A Network Clustering based Software Attribute Selection for Identifying Fault-Prone Modules

ICITCS 15’

Software fault prediction

How to measure the software quality?

- Static code analysis

How to detect the software fault?

- Poor design
- Structural problems

Complexity measures

Module-based software complexity measures

McCabe

- Graph-theoretic complexity
- The more complicated software → error-prone

Halstead

- Based on the number of operators and operands
- The software is harder to read or understand → error-prone

Complexity measures

Feature	Description
total loc	The total lines of code
blank loc	Lines of blanks
comment loc	Lines of comments
code and comment loc	Lines of code and comments
executable loc	The executable source lines of code
branch count	Branch count of the flow graph
decision count	Decision count
call pairs	Executable call pairs between modules
condition count	Condition count
multiple condition count	Multiple condition count
cyclomatic complexity	Cyclomatic complexity
cyclomatic density	Cyclomatic density (cyclomatic complexity divided by the lines of code)
decision density	Decision density (condition decision metric divided b y the decision count)
design complexity	Design complexity (the number of paths which calls s omething in the control flow)
design density	Design density (design complexity divided by cyclom atic complexity)
normalized cyclomatic complexity	Normalized cyclomatic complexity
formal parameters	The number of formal parameters

Feature	Description
unique operands(n_2)	The number of unique operands
unique operators(n_1)	The number of unique operators
total operands(N_2)	The number of operands
total operators(N_1)	The number of operators
halstead vocabulary(n)	The length of unique operands and operat ors ($n_1 + n_2$)
halstead length(N)	The length of operands and operators ($N_1 + N_2$)
halstead volume(V)	The measure of complexity (N * $\log_2 n$)
halstead level(L)	The implementation level of the program ($\frac{2*n_2}{n_1*N_2}$)
halstead difficulty(D)	The measure of difficulty ($\frac{n_1}{2} * \frac{N_2}{n_2}$)
halstead effort(EFF)	The efforts required to understand or impl ement the program (D * V)
halstead error(ERR)	The estimated number of errors in the im plementation (V/3000)
halstead time(T)	The time required to understand or imple ment the program (EFF/18)

Complexity measures

total loc	The total lines of code
blank loc	Lines of blanks
comment loc	Lines of comments
code and comment loc	Lines of code and comments
executable loc	The executable source lines of code
branch count	Branch count of the flow graph
decision count	
call pairs	
condition count	
multiple condition count	
cyclomatic complexity	
cyclomatic density	
decision density	y the decision count)
design complexity	Design complexity (the number of paths which calls something in the control flow)
design density	Design density (design complexity divided by cyclomatic complexity)
normalized cyclomatic complexity	Normalized cyclomatic complexity
formal parameters	The number of formal parameters

unique operands(n_2)	The number of unique operands
unique operators(n_1)	The number of unique operators
total operands(N_2)	The number of operands
total operators(N_1)	The number of operators
	The length of unique operands and operat ... $+ n_2$)
	...gth of operands and operators (...)
	...asure of complexity ($N * \log_2 n$)
	...lementation level of the program
halstead difficulty(D)	The measure of difficulty ($\frac{n_1}{2} * \frac{N_2}{n_2}$)
halstead effort(EFF)	The efforts required to understand or implement the program (D * V)
halstead error(ERR)	The estimated number of errors in the implementation (V/3000)
halstead time(T)	The time required to understand or implement the program (EFF/18)

Need to select some highly correlated features with the software defect

&

Remove some redundant features

Dataset

PROMISE Software Engineering Repository data set

- SoftLab data - a Turkish white-goods manufacturer Embedded software implemented in C
- Measured by McCabe & Halstead metrics

- 428 modules(observation), 29 features(predictor)
- Binary response - defective(1) / defect-free(0)

Feature network

To measure the correlation between features

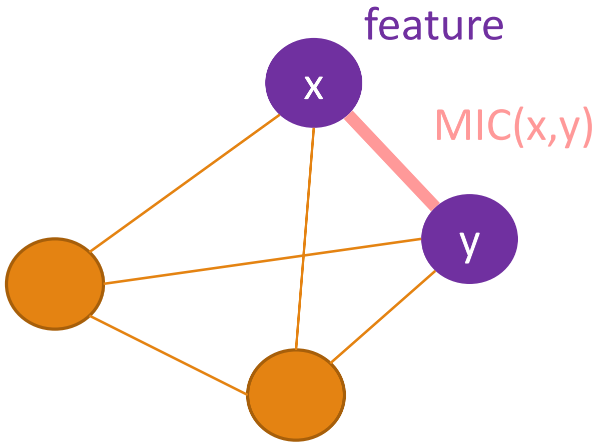
MIC(Maximal Information Coefficient) feature network construction

