경희대학교
KYUNG HEE UNIVERSITY

Ubiquitous Computing Laboratory
Kyung Hee University, Korea

PhD Thesis Presentation

Accurate Multi-Criteria Decision Making for Evaluating Classification and Recommendation Algorithms

Rahman Ali

Department of Computer Science and Engineering

Kyung Hee University, South Korea

Email: rahmanali@oslab.khu.ac.kr

April 25, 2016

Advisor: Prof. Sungyoung Lee, PhD

Agenda

Introduction

- Introduction, background and motivation
- Problem statement
- Taxonomy

Related work

Proposed methodologies

- **Rough Set Classification Model (H2RM):** a model for semantics-preserved accurate classification in real-world applications
- **Hybrid-CBR Model (Hybrid-CBR):** a model for accurate and precise wellness recommendations
- **Accurate multi-criteria decision making (AMD):** a methodology for accurate empirical analysis and evaluation of classifiers

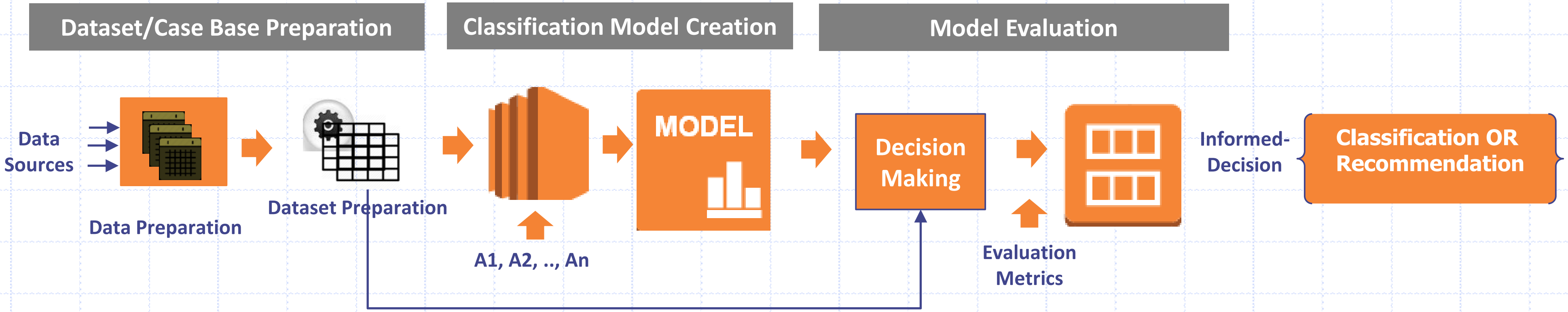
Thesis contributions

Conclusion, future directions, and achievements

경희대학교
KYUNG HEE UNIVERSITY

Introduction

- Organizations make **informed-decisions**, interest of corporates
- **Data mining** processes, managed by **machine learning** technology
- **Classification**, a data mining function, **accurately predicting/decision making** of target decision [1a]
- Correct **data preparation** , model is built right, right decision [1b]
- Data preparation and model creation, as **per domain requirements**

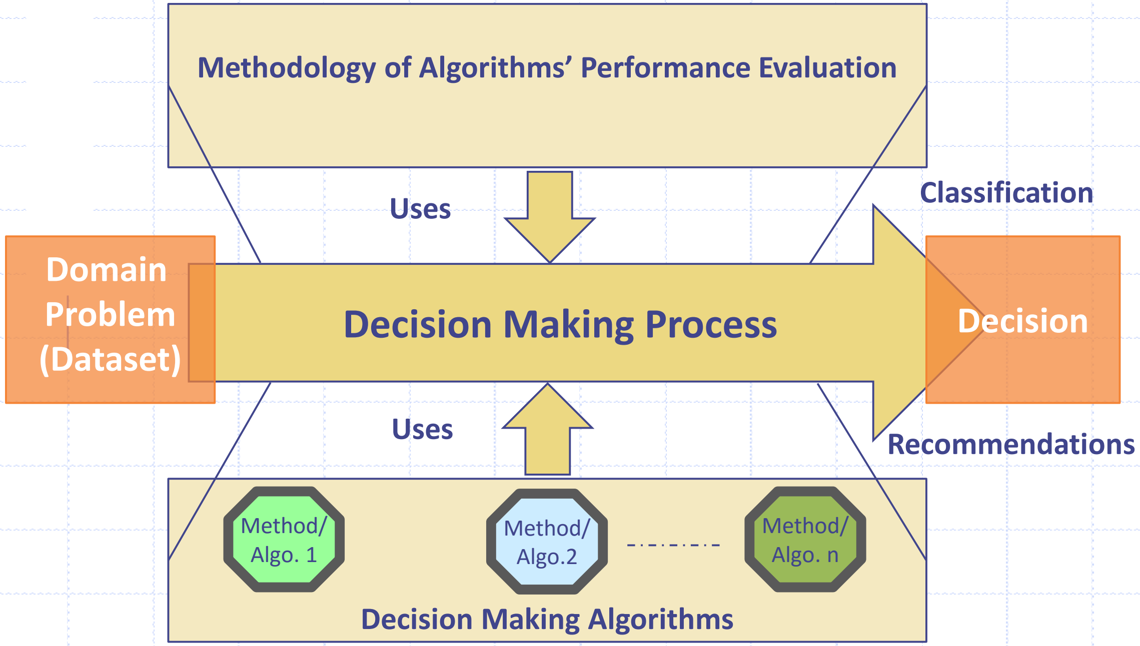


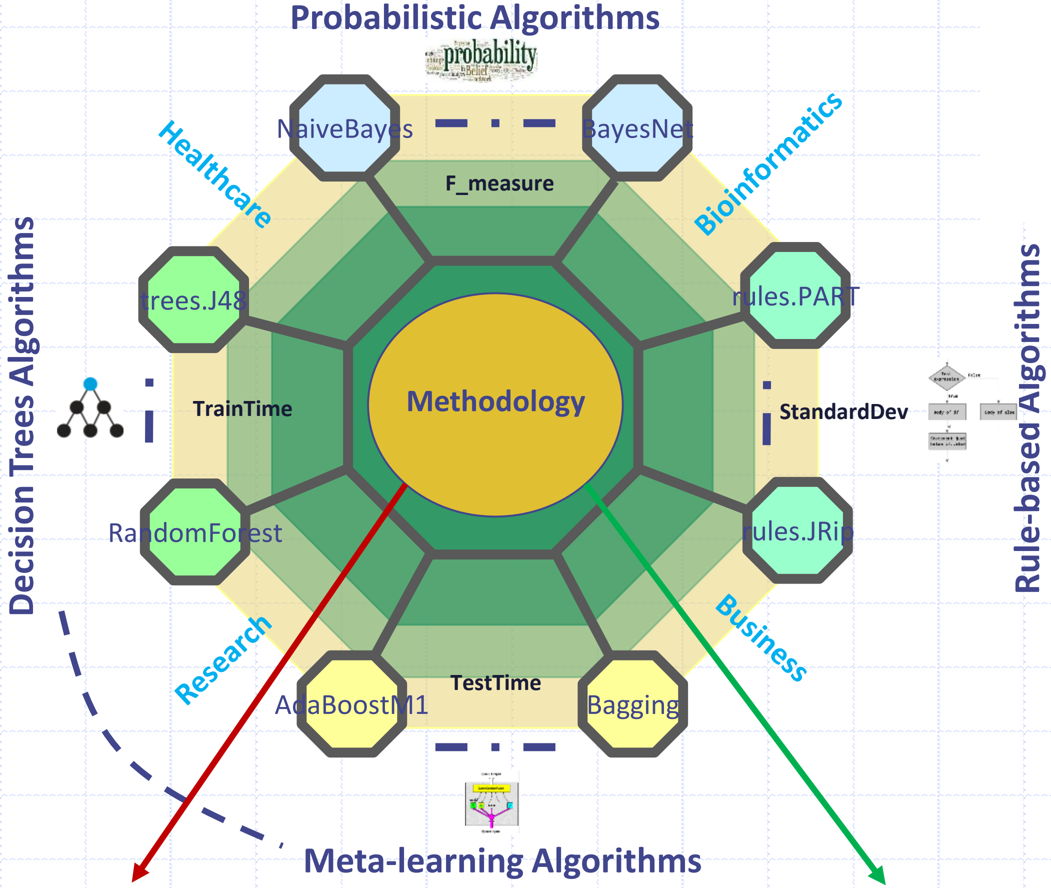
- Available as Rule-based(e.g., Rough Set), Instance-based(e.g., CBR), Meta-learning, Probabilistic, etc.

[1a] https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/classify.htm
[1b] Pyle, D., 1999. Data preparation for data mining (Vol. 1). Morgan Kaufmann.

경희대학교
KYUNG HEE UNIVERSITY

Background and Motivation





- **R** Large number of algorithms available
- **R** Algorithm has capabilities, limitations, & constraints

- **O** Correct data preparation
- **O** Accurate analysis algorithms performance
- **O** Accurate model creation

- **A** Correct dataset
- **A** Suitable algorithm
- **A** Accurate decision

R Reasons **O** Characteristics **A** Advantages

경희대학교
KYUNG HEE UNIVERSITY

Problem Statement

- **Incorrect** classification **decisions** are **drawn** based on **poor** prepared **data** and use of **inappropriate algorithm**– where the decisions have serious implications in real-world applications [1][[2][3]
  - **Classification decision**: For classification, in real-world applications, data/dataset preparation ensures decision correctness; however methods are lacking for correct dataset preparation
  - **Optimum performance algorithm**: The availability of algorithms provides freedom during selection, but increases chances of picking inappropriate and sub-optimal algorithm, specially in multi-criteria situation

Goals

- Accurate classification model creation based on correct dataset and accurately selected appropriate optimum performance algorithm

Objectives

- Correct dataset preparation and classification/recommendation model creation in real-world applications
- Appropriate optimum performance classification algorithm selection based on multi-criteria analysis

Challenges

How to prepare real-world applica. data
How to prepare real-world cases
How to accurately design dataset

