Knowledge-based Systems for Bioinformatics

Lecture 5: Decision trees and Monte Carlo Feature Selection

Senior Professor **Jan Komorowski**

Computational Biology and Bioinformatics Program,
Department of Cell and Molecular Biology, Science for Life Laboratory, Uppsala University
jan.komorowski@icm.uu.se

Winter 2026

UPPSALA
UNIVERSITET

Agenda

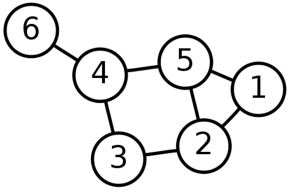
Overview

- Decision trees are also transparent/interpretable
- Rough-Sets (RS) cannot deal with very large numbers of features (finding all reducts is NP-hard); we introduce a Monte Carlo-based method to handle such cases
- We will also look into dependencies between features
- And, finally, we will start the art of interpreting transparent models

Some observations about RS

- The features are treated equally. Is this always the case?
- The values of features are discrete. Can you comment on that? Pros and cons

Graphs

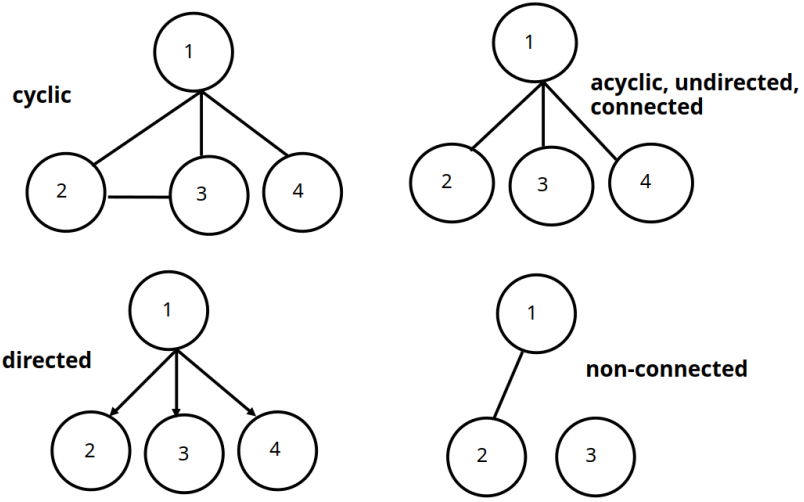


- The edges of a graph define a symmetric relation on the vertices, called the adjacency relation.
- Two vertices x and y are adjacent if $\{x, y\}$ is an edge.
- A graph is fully determined by its adjacency matrix A, which is an n × n square matrix, with A_{ij} specifying the number of connections from vertex i to vertex j.

	1	2	3	4	5	6
1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	1	0	0
4	0	0	1	0	1	1
5	1	1	0	1	0	0
6	0	0	0	1	0	0

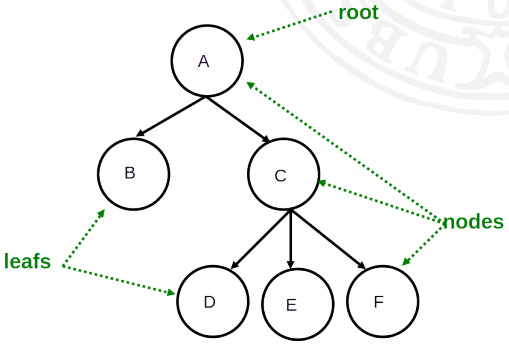
Adjacency Matrix A

Properties of Graphs



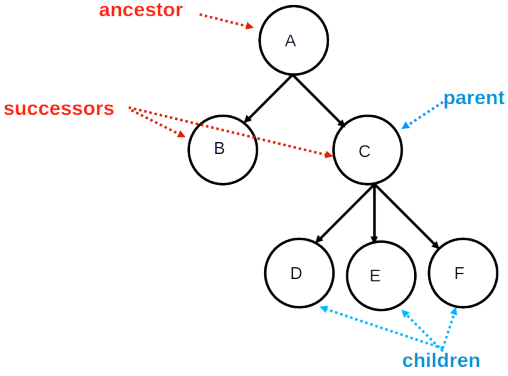
A directed tree

- A connected, acyclic and directed graph



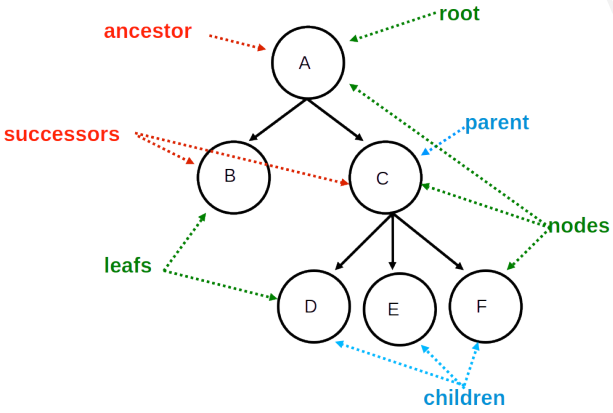
A directed tree

- A connected, acyclic and directed graph

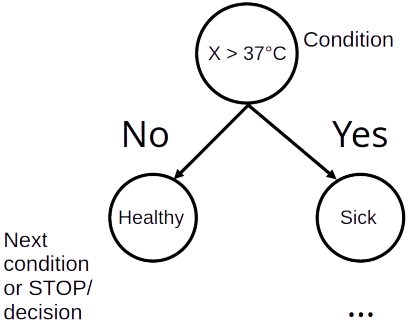


A directed tree

- A connected, acyclic and directed graph

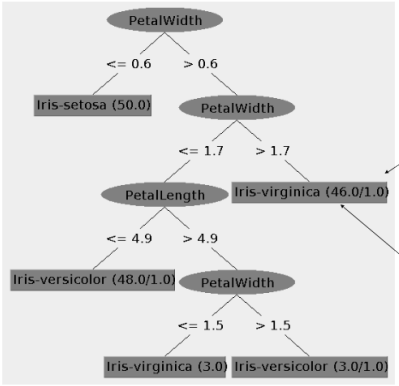


DT's can naturally be used to support decisions



A directed tree

- A decision tree is represented by a directed tree
- An example of a tree, created on the iris data using WEKA



# of instances incorrectly classified in this node

Total # of instances classified in this node

Tree construction

- Assume that we would like to classify proteins as modified or non-modified based on certain properties.
  - Collect data for known proteins.
  - Train a classifier.
- Many different algorithms, this time using decision trees.
- Top-down approach: start by taking the "best" attribute to split on the root node.
- What is "the best"?