- A measure of the strength of the linear or non-linear association between two variables *X* and *Y*
- Binning scheme - apply mutual information on continuous random variables
- Mutual information between x and y

$$I(x, y) = H(x) + H(y) - H(x, y)$$

- Try all the possible binnings and pick the maximum

$$MIC(x, y) = max\left(\frac{I(x, y)}{log_2 min(n_x, n_y)}\right)$$



Feature selection

F-score(Fisher score) based feature selection

- Fisher score of i-th feature

$$F(i) = \frac{\left(\overline{x_i}^{(+)} - \overline{x_i}\right)^2 + \left(\overline{x_i}^{(-)} - \overline{x_i}\right)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}\left(x_{k,i}^{(+)} - \overline{x_i}^{(+)}\right)^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}\left(x_{k,i}^{(-)} - \overline{x_i}^{(-)}\right)^2}$$

n_+ , n_- : the positive and negative instances, $\overline{x_i}, \overline{x_i}^{(+)}, \overline{x_i}^{(-)}$: the average of the *i*-th feature of the whole, positive, and negative data sets, $x_{k,i}^{(+)}$, $x_{k,i}^{(-)}$: the *i*-th feature of the *k*-th positive/negative instance.

- Measures the discrimination of the feature
- Automatically finds the feature subset with high discrimination
- Don’t reveal the mutual information between features

Feature selection

k-means clustering

- Given the p-by-n data matrix (p predictors, n modules),
- Select k mutually exclusive clusters from p predictors
- The standard k-means algorithm, Lloyd’s algorithm &
  k-means++ algorithm for the centroid initialization

Feature selection

Spectral clustering

- Standard graph cut algorithm
- Uses the spectrum(eigenvalues) of the graph for dimensionality reduction before clustering
- The normalized cuts algorithm
- The affinity matrix

$$W = e^{\left(-\frac{G}{2*\sigma^2}\right)}$$

σ: scaling parameter

- The graph is clustered using eigenvectors with the second smallest eigenvalue solving the symmetric normalized laplacian matrix