- RST Model
- Hybrid-CBR Model

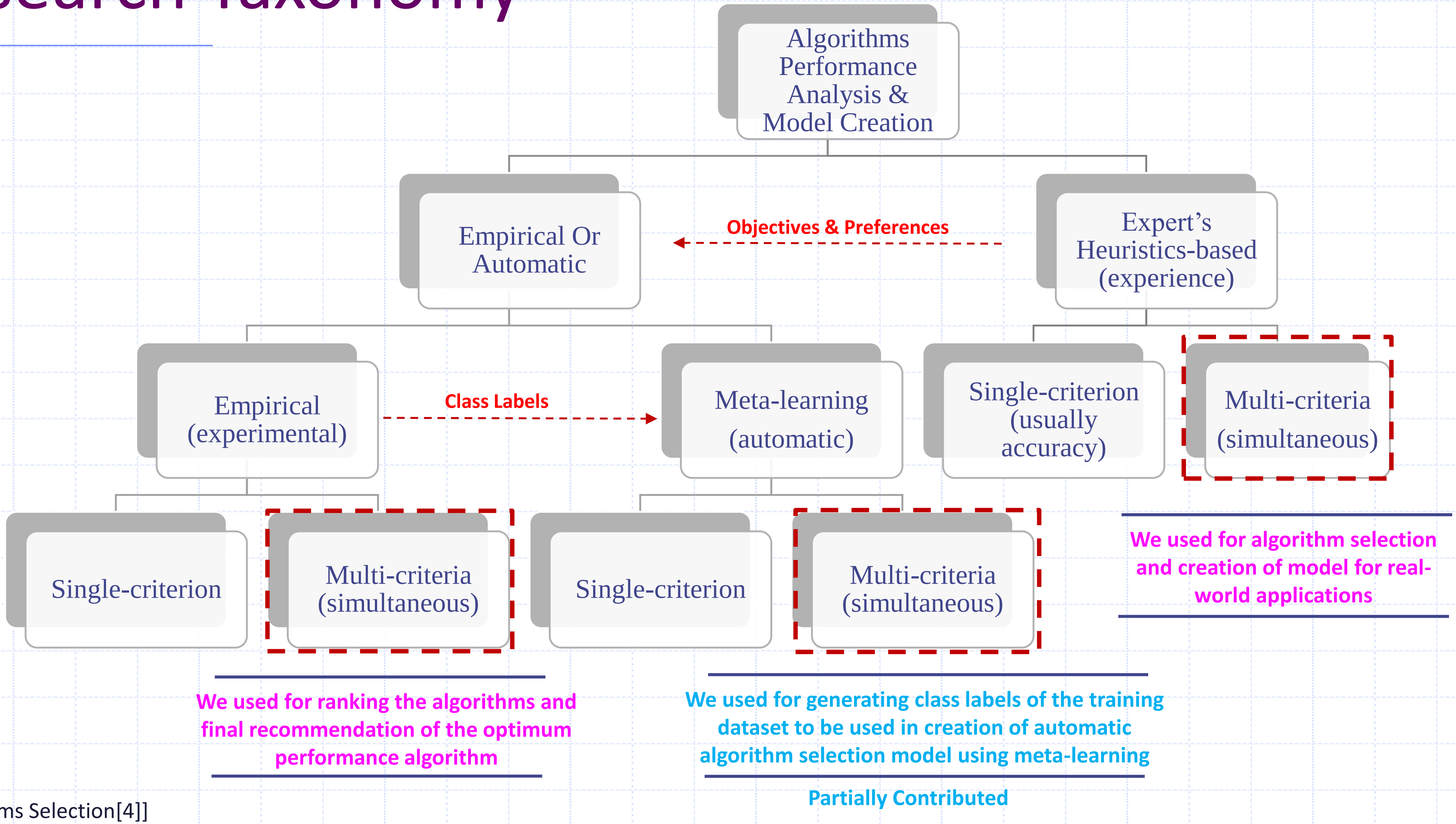
\+

- AMD

How to select suitable performance metrics
How to weight metrics
Satisfy constraints and rank algorithms

경희대학교
KYUNG HEE UNIVERSITY

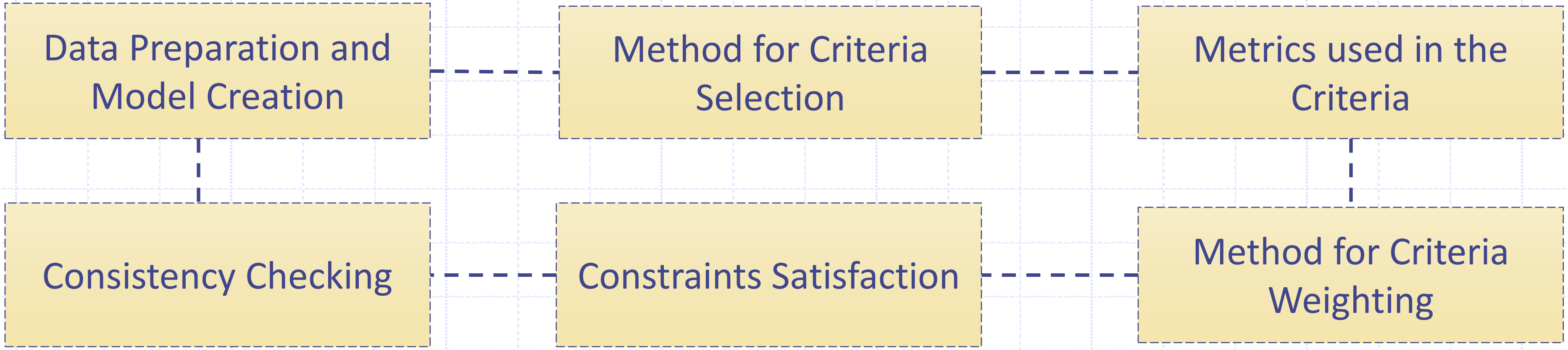
Research Taxonomy



[Taxonomy of Algorithms Selection[4]]

Related Work

- Decision making (classification and recommendation) model creation and method selection techniques [5] [6,7] [1]
  - Focus on heuristic-based, over the educated guess of the expert
  - Focus on empirical analysis, using cross-validation techniques and analysis of results
  - Focus on automatic meta-learning mechanism, using learning meta-characteristics
- A complete framework for model creation and algorithms performance analysis requires [8,9,10]



Related Work Summary

Evaluation criteria	Data Preparation	Evaluation Metric(s)	Standard method criteria selection	Preferences or criteria weighting	Constraints satisfaction	Consistency measure	References
Heuristic-based decision making	No standard	Accuracy	Heuristics	x	x	x	Cho (1990), Aha (1992), Brodley (1993), Brazdil & Henery (1994), Gama & Brazdil (1995), Linder and Studer (1999), Kalousis & Theoharis (1999), Smith et al. (2002),
		Average error, accuracy	Heuristics	Absolut weighting	x	x	Smith, K.A (2001)
		Accuracy , comprehensibility	Heuristics	x	x	x	Gang Luo (2015)
Automatic Empirical using CV	Pre-defined (prepared)	Accuracy and Time, Tree Size	x	x	Partial	x	Lim et al. (2001)
		Accuracy and Time (Train,test)	x	Partial Relative weighting	x	x	Brazdil et al. (2003)
		Sens, Prec, F-score, AUC	x	x	Partial	x	C Romero (2013)
Automatic Meta-learning	Pre-defined (prepared)	Avg. Train Time, Accuracy, Memory Usage	x	Partial relative weighting	Partial	x	Khanmohammadi S (2014)
		RMSE, PMCC	x	Absolut weighting	Partial	x	M Reif (2014)
		Avg Accuracy, Tim Complexity (Train, Testing)	x	Absolute weighting	Partial	x	Shawkat Ali (2005)
		Accuracy, Training & Testing Time	x	Partial Relative weighting	Partial	x	Zhang X et. al (2012)
		T-test and Freidman test-Holm	x	Absolute weighting	Partial	x	Wang G (2014)
Proposed	Both	F-score, Training and Testing Time, Consistency	Experts' Consensus-based Grouped DM	Group decision making for relative consistent weighting	Full (Implicit & explicit)	Consistency (Avg. Stdev.)	Ali R (2015[11], 2016a[12], 2016b[24])

Limitations of Existing Work

- Lack of accurate models and proper dataset/cases preparation
- Lack of standard method for suitable criteria selection
- Use of absolute criteria weighting
- Lack of support for implicit and explicit constrains on criteria
- Lack of appropriate consistency measure in the evaluation process

Proposed Methodologies

Limitations

- Lack of accurate classification and recommendations model for Real-world Applications
- Lack of proper dataset/cases preparation

- Lack of standard method for selecting criteria and use of absolute weights
- Limited support for constraints and lack of consistency measure for performance analysis

Objectives

Accurate Data preparation and Models Creation for Real-world Applications

Accurately Selecting Optimum Performance Consistent Algorithm for Classification Problem

Solutions

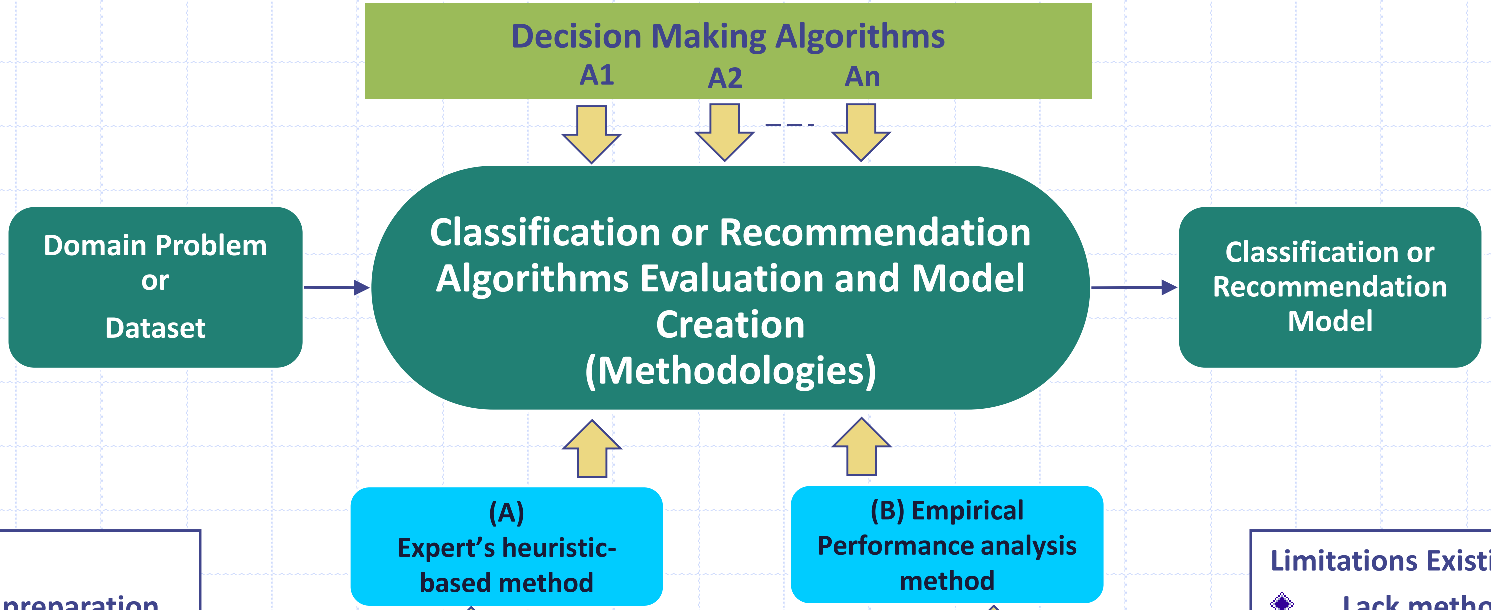
Heuristic-based Model Creation (A)

- A-1 **H2RM**: Rough set classification model for semantics preserved accuracy
- A-2 **Hybrid-CBR:** CBR model for accurate precise recommendations

Empirical Performance Evaluation of Algorithms (B)

- **AMD:** Accurate multi-criteria decision making with suitable criteria selection, relative weighting constarinats satisfication for accuratle and consistent classification algorithm.