Size	Charge	Polarity	Aromatic	Modified
small	neutral	high	FALSE	no
large	neutral	high	TRUE	no
small	positive	high	FALSE	no
small	positive	high	TRUE	no
large	negative	low	TRUE	no
medium	negative	low	TRUE	yes
small	neutral	low	TRUE	yes
medium	neutral	high	TRUE	yes
medium	positive	high	FALSE	yes
medium	positive	low	FALSE	yes
small	negative	low	FALSE	yes
large	neutral	high	FALSE	yes
large	neutral	low	FALSE	yes
large	negative	low	FALSE	yes

Tree construction

- Information value is defined as
  - $info([x, y]) = -x/n * log_2(x/n) - y/n * log_2(y/n)$
- where x and y are the number of objects from each decision class, and n is the total number of objects (i.e. $n = x + y$)
- Example, for some hypothetical decision table:

  $info([1, 3]) = -1/4 * log_2(1/4) - 3/4 * log_2(3/4) = 0.811 bits$
  $info([2, 2]) = -2/4 * log_2(2/4) - 2/4 * log_2(2/4) = 1 bits$
- If $x = n$ or $y = n$ then the outcome is almost certain and the information value is close to 0

  $info([0, 4]) = -0/4 - 4/4 * log_2(4/4) = 0 bits$

Tree construction

- Strategy:
  - Calculate the *information value* for the whole data set
  - Consider each possible split, and calculate the *information value* for the resulting data sets
  - Choose the split that reduces the *information value* the most

Tree construction

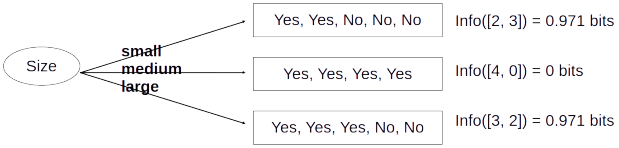
- Recall the definition:
  - $info([x, y]) = -x/n * log_2(x/n)y/n * log_2(y/n)$
- There are 14 objects in the data set: 9 with decision attribute "yes" and 5 with decision attribute "no".
- The information value for the data set is ($n = 14, x = 9, y = 5$):
  - $info([9, 5]) = 9/14 * log_2(9/14)5/14 * log_2(5/14) =$ ***0.940 bits***

Size	Charge	Polarity	Aromatic	Modified
small	neutral	high	FALSE	no
large	neutral	high	TRUE	no
small	positive	high	FALSE	no
small	positive	high	TRUE	no
large	negative	low	TRUE	no
medium	negative	low	TRUE	yes
small	neutral	low	TRUE	yes
medium	neutral	high	TRUE	yes
medium	positive	high	FALSE	yes
medium	positive	low	FALSE	yes
small	negative	low	FALSE	yes
large	neutral	high	FALSE	yes
large	neutral	low	FALSE	yes
large	negative	low	FALSE	yes

Tree construction

- Now consider the attribute "size". One can split the data on "size" in the following way, giving a slightly more ordered data

Size	Charge	Polarity	Aromatic	Modified
small	neutral	high	FALSE	no
large	neutral	high	TRUE	no
small	positive	high	FALSE	no
small	positive	high	TRUE	no
large	negative	low	TRUE	no
medium	negative	low	TRUE	yes
small	neutral	low	TRUE	yes
medium	neutral	high	TRUE	yes
medium	positive	high	FALSE	yes
medium	positive	low	FALSE	yes
small	negative	low	FALSE	yes
large	neutral	high	FALSE	yes
large	neutral	low	FALSE	yes
large	negative	low	FALSE	yes



Tree construction

- A weighted average of these three information values may be calculated:

$$\text{info}([2,3],[0,4],[3,2]) =$$
$$(5/14) * \textit{info}([2,3]) + (4/14) * \textit{info}([0,4]) + (5/14) * \textit{info}([3,2]) =$$
$$(5/14) * 0.971 \quad + \quad (4/14) * 0 \quad + (5/14) * 0.971 = \textbf{\textit{0.693 bits.}}$$

- Now, recall that the information before the split was 0.940 bits. We may calculate how much information we gain by the split on "size" as:
  - $\textit{gain}(\textit{size}) = \textit{info}([9,5]) - \textit{info}([2,3],[0,4],[3,2]) = 0.940 - 0.693 = \textbf{\textit{0.247 bits.}}$

Size	Charge	Polarity	Aromatic	Modified
small	neutral	high	FALSE	no
large	neutral	high	TRUE	no
small	positive	high	FALSE	no
small	positive	high	TRUE	no
large	negative	low	TRUE	no
medium	negative	low	TRUE	yes
small	neutral	low	TRUE	yes
medium	neutral	high	TRUE	yes
medium	positive	high	FALSE	yes
medium	positive	low	FALSE	yes
small	negative	low	FALSE	yes
large	neutral	high	FALSE	yes
large	neutral	low	FALSE	yes
large	negative	low	FALSE	yes

Tree construction

- This information gain is however biased towards attributes with many values. In the extreme case, if each instance has a unique ID, this ID will be used to construct the best and the only split.
- To account for this, the information gain ratio is calculated by dividing the information gain by the split information value:
  - $gain\_ratio(size) = gain(size)/split\_info(size)$

