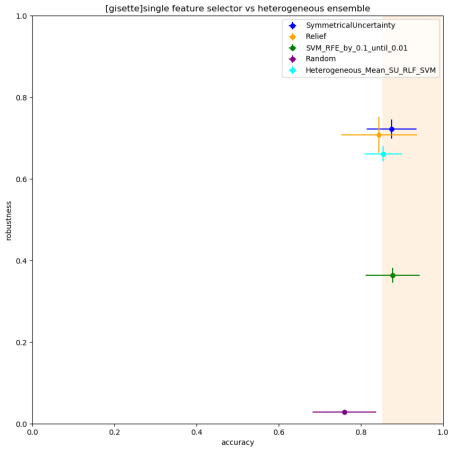

(g) homogeneous ensemble vs single feature selector on dexter data set




(h) heterogeneous ensemble vs single feature selector on dexter data set




(i) homogeneous ensemble vs single feature selector on gisette data set




(j) heterogeneous ensemble vs single feature selector on gisette data set

**Figure 2:** Comparsion between homogeneous ensemble, heterogeneous ensemble and single feature selectors with classification accuracy against robustness on different dataset. The error bar of x axis represent standard deviation of jaccard index on all pairwise combinations of different fold's feature selection. The errorbar of y axis stands for standard deviation of classification accuracy on all folds' test set. the fleshcolor area is the accuracy with all features on every folds' test set.