Proposed Methodologies – A conceptual model



Limitations Existing Studies
- Data/dataset /case base preparation
- Accuracy is not preserved
- Low accuracy for multi-class problems

Proposed H2RM [11] & hybrid-CBR[13]
- A-1: Semantics preserved accuracy
- A-2: Highly accurate and precise decision

Algorithm A-1
- H2RM: Rough set classification model

Algorithm A-2
- Hybrid-CBR recommendation model

Algorithm B
- AMD: Accurate multi-criteria decision making for algorithm selection

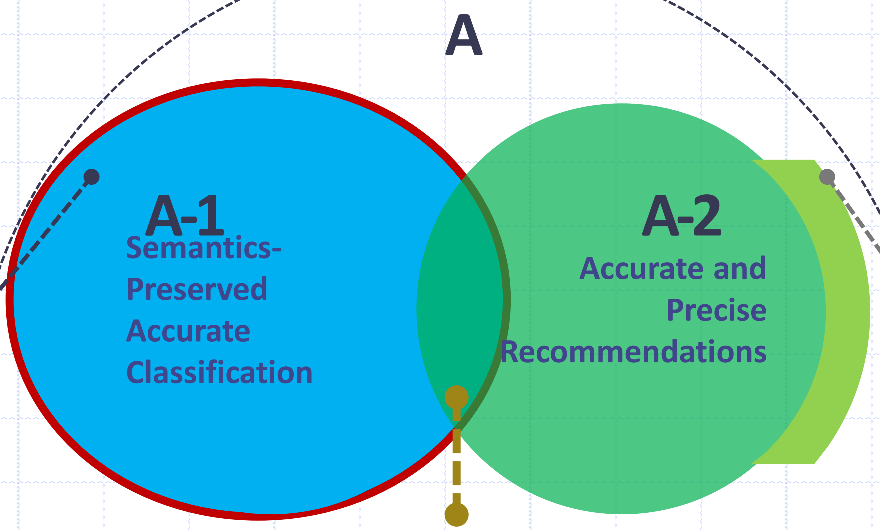
Limitations Existing Studies
- Lack method for criteria selection
- Absolute criteria weighting
- Lacks constraints and consistency

Proposed AMD (B[24])
- Standard method for criteria selection
- Relative consistent weights estimation
- Constraints satisfaction & consistency measure used

[11] **Ali R**, et al.,. H2RM: A hybrid rough set reasoning model for prediction and management of diabetes mellitus. Sensors. 2015 Jul 3;15(7):15921-51.
[13] **Ali R,** et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.
[24] **Ali R,** et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Heuristics-based models for real-world applications (Solution A1)

Semantics-Preserved Accurate Rough Set Classification Model for Real-world Application



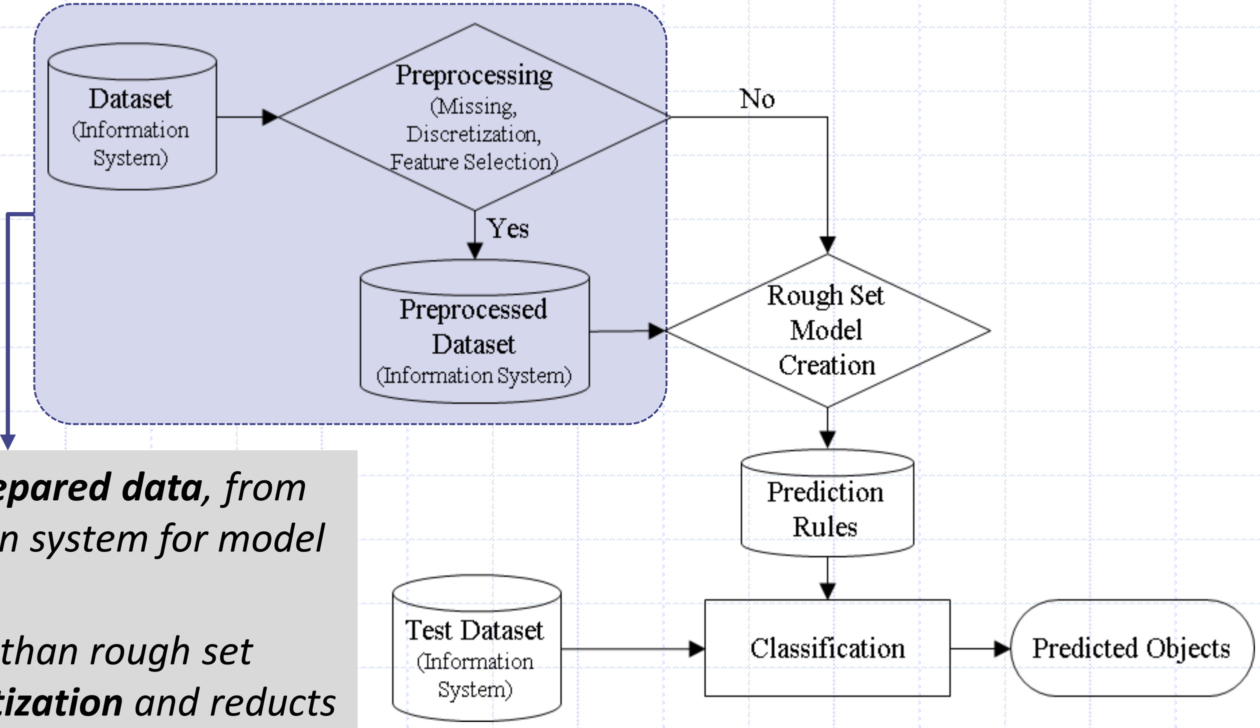
Hybrid Case-based Reasoning Model for Accurate Real-world Application Services

- Correct data and dataset preparation
- Correct case and case-base design and preparation

Data Preparation	Model Evaluation & Selection
Domain specific guidelines-enabled rule-based method for data preparation	Models/algorithms are selected based on the experts' heuristic knowledge

Underlying Technologies : Rough Sets Theory (Solution A-1)

Rough set **classification process**, based on rough set theory (RST), which uses a formalism for **representing** and analyzing **data** in a specific structured format called **information system**



Classical RS Classification Model

__Rough set classification__ uses the concepts of __lower and upper approximations__ to roughly estimate the classes that __cannot be distinguished__ based on the available attributes set

Limits

- ***Consumes, prepared data**, from the information system for model creation.*
- *Nothing more than rough set **default discretization** and reducts and core generation*

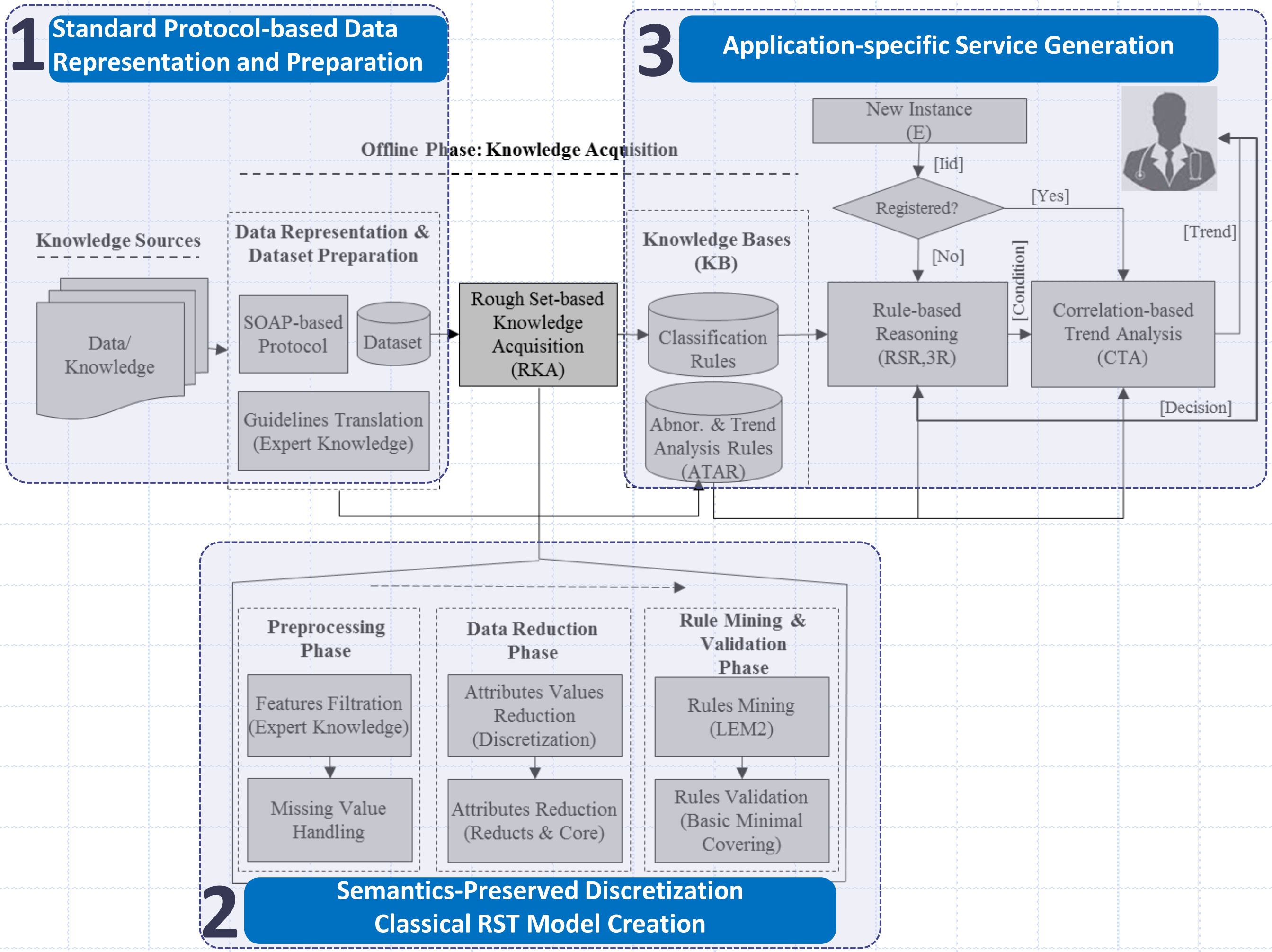
Why RST? - Expert's Heuristics Criteria

- *Structural relationships in **imprecise & noisy data***
- *Better approximations of **vague boundaries** data*
- ***No extra parameters** setting*
- ***Interpretable model***

Bonikowski, Z.; Bryniarski, E.; Wybraniec-Skardowska, U., Extensions and intentions in the rough set theory. Information sciences 1998, 107, (1), 149-167.