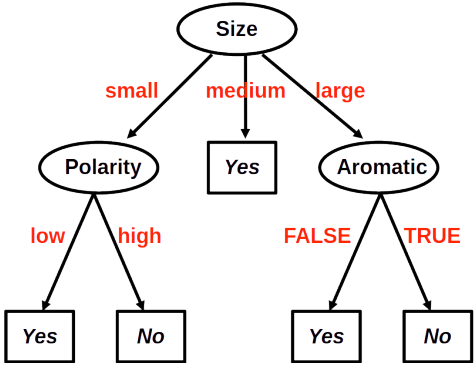
Size	Charge	Polarity	Aromatic	Modified
small	neutral	high	FALSE	no
large	neutral	high	TRUE	no
small	positive	high	FALSE	no
small	positive	high	TRUE	no
large	negative	low	TRUE	no
medium	negative	low	TRUE	yes
small	neutral	low	TRUE	yes
medium	neutral	high	TRUE	yes
medium	positive	high	FALSE	yes
medium	positive	low	FALSE	yes
small	negative	low	FALSE	yes
large	neutral	high	FALSE	yes
large	neutral	low	FALSE	yes
large	negative	low	FALSE	yes

Tree construction

- To calculate the split information value, the decision class is ignored, and we only notice that splitting on the attribute "size" will result in three branches, containing 5, 4 and 5 objects each.
  - $split\_info(size) = info([5,4,5]) = info([5,9]) + 9/14 x info([4,5]) =$ ***1.577 bits.***
- The information gain ratio is then calculated as:
  - $gain\_ratio(size) = gain(size)/split\_info(size) = 0.247/1.577 =$ ***0.157 bits.***
- The attribute with the highest information gain ratio will be used to construct the first node.
- The procedure is then repeated until all leaf nodes contain only one class, or cannot be further divided.

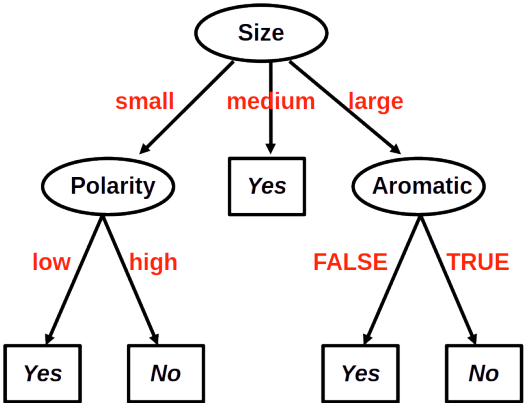
Size	Charge	Polarity	Aromatic	Modified
small	neutral	high	FALSE	no
large	neutral	high	TRUE	no
small	positive	high	FALSE	no
small	positive	high	TRUE	no
large	negative	low	TRUE	no
medium	negative	low	TRUE	yes
small	neutral	low	TRUE	yes
medium	neutral	high	TRUE	yes
medium	positive	high	FALSE	yes
medium	positive	low	FALSE	yes
small	negative	low	FALSE	yes
large	neutral	high	FALSE	yes
large	neutral	low	FALSE	yes
large	negative	low	FALSE	yes

Tree construction



Size	Charge	Polarity	Aromatic	Modified
small	neutral	high	FALSE	no
large	neutral	high	TRUE	no
small	positive	high	FALSE	no
small	positive	high	TRUE	no
large	negative	low	TRUE	no
medium	negative	low	TRUE	yes
small	neutral	low	TRUE	yes
medium	neutral	high	TRUE	yes
medium	positive	high	FALSE	yes
medium	positive	low	FALSE	yes
small	negative	low	FALSE	yes
large	neutral	high	FALSE	yes
large	neutral	low	FALSE	yes
large	negative	low	FALSE	yes

DT's and RBM are two sides of the same coin!



Similar to rules, e.g:

IF *Size*=large **AND** *Aromatic*=FALSE **THEN** *modified*=Yes

Rule	Length	Accuracy	Support	pValue
IF Polarity(low) THEN yes	1	1	7	0.2797203
IF Size(small) AND Polarity(high) THEN no	2	1	3	0.3296703
IF Size(small) AND Charge(positive) THEN no	2	1	2	1.0000000
IF Size(large) AND Aromatic(TRUE) THEN no	2	1	2	1.0000000
IF Size(medium) THEN yes	1	1	4	1.0000000
IF Size(small) AND Polarity(low) THEN yes	2	1	2	1.0000000
IF Size(small) AND Charge(negative) THEN yes	2	1	1	1.0000000
IF Size(large) AND Charge(neutral) THEN yes	2	1	2	1.0000000
IF Size(large) AND Aromatic(FALSE) THEN yes	2	1	3	1.0000000
IF Charge(neutral) AND Polarity(low) THEN yes	2	1	2	1.0000000
IF Size(large) AND Polarity(high) THEN no	2	1	1	1.0000000
IF Charge(negative) THEN yes	1	1	4	1.0000000
IF Size(small) AND Charge(neutral) THEN yes	2	1	1	1.0000000
IF Size(small) AND Charge(neutral) THEN no	2	1	1	1.0000000
IF Size(large) AND Charge(negative) THEN no	2	1	1	1.0000000

Pruning the tree

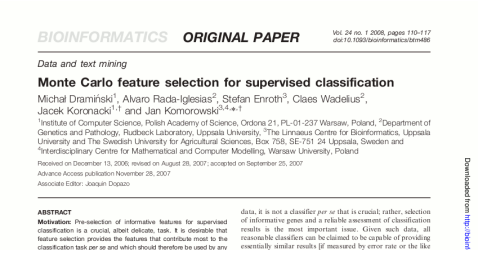
- The tree is grown to the maximum size using a training set
- The last nodes will contain very few examples, and possibly be overfitted to the training data. To account for this, pruning is used.
- Remove nodes that worsen the results on the test set