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}x = \lambda x$$

x: eigenvectors, λ: eigenvalue

Feature selection

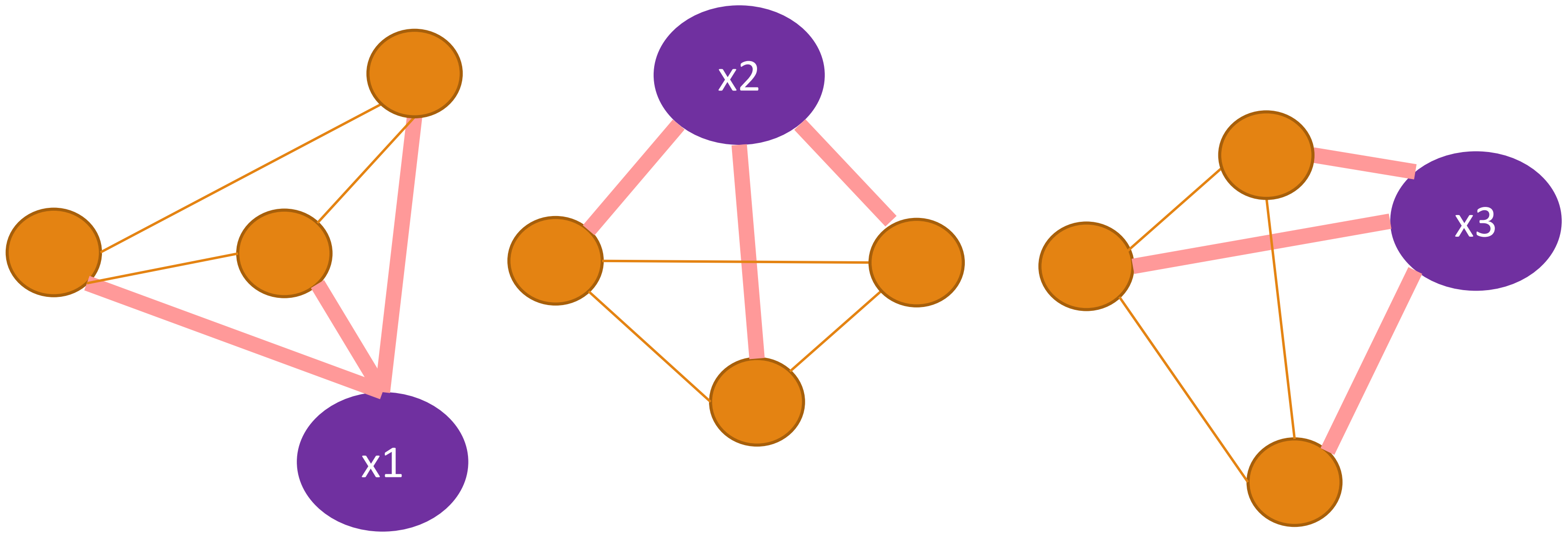
Hierarchical clustering

- Construct clusters from the agglomerative hierarchical cluster tree
- To encode the hierarchical cluster tree, linkage methods are used
  - average = centroid = weighted average > Ward's method
- *k* clusters are obtained by cutting off the hierarchical tree at the smallest height

Feature selection

Given network or data matrix,

- Cluster into 3 feature groups,
- Select 3 features which have the highest value of averaged edge weights within each cluster



Fault classification

SVM classification

- The prediction model is trained with the selected features
- Kernel : linear > polynomial, RBF, sigmoid

Defect prediction on validation set

- Perform 5-fold cross validation
  - determine optimal parameter, avoid over-fitting to the training data
- 80% (≈ 342) for training, 20% (≈ 85) for test
- Evaluate the performance with the averaged 5-fold cross validation accuracy

Results

Averaged 5-fold cross validation accuracy comparison

Method	Cross Validation Accuracy
Whole	78.27%
FFS	86.21%
K-means	85.75%
Spectral	87.85%
Hierarchical	86.68%

Results

Feature	Description
total loc	The total lines of code
blank loc	Lines of blanks
comment loc	Lines of comments
code and comment loc	Lines
executable loc	The executable source lines of code
branch count	Branch count of the flow graph
decision count	Decision count
call pairs	Executable call pairs between modules
condition count	Condition count
multiple condition count	Multiple condition count
cyclomatic complexity	Cyclomatic complexity
cyclomatic density	Cyclomatic density (cyc the lines of code)
decision density	Decision density (condition decision metric divided by the decision count)
design complexity	Design complexity (the number of paths which calls something in the control flow)
design density	Design density (design complexity divided by cyclomatic complexity)
normalized cyclomatic complexity	Normalized cyclomatic complexity
formal parameters	The number of formal parameters

F-score based feature selection

K-means clustering

Feature	Description
unique operands(n_2)	The number of unique operands
unique operators(n_1)	The number of unique operators
total operands(N_2)	The number of operands
total operators(N_1)	The number of operators
halstead vocabulary(n)	The length of unique operands and operators ($n_1 + n_2$)
halstead length(N)	The length of operands and operators ($N_1 + N_2$)
halstead volume(V)	The measure of complexity ($N * \log_2 n$)
halstead level(L)	The implementation level of the program ($\frac{2*n_2}{n_1*N_2}$)
halstead difficulty(D)	The measure of difficulty ($\frac{n_1}{2} * \frac{N_2}{n_2}$)
halstead effort(EFF)	The efforts required to understand or implement the program (D * V)
halstead error(ERR)	The estimated number of errors in the implementation (V/3000)
halstead time(T)	The time required to understand or implement the program (EFF/18)