경희대학교 KYUNG HEE UNIVERSITY

Accurate Rough Sets Classification Model (Solution A-1)



Algorithm. Rough Set Classification and Reasoning

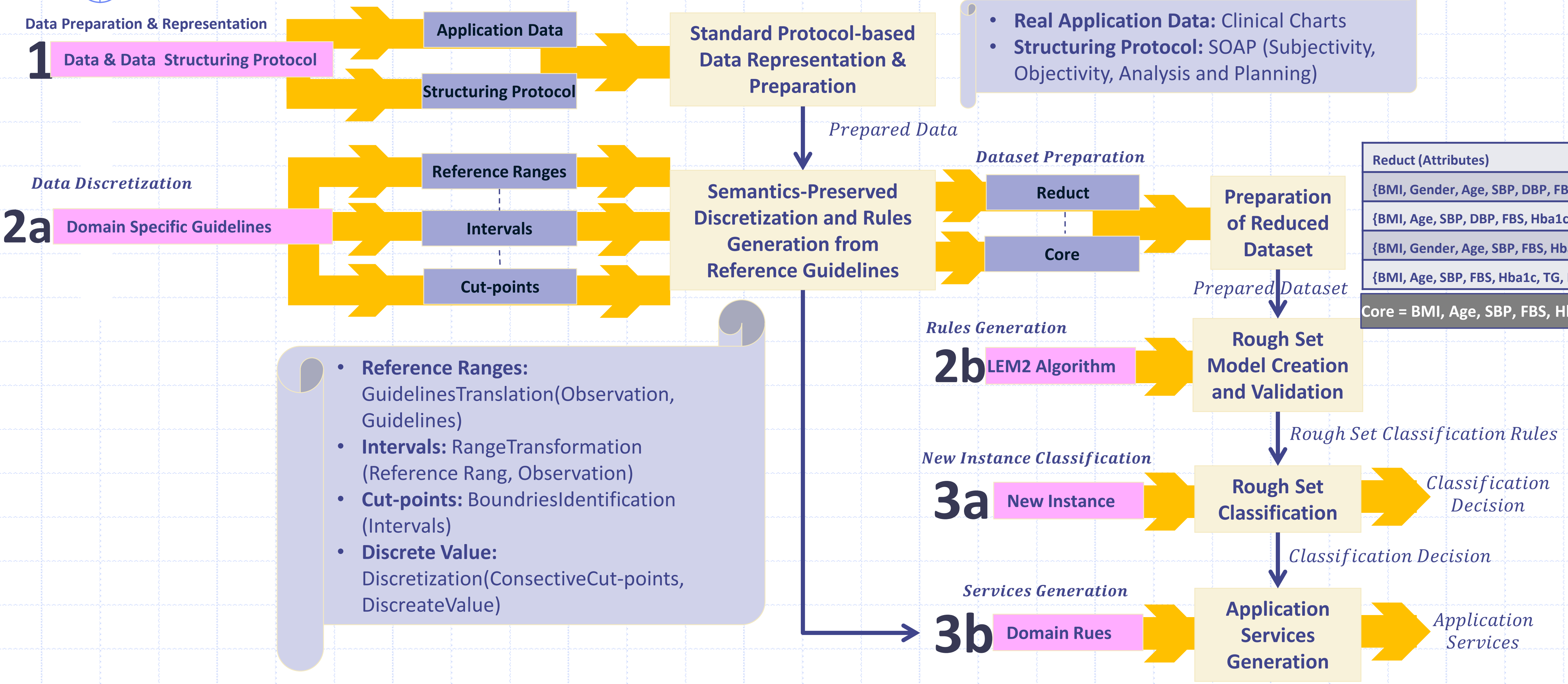
Input: Data, E: New Instance
Output: classType, INTERPRETATION

Begin
CreateRSModel (Data)
[Prepare Dataset]
1. IS: = prepareInfoSystem (Guidines, Data);
[Rough Set Classification Model Creation]
2. IS′ = preprocessIS(IS);
3. DIS = semanticsEnabledDiscretize(Guidlines, IS′)
4. RDS = reductAndCoreGeneration(DIS, Core(Reduct))
5. RSRules= createModel(RDS, LEM2);
[Rough Set Classification using Rough Set Reasoning]
ApplyRBR (E), where {E|E is New Instance, E: = {Iid, Cond}}
A. PerformRSR(E) // Rough Set Reasoning
[Load Classification Rules From Knowledge Base]
1. DMPR: =
LoadRulesFromKB(RSRules that contain classType as CONC); where CONC: = { classType 1, . . , classType n [Execute Rules For Classification of CONC]
2. Foreach RULE in DMPR
a. Foreach CA in RULE //CA: = {Cond} are conditions of the rule
b. If CA. values ≠ E. Cond. value
THEN Try next RULE
EndIf
c. classType ≔ CONC of the RULE;
d. Goto Step B
e. EndFor
EndFor
3. classType = Message(UNDEFINED);
[Reference Range-based Reasoning for Risk Prediction]
B. Perform3R (I) // Reference Range-based Reasoning
[Load Reference Range Rules From Knowledge Base]
4. ATAR: =
LoadRulesFromKB(RULES that contain INTERPRETATION as CONC); where CONC: = { INTERPRETATION. Rules. Value}
[Execute Rules For Finding Current Status of Each Observations]
5. Foreach RULE in ATAR
a. Foreach CA in RULE
b. If CA. values ≠ E. OBS. value
THEN Try next RULE
EndIf
c. INTERPRETATION [] ≔ CONC of the RULE;
EndFor
EndFor

[Classification Results Generation]
C. RSResults ≔ ProvideResults (Iid, classType, INTERPRETATION)
End

[11] **Ali R**, Hussain J, Siddiqi MH, Hussain M, Lee S. H2RM: A hybrid rough set reasoning model for prediction and management of diabetes mellitus. Sensors. 2015 Jul 3;15(7):15921-51.

Accurate Rough Sets Classification Model (Solution A-1)

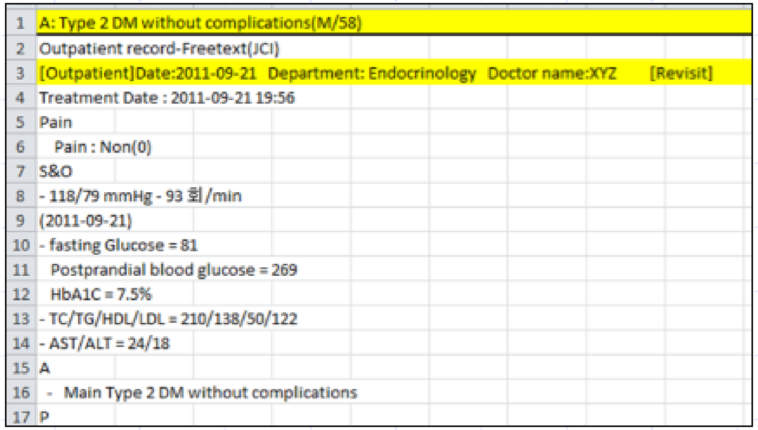


Reduct (Attributes)
{BMI, Gender, Age, SBP, DBP, FBS, Hba1c, HDL, LDL, PT}
{BMI, Age, SBP, DBP, FBS, Hba1c, TG, HDL, LDL, PT}
{BMI, Gender, Age, SBP, FBS, Hba1c, HDL, LDL, OT, PT}
{BMI, Age, SBP, FBS, Hba1c, TG, HDL, LDL, OT, PT}
Core = BMI, Age, SBP, FBS, Hba1c, HDL, LDL, PT}

[11] **Ali R**, Hussain J, Siddiqi MH, Hussain M, Lee S. H2RM: A hybrid rough set reasoning model for prediction and management of diabetes mellitus. Sensors. 2015 Jul 3;15(7):15921-51.

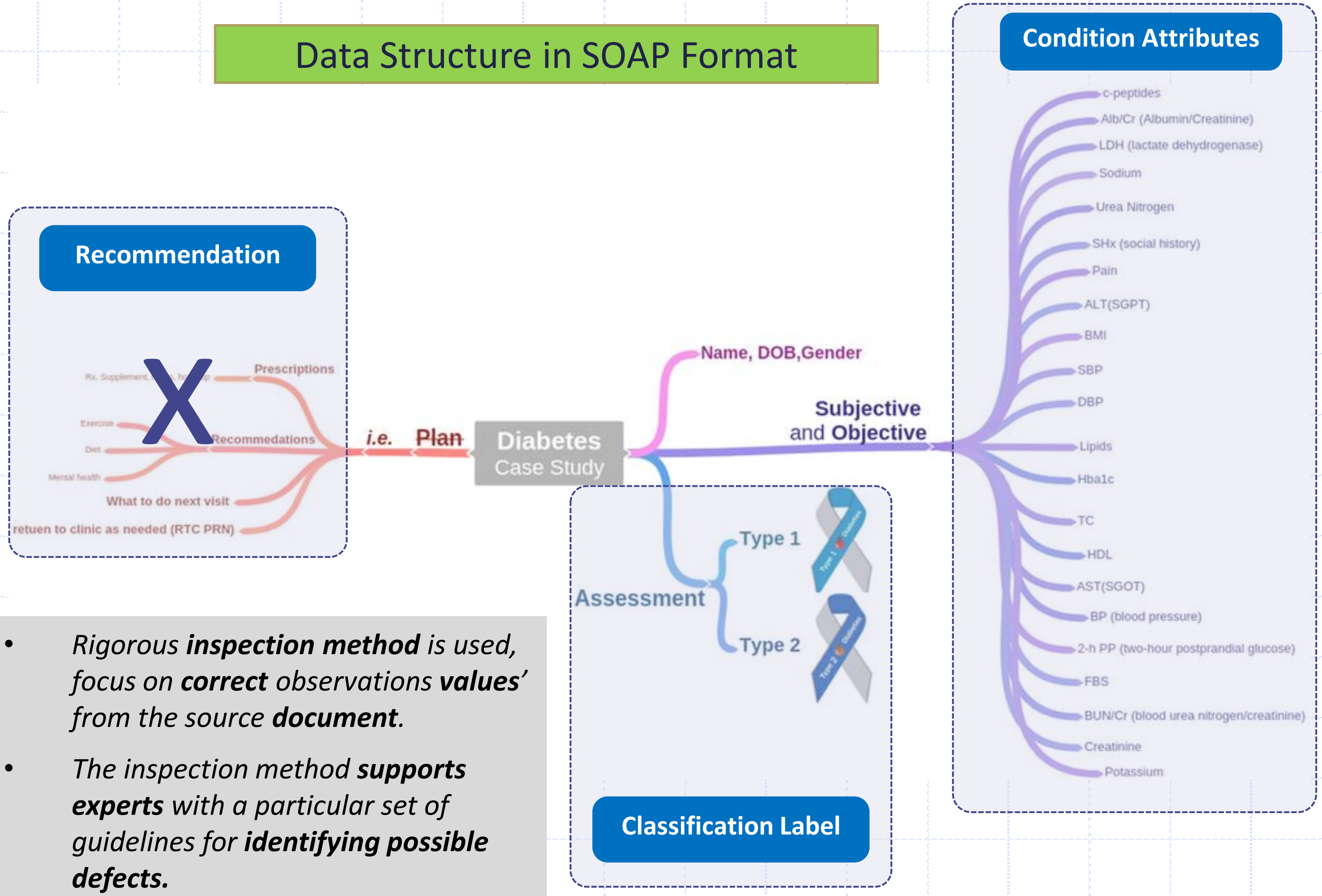
Accurate Rough Sets Classification Model (Solution A-1)

1 Application Data Structure in Clinical Notes



S.No	Predictor	Guidelines	References
1	BMI	WHO: BMI classification	WHO [36]
2	BP: SBP, DBP	JNC 7 report, AHA	JNC [37–39]
3	FBS	American Diabetes Association. Diabetes Care	ADA [40,41]
4	HBA1c	American Diabetes Association, NICE	ADA [40], NICE [42,43]
5	Lipids: TC, TG, HDL, LDL	NCEP, ADA	NCEP [44], ADA [45]
6	LFT: ALT, AST	Liver disease (LD), Mayo Clinic	LD [46], Mayo Clinic [47]

Data Structure in SOAP Format



- *Rigorous **inspection method** is used, focus on **correct** observations **values**' from the source **document**.*
- *The inspection method **supports experts** with a particular set of guidelines for **identifying possible defects.***

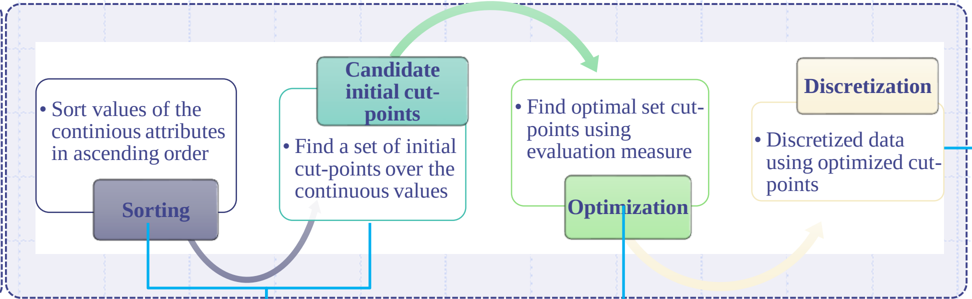
http://www.bcs.org/upload/pdf/ewic_hc07_sppaper25.pdf

[11] **Ali R**, Hussain J, Siddiqi MH, Hussain M, Lee S. H2RM: A hybrid rough set reasoning model for prediction and management of diabetes mellitus. Sensors. 2015 Jul 3;15(7):15921-51.