Summary of decision trees

- A method for classification
- Similar to rule-based classifiers
- Easy to interpret
- Easy to implement
- Computationally cheap
- Risk for overfitting; pruning the tree is needed
- Not always the best performance

Monte Carlo Feature Selection (MCFS)

- Biological data often contains few objects and thousands of features (e.g. data from micro arrays or next-generation sequencing)
- To choose the best features for the classifier can be more important than which classifier that is applied
- MCFS is an algorithm for feature selection and ranking



BIOINFORMATICS **ORIGINAL PAPER**

Vol. 24 no. 1 2008, pages 110–117
doi:10.1093/bioinformatics/btm486

Data and text mining

Monte Carlo feature selection for supervised classification

Michał Draminski[1], Alvaro Rada-Iglesias[2], Stefan Enroth[3], Claes Wadelius[2], Jacek Koronacki[1,†] and Jan Komorowski[3,4,*,†]

[1]Institute of Computer Science, Polish Academy of Science, Ordona 21, PL-01-237 Warsaw, Poland, [2]Department of Genetics and Pathology, Rudbeck Laboratory, Uppsala University, [3]The Linnaeus Centre for Bioinformatics, Uppsala University and The Swedish University for Agricultural Sciences, Box 758, SE-751 24 Uppsala, Sweden and [4]Interdisciplinary Centre for Mathematical and Computer Modelling, Warsaw University, Poland

Received on December 13, 2006; revised on August 28, 2007; accepted on September 25, 2007

Advance Access publication November 28, 2007

Associate Editor: Joaquin Dopazo

ABSTRACT

Motivation: Pre-selection of informative features for supervised classification is a crucial, albeit delicate, task. It is desirable that feature selection provides the features that contribute most to the classification task *per se* and which should therefore be used by any data, it is not a classifier *per se* that is crucial; rather, selection of informative genes and a reliable assessment of classification results is the most important issue. Given such data, all reasonable classifiers can be claimed to be capable of providing essentially similar results [if measured by error rate or the like

Journal of Statistical Software

July 2018, Volume 85, Issue 12. doi: 10.18637/jss.v085.i12

rmcfs: An R Package for Monte Carlo Feature Selection and Interdependency Discovery

Michał Dramiński
IPI PAN

Jacek Koronacki
IPI PAN

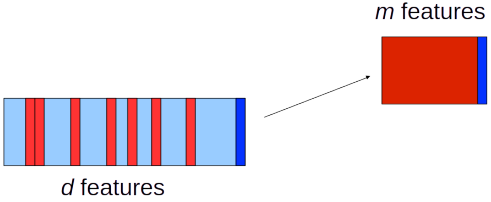
MCFS algorithm

- Conceptually simple, but computer-intensive
- Create a large number of classification trees for different subsets of the features and objects
- A feature is considered to be important, or informative, if it is likely to take part in the process of classifying samples into classes "more often than not"
- Idea:
  - If the feature ***aromatic*** is part of 1000 trees and the feature ***charge*** is part of 700 trees, then ***aromatic*** is probably more important for classification than ***charge***