Results

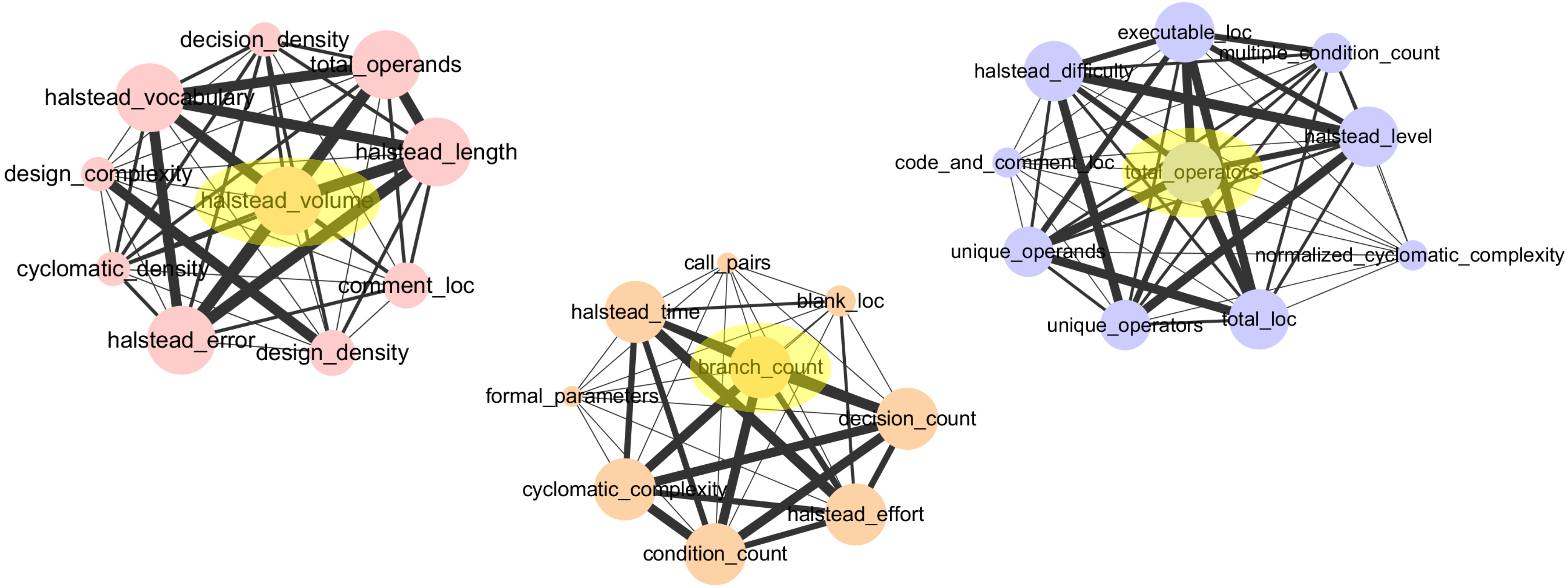
Feature	Description
total loc	The total lines of code
blank loc	Lines of blanks
comment loc	Lines of comments
code and comment loc	Lines of code and comments
executable loc	The executable source lines of code
branch count	Branch count of the flow graph
decision count	Decision count
call pairs	Executable call pairs between modules
condition count	Condition count
multiple condition count	Multiple condition count
cyclomatic complexity	Cyclomatic complexity
cyclomatic density	Cyclomatic … ed by the lines o…
decision density	Decision density (condition decision metric divided by the decision count)
design complexity	Design complexity (the number of paths which calls something in the control flow)
design density	Design density (design complexity divided by cyclomatic complexity)
normalized cyclomatic complexity	Normalized cyclomatic complexity
formal parameters	The number of formal parameters

Spectral clustering

Hierarchical clustering

Feature	Description
unique operands(n_2)	The number of unique operands
unique operators(n_1)	The number of unique operators
total operands(N_2)	The number of operands
total operators(N_1)	The number of operators
halstead vocabulary(n)	The length of unique operands and operators ($n_1 + n_2$)
halstead length(N)	The length of operands and operators ($N_1 + N_2$)
halstead volume(V)	The measure of complexity (N * $\log_2 n$)
halstead level(L)	The implementation level of the program ($\frac{2*n_2}{n_1*N_2}$)
halstead difficulty(D)	The measure of difficulty ($\frac{n_1}{2} * \frac{N_2}{n_2}$)
halstead effort(EFF)	The efforts required to understand or implement the program (D * V)
halstead error(ERR)	The estimated number of errors in the implementation (V/3000)
halstead time(T)	The time required to understand or implement the program (EFF/18)

Results


decision_density
total_operands
halstead_vocabulary
halstead_length
design_complexity
halstead_volume
cyclomatic_density
comment_loc
halstead_error
design_density
executable_loc
multiple_condition_count
halstead_difficulty
halstead_level
code_and_comment_loc
total_operators
unique_operands
normalized_cyclomatic_complexity
unique_operators
total_loc
call_pairs
blank_loc
halstead_time
branch_count
formal_parameters
decision_count
cyclomatic_complexity
halstead_effort
condition_count

Thanks !

Q & A