Accurate Rough Sets Classification Model (Solution A-1)

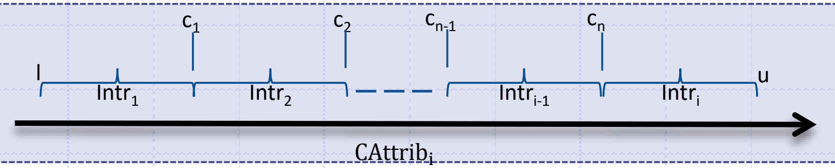
2a Rough Set-based Discretization [12]

Discretization Process



$$C = \left\{\left(\frac{v_1+v_2}{2}\right), \left(\frac{v_2+v_3}{2}\right), \dots, \left(\frac{v_{n-2}+v_{n-1}}{2}\right), \left(\frac{v_{n-1}+v_n}{2}\right)\right\} = \{c_1, c_2, \dots, c_n\}$$

Statistical, entropy, genetic, fuzzy theory and Boolean Reasoning



Example: SBP Attribute

Existing (Boolean Method) -- >

- (SBP < 110), (SBP 110-116), (SBP ≥117)

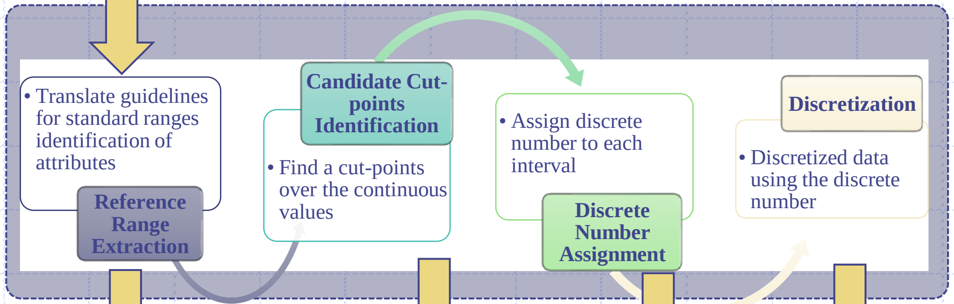
Required Discretized Value -- >

- (-α, 120),[120, 139],[140, 159],[160, 180],[181, α)

- **Semantics Distorted**
- **Model semantically incorrect**

Semantics-Enabled Discretization

S.No	Predictor	Guidelines	References
1	BMI	WHO: BMI classification	WHO [36]
2	BP: SBP, DBP	JNC 7 report, AHA	JNC [37–39]
3	FBS	American Diabetes Association. Diabetes Care	ADA [40,41]



(c) SBP		Cut-Points	Discreet Number
Interval	**Semantics**		
(−∞, 120)	normal	120	0
[120, 139]	prehypertension	140	1
[140, 159]	hypertension stage 1	160	2
[160, 180]	hypertension stage 2	181	3
[181, ∞)	hypertensive crisis		

R #	Class	Prediction Rule
1	(T1DM)	(BMI = [18.5, 24.9]) and (Age = (50, ∞)) and (SBP = [120, 139]) and (Hba1c = (7.4, ∞)) and (TC = (−∞, 200)) and (SGPT = [7, 56])
2	(T2DM)	(Gender = M) and (SBP = (−∞, 120)) and (Hba1c = (6.4, 7.4]) and (LDL = [100, 129])

[12] Ali R, Siddiqi MH, Lee S. Rough set-based approaches for discretization: A compact review. Artificial Intelligence Review. 2015 Aug 1;44(2):235-63.

H2RM

Semantics-preserved accurate rough set classification model

경희대학교 KYUNG HEE UNIVERSITY

Accurate Rough Sets Classification Model (Solution A-1)

- **Evaluation Criteria**
  - **Average** and balanced **accuracy**
- **Experimental Setup**
  - Windows, PC, RAM 4GB.
  - ROSE 2 [13],default parameters,10-fold CV.
- **Dataset**
  - Real Dataset St. Mary's 50 patient (20 Type-1 & 30 Type-2).
  - **391 records,** 278 encounter Type-2 & 113 for type-1.
  - **Attributes: 8** {BMI, Age, SBP, FBS, Hba1c, HDL, LDL, PT}
- **Results: Semantic Preserved Classification Accuracy**

Average Accuracy

Type of DM	Correct	Incorrect	None
T1DM	94.59 ± 6.16	5.41 ± 6.16	0.00 ± 0.00
T2DM	96.85 ± 4.11	3.15 ± 4.11	0.00 ± 0.00
Total	**95.91 ± 2.61**	**4.09 ± 2.61**	**0.00 ± 0.00**

Balanced Accuracy

Type of DM	T1DM	T2DM	None
T1DM	106 (TP)	7 (FN)	0
T2DM	9 (FP)	269 (TN)	0

$$\text{Balanced accuracy} = \frac{0.5 * \text{TP}}{\text{TP} + \text{FN}} + \frac{0.5 * \text{TN}}{\text{TN} + \text{FP}} = 0.9522$$

Average Error

Fold	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6	Pass 7	Pass 8	Pass 9	Pass 10	Average Error
Percent Error	2.50	7.69	5.13	7.69	5.13	0.00	2.56	5.13	2.56	2.56	**4.10**

- **Comparison**
  - Comparison with Seven Rule-based Algorithm from Weka environment, default setting
  - Comparison on the basis of **Average accuracy**

Classifier	rules.DTNB	rules.JRip	rules.NNge	rules.PART	rules.Ridor	rules.DecisionTable	Rough.Set.LEM2
Average Accuracy	91.31(2.74)	95.13(2.73)	94.16(3.72)	**96.16(2.18)**	94.88(2.79)	89.52(3.69)	**95.9(2.6)**

Less than

Greater than, but not significant. Both proposed and PART has same significant

[11] **Ali R**, Hussain J, Siddiqi MH, Hussain M, Lee S. H2RM: A hybrid rough set reasoning model for prediction and management of diabetes mellitus. Sensors. 2015 Jul 3;15(7):15921-51.

Accurate Rough Sets Classification Model (Solution A-1)

Contributions

1 Semantics-preserved discretization

- Guidelines-enabled discretization scheme for retaining or preserving semantics while data is transformed from continuous values to discreat in the rules.

2 Guidelines enabled data and dataset preparation

- Rigorous inspection-based method for real world dataset preparation using standard domain knowledge in the form of guidelines

3 Classification dataset for research community

- An accurate dataset is prepared for research purpose and made available to the community in anonymized form

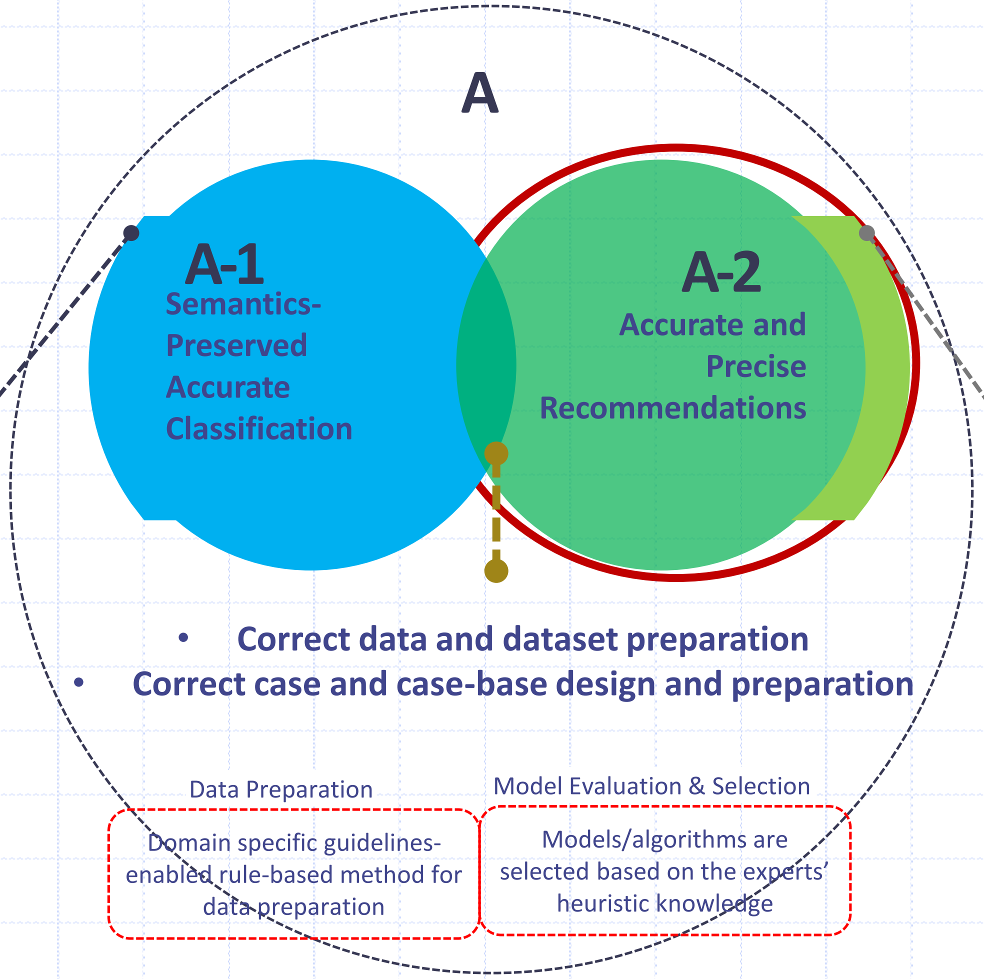
Discussion on RS-based Classification Model

- Needs **labeled** and properly structured dataset
- Needs **large datasets** for better performance
- Datasets with **large number of classes**, lower accuracy
- **Lack** of **incremental learning**
- Supports **generalization** rather than specialization

[11] **Ali R**, Hussain J, Siddiqi MH, Hussain M, Lee S. H2RM: A hybrid rough set reasoning model for prediction and management of diabetes mellitus. Sensors. 2015 Jul 3;15(7):15921-51.