Monte Carlo Feature Selection

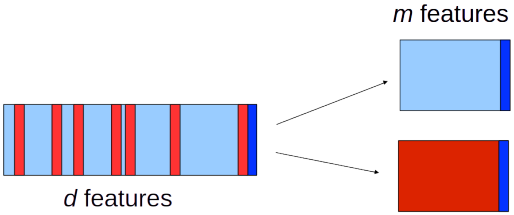
d features

Monte Carlo Feature Selection

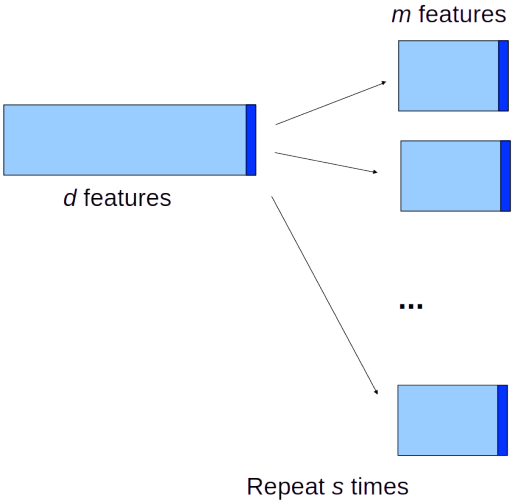


Select $m << d$ features at random

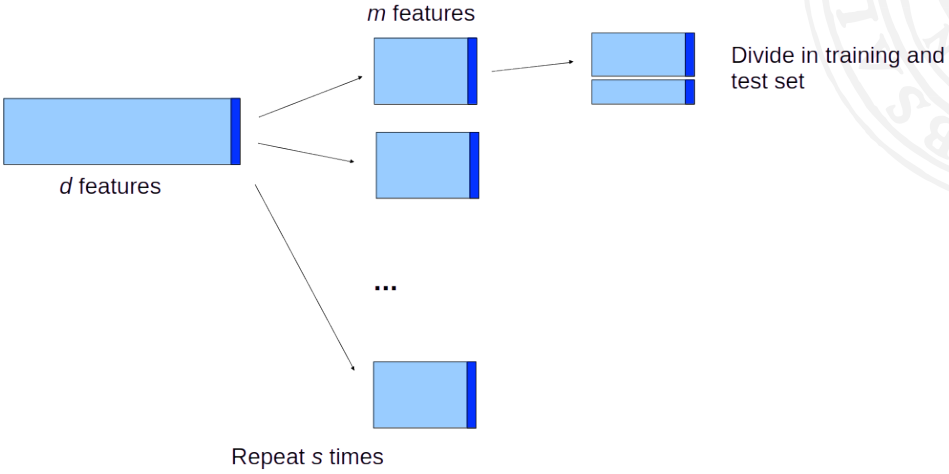
Monte Carlo Feature Selection



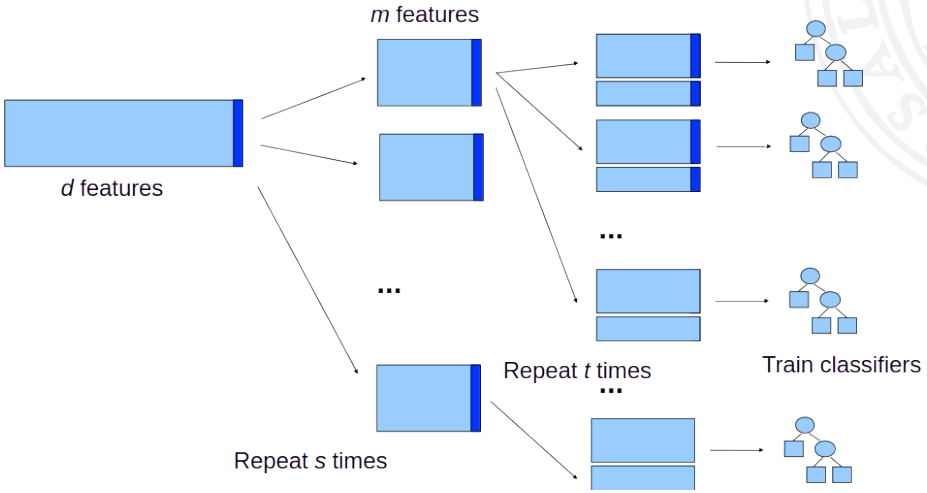
Monte Carlo Feature Selection



Monte Carlo Feature Selection

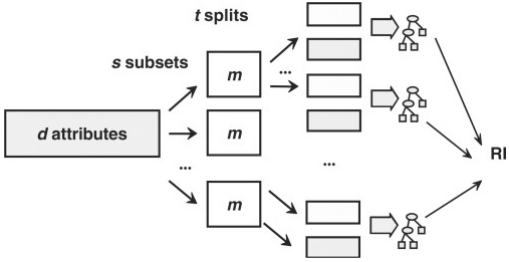


Monte Carlo Feature Selection



MCFS algorithm

- Out of all d features, s subsets of m randomly chosen features are created $m << d$
- For each subset of features the objects are randomly divided in t training and test sets (0.66/0.34).
- For each training set, a tree classifier is constructed and its performance is assessed
- A relative importance (RI) value is calculated for each feature



Weighted Accuracy

- To determine relative importance, weighted accuracy of a tree is introduced as a means to assess the classification ability of the tree on a test set.
- For a classification problem with c classes, let n_{ij} denote the number of samples from class i classified as those from class j; clearly, $i, j = 1, 2, \ldots, c$.

$$\sum_{ij} n_{ij} = n$$

- Now, one can define weighted accuracy as:

$$wAcc = \frac{1}{c} \sum_{i=1}^{n} \frac{n_{ii}}{n_{ii} + n_{12} + \cdots + n_{ic}}$$

- i.e., as the mean of c true positive rates.

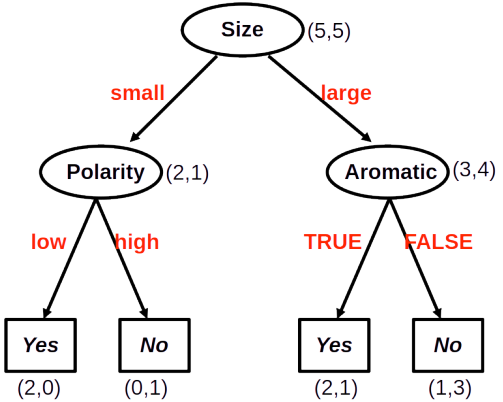
Calculation of RI

- The relative importance of a feature g_k is calculated as:

$$RI_{g_k} = \sum_{\tau=1}^{s \cdot t} \underbrace{(wAcc_\tau)^u}_{\text{Weighted accuracy of the tree } \tau} \cdot \sum_{n_{g_k}(\tau)} \underbrace{IG(n_{g_k}(\tau))}_{\text{Information gain in node } n_{g_k}} \cdot \underbrace{\left(\frac{\text{no. in } n_{g_k}(\tau)}{\text{no. in } \tau} \right)^v}_{\text{Fraction of objects in node } n_{g_k} \text{ compared to in the root}}$$

- A summation over all st trees and, within each tree, τ, over all nodes $n_{g_k}(\tau)$ of that tree on which the split is made on feature g_k.
  $IG(n_{g_k}(\tau))$ stands for information gain for node $n_{g_k}(\tau)$, (no. in $n_{g_k}(\tau)$) denotes the number of samples in node $n_{g_k}(\tau)$, (no. in τ) denotes the number of samples in the root of the tree τ, and u and v are fixed positive reals.

What is the contribution to RI Polarity?



$$RI_{g_k} = \sum_{\tau=1}^{s \cdot t} (wAcc_\tau)^u \cdot \sum_{n_{g_k}(\tau)} IG(n_{g_k}(\tau)) \cdot \left(\frac{\text{no. in } n_{g_k}(\tau)}{\text{no. in } \tau} \right)^v$$

Parameters

- In the procedure, there are five parameters, m, s, t, u, and v, to be set by an experimenter.
- For a given m, s is made a running parameter of the procedure, and the procedure is executed for $s = s_1, s_1 + 10, s_1 + 20, \ldots$ until the rankings for successive values of s of top 5–20% features prove (almost) the same.