Heuristics-based models for real-world applications (Solution A2)

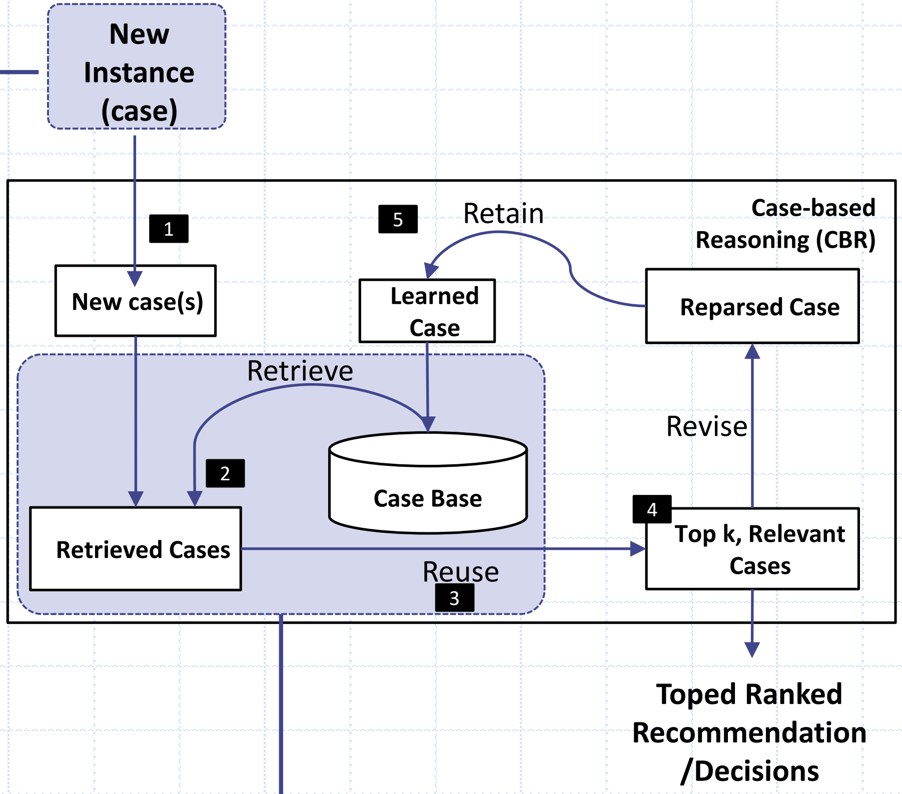
Semantics-Preserved Accurate Rough Set Classification Model for Real-world Application



Hybrid Case-based Reasoning Model for Accurate Real-world Application Services

Underlying Technologies : Case-based Reasoning (Solution A-2)

Case-Based Reasoning (**CBR**) is a **classification method** that uses **past experiences rather** than general **knowledge** and a problem P is represented as a collection of examples or cases, i.e., $P = \{c_1, c_2, \dots, c_n\}$, where each $C_i = \{f_1, f_1, \dots, f_n\}$



Limitations of Existing

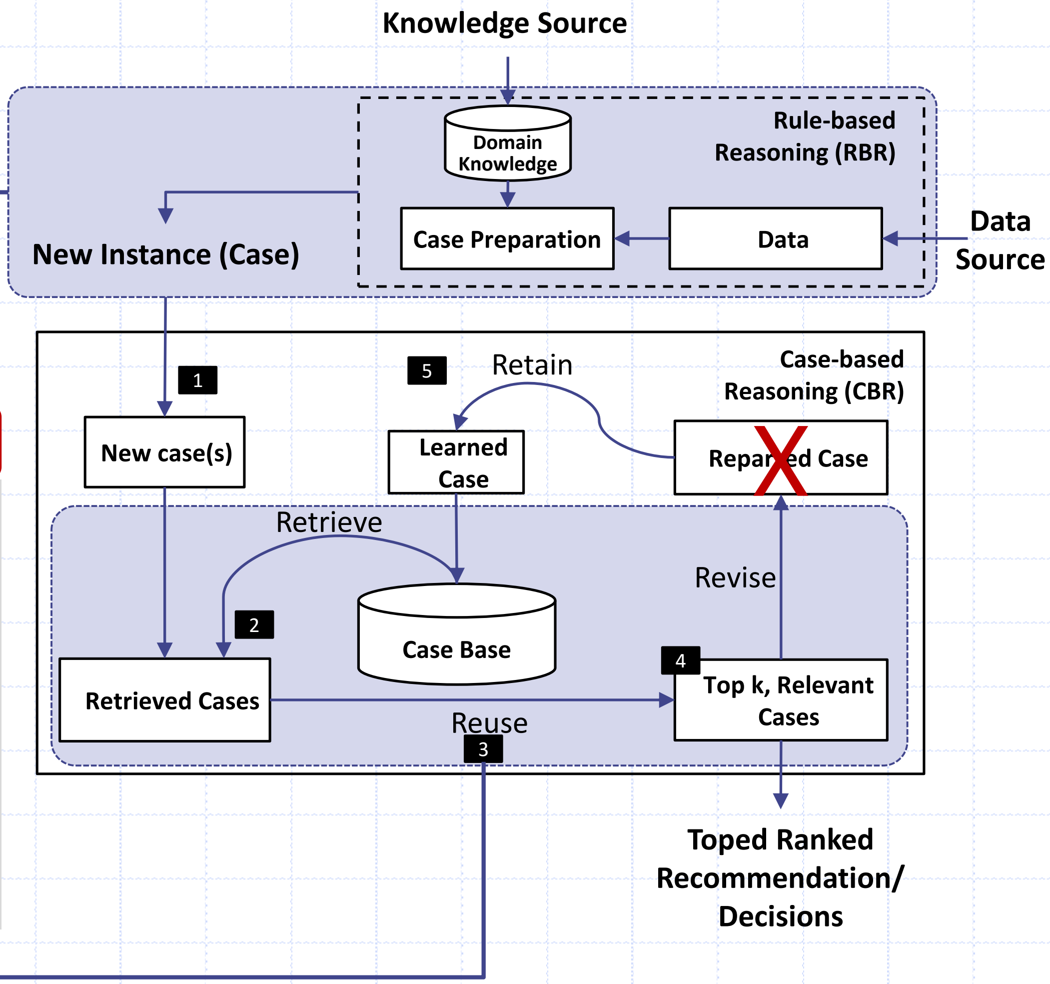
- ***Consumes, prepared cases,*** *from the applications*
- *No proper mechanism of* ***new case creation at run-time*** *for real applications*
- *Uses* ***default similarity functions***

Why CBR? - Expert's Heuristics

- ***Accurate results*** *in case of* ***small dataset*** *and* ***large number*** *of* ***classes***
- *Recommends* ***similar solution even if no exact matched found***
- ***Incremental learning***
- ***Supports specialization***
- ***Ranking recommendation***

Leake DB. Case-Based Reasoning: Experiences, lessons and future directions. MIT press; 1996 Sep 1.

Hybrid CBR Recommendation Model (Solution A-2)



Key contributions

- *Guidelines-enabled **rule-based method** for case **preparation***
- ***Accurate case retrieval** using local & global **similarity functions***

Algorithm. CBR methodology for accurate recommendations

Input: nC:= new Case
Output: List R <Recommendations>

Begin

[Create Successful Case for Case Base]

$\text{METCB} = \text{createCasesUsingRBR}(\text{Data}, \text{Domain Knowledge})$

Let R:= A set of top-k relevant recommendations

Sim_g[]:= Array of global similarities of existing cases

[Loading Cases from Case Bbase]

$\text{METCB}_r := \text{ReteriveCaseBaseFromKB}(\text{METCBurl})$, Where METCB_r is the matrix $eC_m x A_n$, eC_m is the set of existing cases, i.e., $eC = eC_1, eC_2, eC_3, \dots, eC_m$. Similarly, A_n is the set of attributes, i.e., $A_n = A_1, A_2, A_3, \dots, A_n$

[Similarity Check of the Case base for the New Case]

1. **For** $i = 1$ to $\text{SizeOfCases}(\text{METCB}_r)$
   Let Sim_l[]:=Array of local similarities of attributes of individual cases
   a. **For** $j = 1$ to $\text{SizeOfAttributes}(\text{METCB}_r)$
   b. $\text{Sim}_l[A_j] := \text{ComputeLocSim}(nC.A_j, \text{METCB}_r[i,j])$;//use eq.11 & eq. 12
   c. **End for**
   d. $\mathbf{Sim_g}[eC_i] := \text{ComputeGlobSim}\ (\mathbf{Sim_l})$; // weightedsum method (eq.13)
2 **End for**

[Selecting Top-k Relevant Cases]

3. $R := \text{ApplyKNN}(\mathbf{Sim_g})$; //where $k = 3$

[Providing Recommendations of the Top-k Relevant Cases to the User]

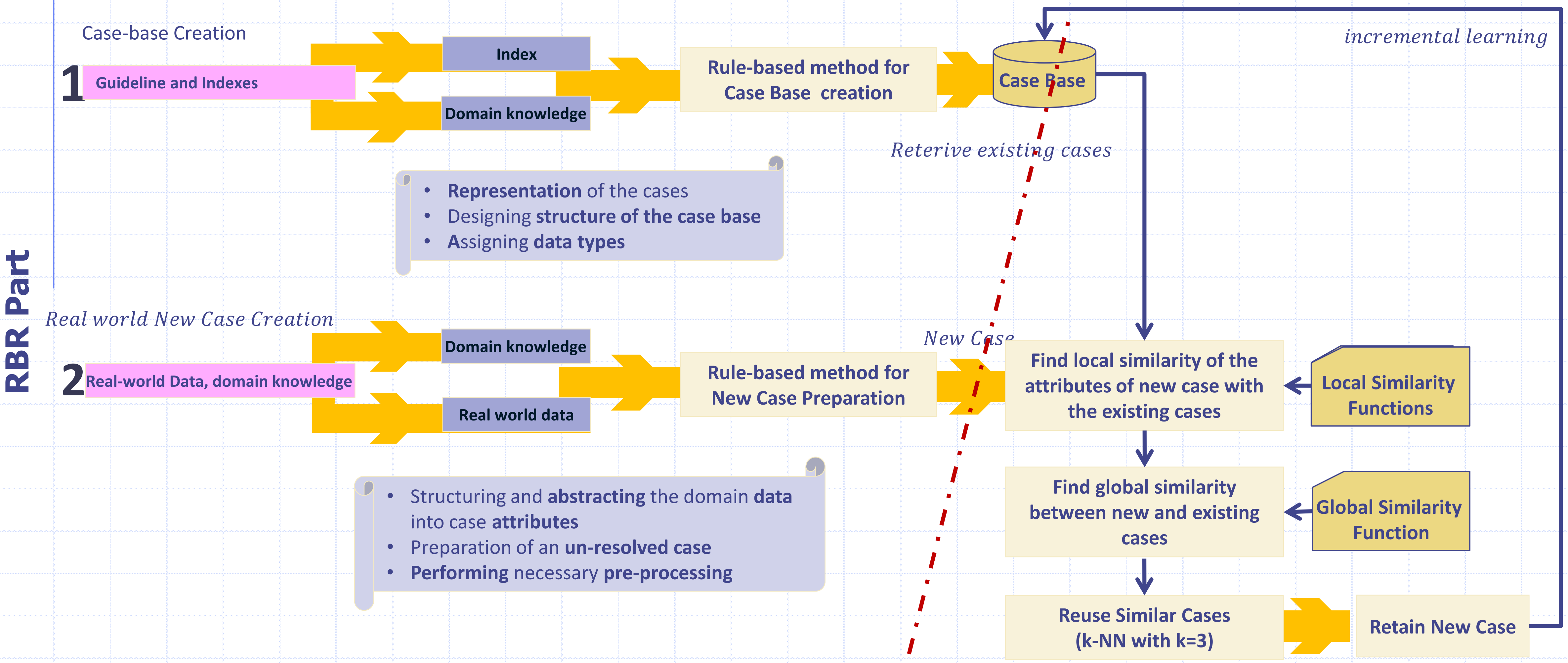
4. PropgateCBRResults (uid, R);

[Retaining the Resolved Case in Case Base]

5. $\text{FCB} \coloneqq \text{RetainCBRPAR}(\text{uid}, \text{R})$;
6. **Exit; End**

[13] Ali R, et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

Hybrid CBR Recommendation Model (Solution A-2)



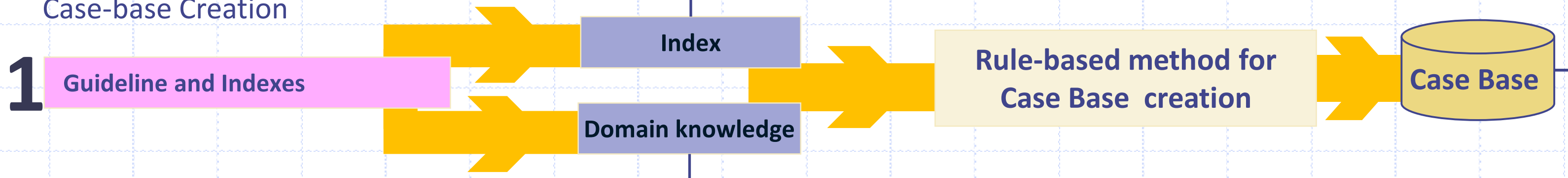
[13] Ali R, et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

경희대학교 KYUNG HEE UNIVERSITY

Hybrid CBR Recommendation Model (Solution A-2)

RBR Part

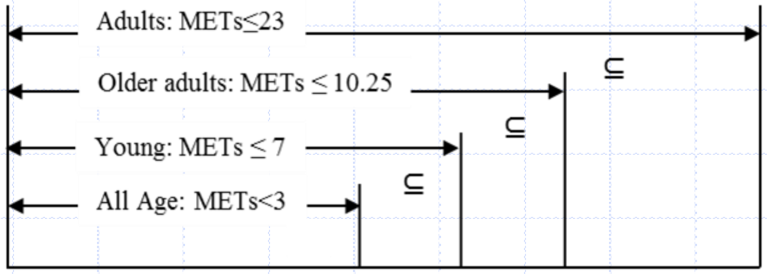
Case-base Creation



Index: METs Index [Compendium of Physical Activity]

Exercise	MET	Recommendation
Cycling	14.0	bicycling, mountain, uphill, vigorous
Cycling	16.0	bicycling, mountain, competitive, racing
Walking	8.5	bicycling, BMX
Jogging	8.5	bicycling, mountain, general

Guidelines: MET Vs Age-group Relationship [WHO [133] and UK [134], ACSM, UK, US]



Rules for case creation

Case Base Structure

Attribute	Data type	Possible value	Description
Age Group	Symbol	{All Age, Young, Adults, Older Adults}	Age of the subject
METs	Float	Min=1.3, Max=23.0	Metabolic Equivalents of Tasks one hour
Recommendations	String	Physical activities {running, walking, cycling, traveling-bus and subways, standing, sitting}	Physical activities

MET Case Base

Case #	Age	MET	Recommendation/Classification
1	Adults	14.0	bicycling, mountain, uphill, vigorous
2	Adults	16.0	bicycling, mountain, competitive, racing
3	Older Adults	8.5	bicycling, BMX
4	Older Adults	8.5	bicycling, mountain, general
5	All Age, Young, Older Adults, Adults	3.5	bicycling, leisure, 5.5 mph
119		…	…

- ***Inspection method** used, focus on **correct** case preparation from guideline and domain knowledge.*
- *Inspection method **supports experts** with a set of guidelines for **identifying possible defects.***

[13] Ali R, et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

Hybrid CBR Recommendation Model (Solution A-2)

BMI Rules

Gender	Age	*BMI* value	Weight status
M or F	>20	<18.5 kg/m^2	Underweight
M or F	>20	>18.5 and <25 kg/m^2	Normal
M or F	>20	>25 and <30 kg/m^2	Overweight

Goal/Plan Rules

Gender	Global Goal (gloGoal) - Kg	Weight Status (WS)	Plan Prescription (PP)
M or F	> 0 (+ive)	Normal or Overweight	Weight Loss Plan (WLP): lose gloGoal(Kg)
M or F	= 0 (neutral)	Normal	Weight Maintenance Plan (WMP): motivational statements
M or F	< 0 (-ive)	Underweight	Weight Gain Plan (WGP): gain gloGoal(Kg)

RBR Part

$$\text{idlWgt} = 51.65\text{ kg} + 1.85\text{ kg/inch over 5 feet} \quad (\text{man})$$

$$\text{idlWgt} = 48.67\text{ kg} + 1.65\text{ kg/inch over 5 feet} \quad (\text{woman})$$

$$\text{wghRedPlan (days)} = \text{roundup}\left(\frac{7(\text{days}) * \text{gloGoal(Kg)}}{0.5\text{ (Kg)}}\right)$$

$$\text{calToBurDay} = \frac{\text{gloGoal(kg)} * \text{Cal (1kg fat)}}{\text{wghRedPlan (days)}}$$

$$\text{METs} = \frac{\text{remCalToBurn}}{(\text{amtAct} = 1\text{h}) * \text{weight (kg)}}$$

CBR Part

[continuous value] Closest match similarity function

$$\mathbf{METSim_l(nC, eC)} = \frac{\mathbf{d_g(Max_{MET}, Min_{MET}) - d_l(nC_{MET}, eC_{MET}) - 1}}{\mathbf{d_g(Max_{MET}, Min_{MET})}}$$

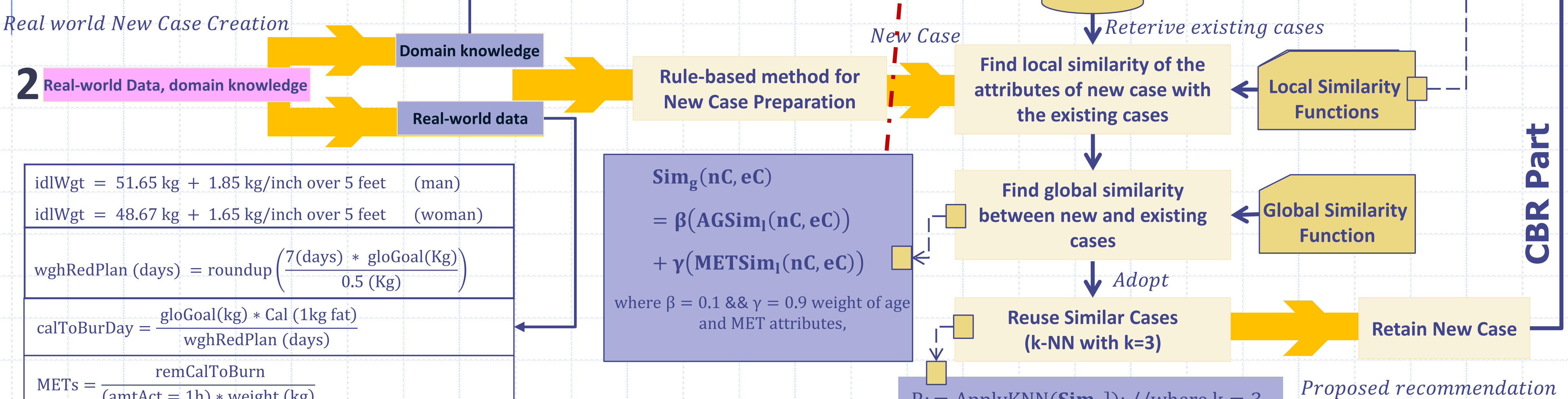
[nominal value] Exact match similarity function

$$\mathbf{AGSim_l(nC, eC)} = \begin{cases} \mathbf{AG_{ij} = 1} & \mathbf{for\ \forall\ (i \geq j) OR (i = 0\ OR\ j = 1)} \\ \mathbf{0} & \mathbf{otherwise} \end{cases}$$

$$\mathbf{Sim_g(nC, eC)} = \boldsymbol{\beta}\mathbf{(AGSim_l(nC, eC))} + \boldsymbol{\gamma}\mathbf{(METSim_l(nC, eC))}$$

where $\beta = 0.1$ && $\gamma = 0.9$ weight of age and MET attributes,

$$\text{R:} = \text{ApplyKNN}(\mathbf{Sim_g}]);\ //\text{where } k = 3$$



[13] Ali R, et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

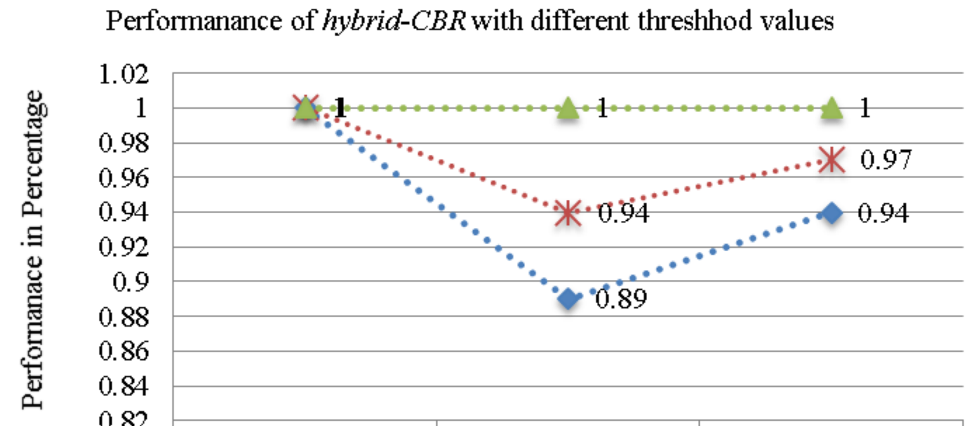
Hybrid CBR Recommendation Model (Solution A-2)

- Experimental setup
  - myCBR, Windows PC, Intel Dual-CoreTM (2.5 GHz), 4 GB RAM.
- Train Dataset (Case Base)
  - 119 METs Cases as knowledge base
- Evaluation criteria
  - Precision, recall, accuracy, and f-score
- Test Dataset (Test Case Base)
  - 64 Test Cases created creted from case base using function $\text{METs. value} = \text{randbetween}(\text{bottom}, \text{top})$

Retrieved Cases and Generated Recommendations

UID	Age-group	METs value)	Retrieved cases (METs value)	Recommendations decision
1	Young	6.5	6.5	climbing hills with 0 to 9 lb load.
			6.5	race walking; rock or mountain climbing
			6.3	climbing hills; no load
2	Adult	7.6	7.3	climbing hills with 10 to 20 lb load
			7.5	bicycling; general
			7.8	backpacking; hiking or organized walking with a daypack
3	Older Adults	7.8	7.8	backpacking; hiking or organized walking with a daypack
			8	running; training; pushing a wheelchair or baby carrier
			8	running; marathon
4	Adults	8.1	8	running; training; pushing a wheelchair or baby carrier
			8	running; marathon

04 Input Test Cases



	Precision	Recall	F-measure
$\mu \geq 95$	1	0.89	0.94
$\mu \geq 90$	1	0.94	0.97
$\mu \geq 85$	1	1	1