Parameter	Description	Typical Value
d	Number of attributes (features) in data set	-
m	Number of attributes in each subset	$\sqrt{d}$ or $0.1d$
s	Number of subsets (projections)	see below
t	Number of trees constructed for each subset	5–10
u	Weighting parameter (for wAcc)	0, 0.5, 1, or 2
v	Weighting parameter (for number of samples)	0, 0.5, 1, or 2

Check for Significant RI

- To check which parameters have significantly high RI, randomization is used:
  1. Permute the decision class randomly.
  2. Repeat the whole procedure of MCFS and save the highest RI.
  3. Repeat steps 1–2 at least 20 times.
  4. If the random values are normally distributed, estimate the 95% confidence interval for the normal distribution fitted to the set of values.
  5. Features with RI values above the 95% interval are considered significantly important ($p \leq 0.05$).
- There is a parameter for the number of permutations in the procedure.

r.mcfs

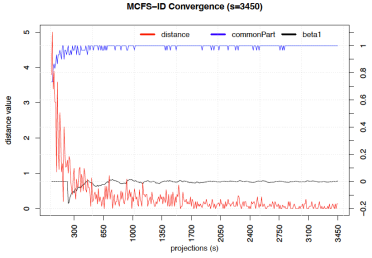


Figure 2: Distance function and common part.

- The distance function shows the difference between two consecutive rankings: zero means no changes between two rankings (see the left *y*-axis).
- The common part gives the fraction of features that overlap for two different rankings (see the right *y*-axis).
- The ranking stabilizes after a number of iterations: the distance tends to zero and the common part tends to 1.
- beta1 shows the slope of the tangent of a smoothed distance function.
- If beta1 tends to 0 (the right *y*-axis) then the distance is given by a flat line.

MCFS–ID Convergence (s=3450)

An example and the algorithm

- BReast invasive CArcinoma (BRCA) and LUng ADenocarcinoma (LUAD)
- Developing a classifier
- Interpreting the classifier

The need for finer methods

- Questions:
  - which genes discern between two (or more) types of cancer?
  - when the cancers are indiscernible
- Help interpret the models

Aims

- Find genes that not only discern (classify, distinguish) BReast invasive CArcinoma (BRCA) from LUng ADenocarcinoma (LUAD) but also show how they collaborate, or interact, towards the outcome
- Find similarities between BRCA and LUAD

Establishing a Common Language

- How do we find the features (i.e. genes) that contribute to discerning BRCA-LUAD?
  - Firstly, find the genes that discern the tissues
  - How?
  - Use the GTEx portal of tissue and cell-specific gene expressions https://www.gtexportal.org/home/

The GTEx Portal

Finding the tissue specific genes

- Feature selection on the normal-normal tissue data set gave 310 tissue-specific genes
- The 310 genes were from the BRCA+LUAD data set
- The intersection of the BRCA+LUAD genes with the genes in the Normal+Normal tissue data set resulted in 17928 genes

MCFS top 10 genes for Normal-Normal

- The most significant genes for classifying normal-normal tissues (breast - lung)
- All significant features (i.e. 310 genes) removed from the BRCA-LUAD

Position	Attribute	RL_norm
1	CD5L	0.87065053
2	CTSE	0.86358388
3	WNT7A	0.8488115
4	PTCRA	0.8396277
5	ABCC13	0.82558876
6	HOXA9	0.8175331
7	GGTLC1	0.79210943
8	VSTM1	0.7649897
9	KHDRBS2	0.74836046
10	AQP10	0.7464942

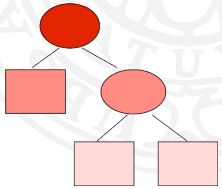
MCFS top 10 genes for Normal-Normal

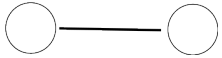
- BRCA-LUAD left with 17928 genes
- MCFS found 603 genes that "collaborate" in discerning between BRCA-LUAD

Position	Attribute	RI_norm
1	SFTA3	0.9608818
2	NKX2-1	0.8805158
3	SCGB2A2	0.69707626
4	SFTPA1	0.6682077
5	SFTPC	0.64998674
6	C20orf56	0.634872
7	APOH	0.5687929
8	EMX2	0.50805193
9	FOXA2	0.4858794
10	HNF1B	0.48406482

A graph of interdependencies between the genes

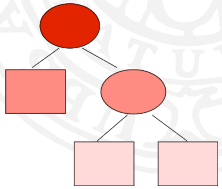
- 610 genes collaborate in classifying BRCA-LUAD.
- Heuristics for finding interdependencies:
  - When two features co-occur in a decision tree, they are interdependent; put a link in the ID-graph.
  - The higher the feature (i.e., gene) in the decision tree, the stronger it contributes.
  - The more often different pairs with the same feature occur, the larger the size of the node.
  - The more often the pair occurs in the trees, the thicker the connection is.
  - The direction of the links follows the depth in the tree: from higher to lower.

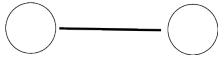




A graph of interdependencies between the genes

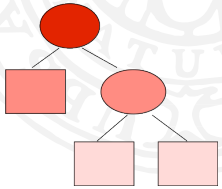
- 610 genes collaborate in classifying BRCA-LUAD.
- Heuristics for finding interdependencies:
  - When two features co-occur in a decision tree, they are interdependent; put a link in the ID-graph.
  - The higher the feature (i.e., gene) in the decision tree, the stronger it contributes.
  - The more often different pairs with the same feature occur, the larger the size of the node.
  - The more often the pair occurs in the trees, the thicker the connection is.
  - The direction of the links follows the depth in the tree: from higher to lower.

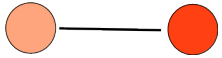




A graph of interdependencies between the genes

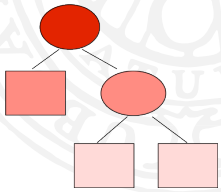
- 610 genes collaborate in classifying BRCA-LUAD.
- Heuristics for finding interdependencies:
  - When two features co-occur in a decision tree, they are interdependent; put a link in the ID-graph.
  - The higher the feature (i.e., gene) in the decision tree, the stronger it contributes.
  - The more often different pairs with the same feature occur, the larger the size of the node.
  - The more often the pair occurs in the trees, the thicker the connection is.
  - The direction of the links follows the depth in the tree: from higher to lower.

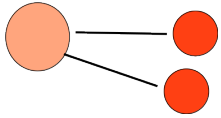




A graph of interdependencies between the genes

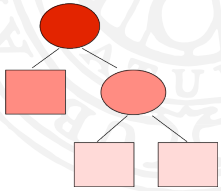
- 610 genes collaborate in classifying BRCA-LUAD.
- Heuristics for finding interdependencies:
  - When two features co-occur in a decision tree, they are interdependent; put a link in the ID-graph.
  - The higher the feature (i.e., gene) in the decision tree, the stronger it contributes.
  - The more often different pairs with the same feature occur, the larger the size of the node.
  - The more often the pair occurs in the trees, the thicker the connection is.
  - The direction of the links follows the depth in the tree: from higher to lower.

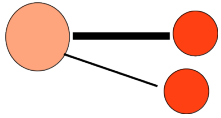




A graph of interdependencies between the genes

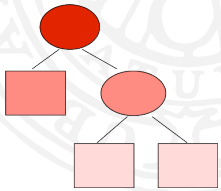
- 610 genes collaborate in classifying BRCA-LUAD.
- Heuristics for finding interdependencies:
  - When two features co-occur in a decision tree, they are interdependent; put a link in the ID-graph.
  - The higher the feature (i.e., gene) in the decision tree, the stronger it contributes.
  - The more often different pairs with the same feature occur, the larger the size of the node.
  - The more often the pair occurs in the trees, the thicker the connection is.
  - The direction of the links follows the depth in the tree: from higher to lower.

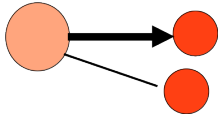




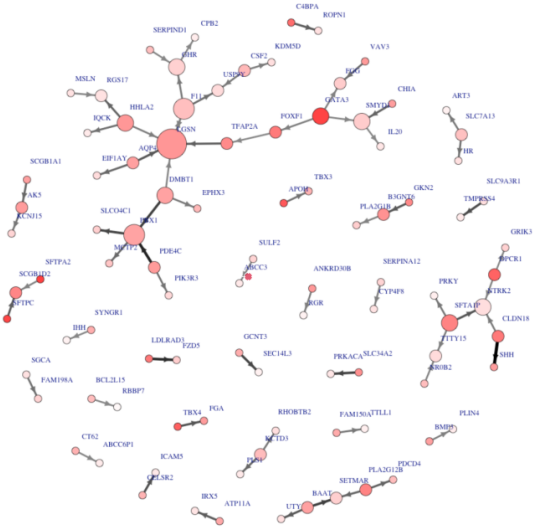
A graph of interdependencies between the genes

- 610 genes collaborate in classifying BRCA-LUAD.
- Heuristics for finding interdependencies:
  - When two features co-occur in a decision tree, they are interdependent; put a link in the ID-graph.
  - The higher the feature (i.e., gene) in the decision tree, the stronger it contributes.
  - The more often different pairs with the same feature occur, the larger the size of the node.
  - The more often the pair occurs in the trees, the thicker the connection is.
  - The direction of the links follows the depth in the tree: from higher to lower.

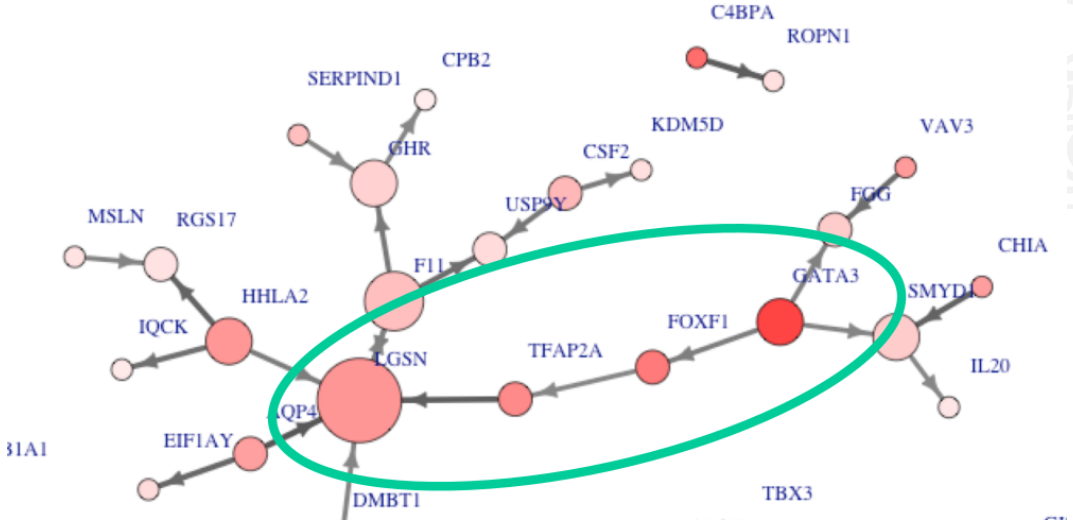




The InterDependency graph for BRCA-LUAD



The GATA3-FOXF1-TFAP2A-LGSN subgraph



Preliminary Validation

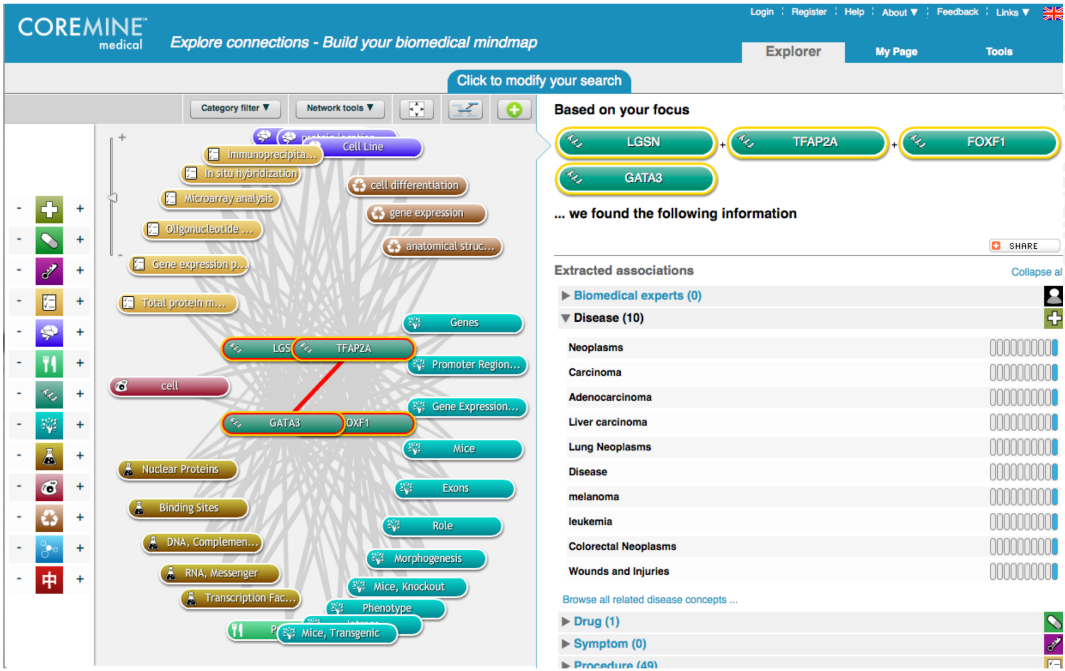
- Perform experiments
- Use an external validation set

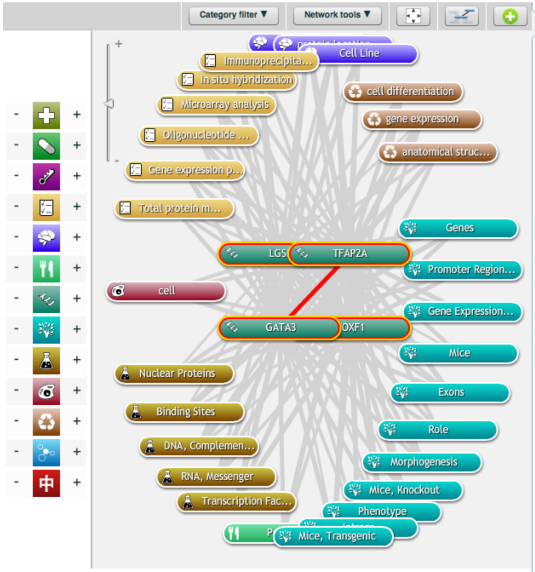
And/Or:

- Perform experiments
- Retrieve annotations from Gene Ontology, Coremine, GeneCards, Ensembl, USCG browser and many others

Gene Annotations

Annotation (GO)	**GATA3**	**FOXF1**	**TFAP2A**	**LGSN**
transcription from RNA polymerase II promoter	yes	yes	yes	
eye development		yes	yes	yes
cell differentiation	yes	yes	yes	yes
cellular response	yes	yes	yes	
embryonic morphogenesis	yes	yes	yes	
positive regulation of transcription from RNA polymerase II promoter	yes	yes	yes	
epithelial morphogenesis	yes	yes		
positive regulation of transcription, DNA-templated	yes	yes	yes	
regulation of cell migration	yes	yes	yes	


CORE MINE medical
Explore connections - Build your biomedical mindmap
Login | Register | Help | About | Feedback | Links
Explorer
My Page
Tools
Click to modify your search
Category filter
Network tools
Based on your focus
LGSN
TFAP2A
FOXF1
GATA3
... we found the following information
SHARE
Extracted associations
Collapse al
Biomedical experts (0)
Disease (10)
Neoplasms
Carcinoma
Adenocarcinoma
Liver carcinoma
Lung Neoplasms
Disease
melanoma
leukemia
Colorectal Neoplasms
Wounds and Injuries
Browse all related disease concepts ...
Drug (1)
Symptom (0)
Cell Line
Immunoprecipita...
In situ hybridization
Microarray analysis
Oligonucleotide ...
Gene expression p...
Total protein m...
cell differentiation
gene expression
anatomical struc...
Genes
Promoter Region...
Gene Expression...
Mice
Exons
Role
Morphogenesis
Mice, Knockout
Phenotype
Mice, Transgenic
Nuclear Proteins
Binding Sites
DNA, Complemen...
RNA, Messenger
Transcription Fac...
cell



FOXF1 (gene/protein, Human) (1650 connections)
GATA3 (gene/protein, Human) (7936 connections)
TFAP2A (gene/protein, Human) (9085 connections)
LGSN (gene/protein, Human) (522 connections)

LGSN + TFAP2A + FOXF1 + GATA3

... we found the following information

SHARE

Extracted associations Collapse all

- ▶ Biomedical experts (0)
- ▶ Disease (10)
- ▶ Drug (1)
- ▶ Symptom (0)
- ▶ Procedure (49)
- ▶ Anatomy (35)
- ▶ Food (2)
- ▶ Gene/Protein (1)
- ▶ MeSH (85)
- ▶ Chemical (27)
- ▶ Cellular component (6)
- ▼ **Biological process (10)**

cell differentiation	
gene expression	
anatomical structure morphogenesis	
signal transduction	
RNA interference	
cell proliferation	
apoptotic process	
transduction	
translation	
cell communication	

Summary of MCFS

- A method for feature selection and ranking
- Conceptually simple
- Computationally expensive
- Good for identifying the most important features for classification in very large sets of features
- MCFS-ID adds interdependencies