[13] Ali R, et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

Hybrid CBR Recommendation Model (Solution A-2)

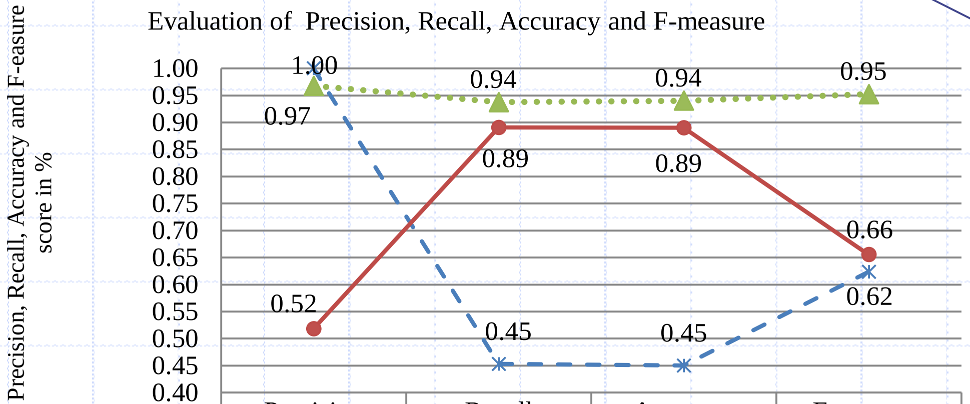
- **Comparison**
  - 64 input cases randomly generated from original cases
- Experiment 1 (Baseline-RBR):
  - RBR with distinct MET rules
- Experiment 2 (Modified-RBR)
  - RBR with ranged-MET rules
- Experiment 3 (Hybrid-CBR)
  - CBR with Test Case Base

MET Distinct Rules

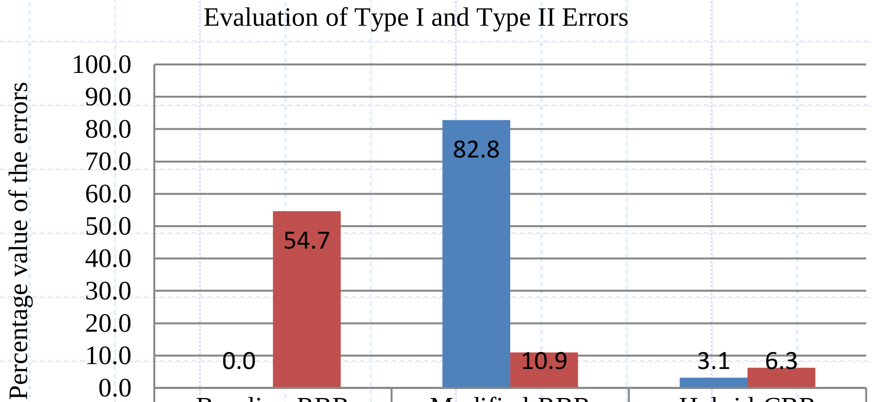
RID	Age	METs	Activity prescription
R#1	Young	2	Walking, household
R#2	Older Adults	6.5	Climbing hills with 0 to 9 lb load; Race walking; rock or mountain climbing
R#3	Young	7.8	Backpacking; hiking or organized walking with a daypack
.	.	.	.
R#122	Adult	15	Running; stairs up

MET Ranged Rules

Rule ID	Age Group	METs value	Activity prescription
R#1	Young, Adults, Older Adults	< 3	Light activity
R#2	Adults	≤ 23	Moderate – vigorous-intensity
R#3	Older Adults	≤ 10.25	Moderate – vigorous (lower intensity level)
R#4	Young	≤ 7	Moderate



	Precision	Recall	Accuracy	F-measure
Baseline-RBR	1.00	0.45	0.45	0.62
Modified-RBR	0.52	0.89	0.89	0.66
Hybrid-CBR	0.97	0.94	0.94	0.95



	Baseline-RBR	Modified-RBR	Hybrid-CBR
Type I error	0.0	82.8	3.1
Type II error	54.7	10.9	6.3

[13] Ali R, et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

Hybrid CBR Recommendation Model (Solution A-2)

Contributions

1. Accurate and precise CBR recommendation model
   - An accurate and precise CBR recommendation model is developed
   - Accurate similarity functions are defined
2. Guidelines enabled case preparation
   - Rigorous inspection method along with the rule-based methodology is used for correct case base and new case creation
3. A useful dataset/case base for research community
   - As an outcome, a correct case-base is released as a useful resources to the research community and people in practicable application fields

Limitations heuristic-based model selection

- Optimality
  - does guarantee the best algorithm will be found (if multiple available)?
- Completeness
  - can heuristic find all suitable algorithms (if available)?
- Accuracy and precision
  - can heuristic provide confidence interval for the claimed algorithm?
- Selection time
  - Is this the best known heuristic for solving this type of problem?

[13] Ali R, et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

Empirical Analysis of Classifiers (Solution B)

B

Empirical Analysis of Classifiers

Multiple Classifiers used for Best Classifier Selection

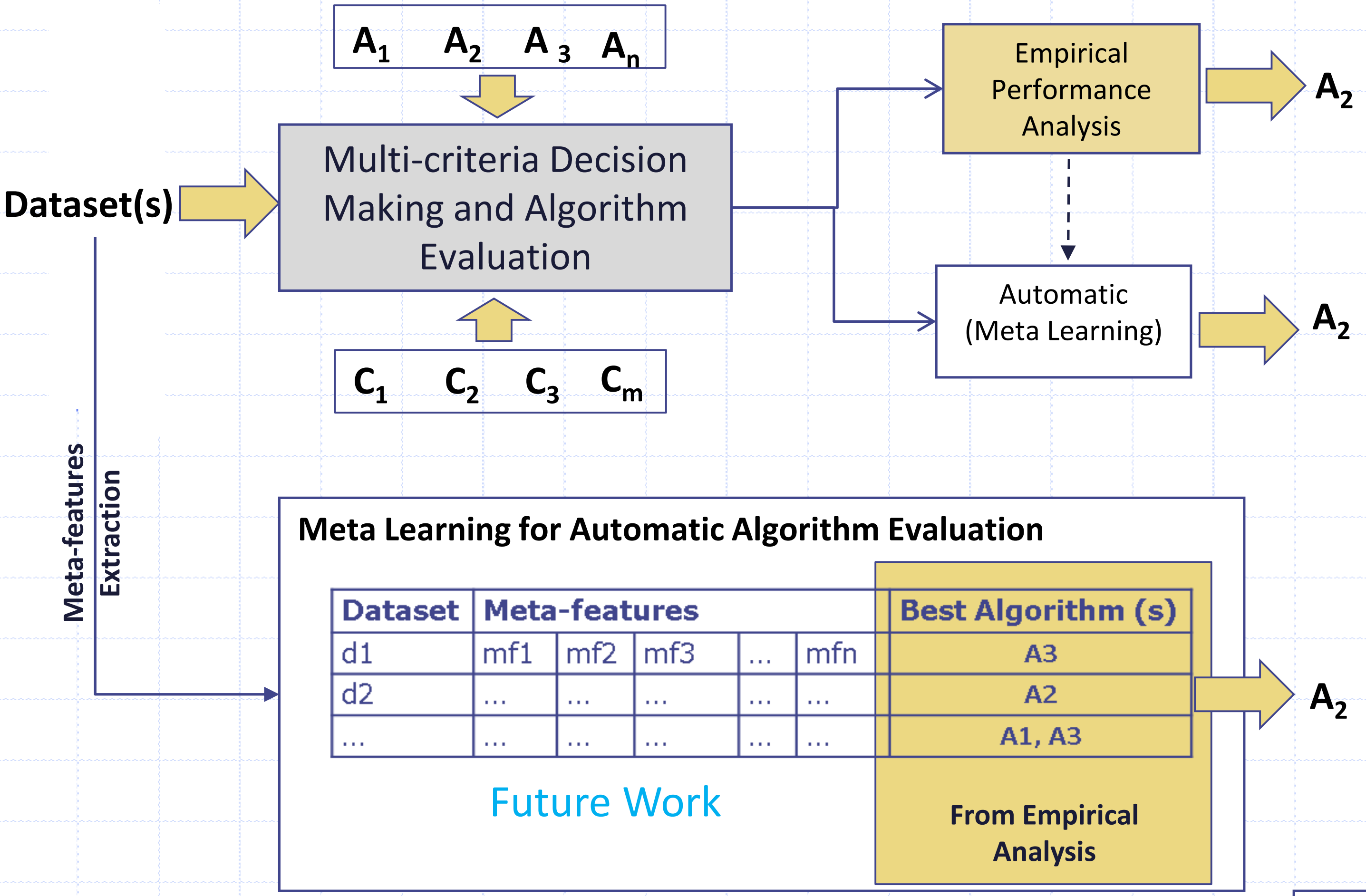
Multiple Criteria are Used to Evaluate and Analyze the Performance

Multi-criteria decision Making

Algorithms are ranked and the top rank is selected for model creation

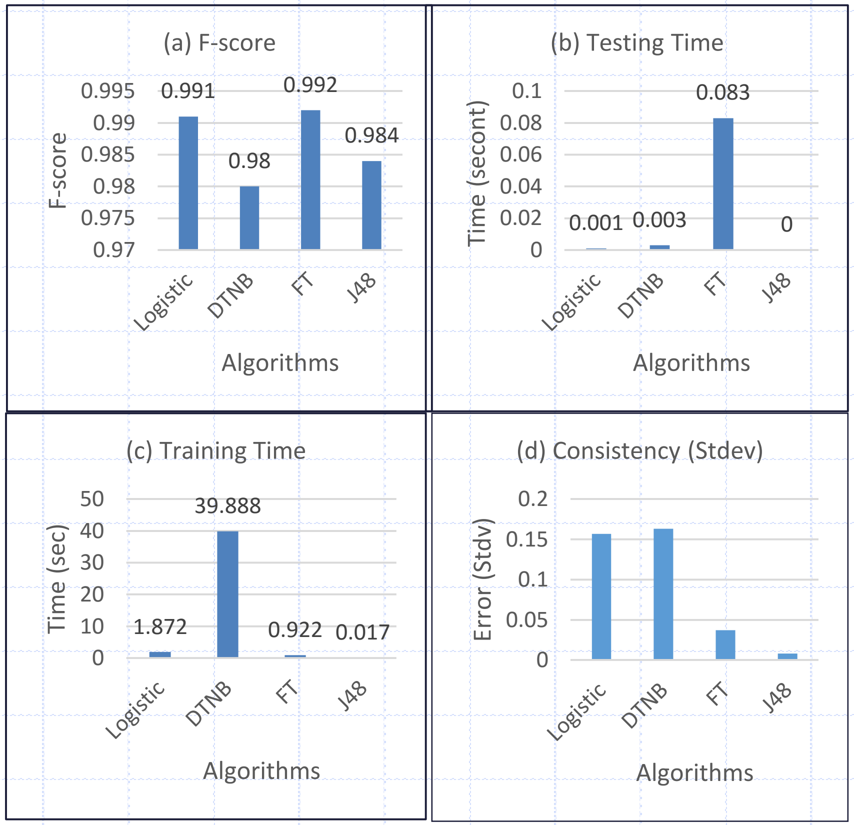
[13] Ali R, et al.,. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

Underlying Technologies : Multi-criteria Decision Making (Solution B)



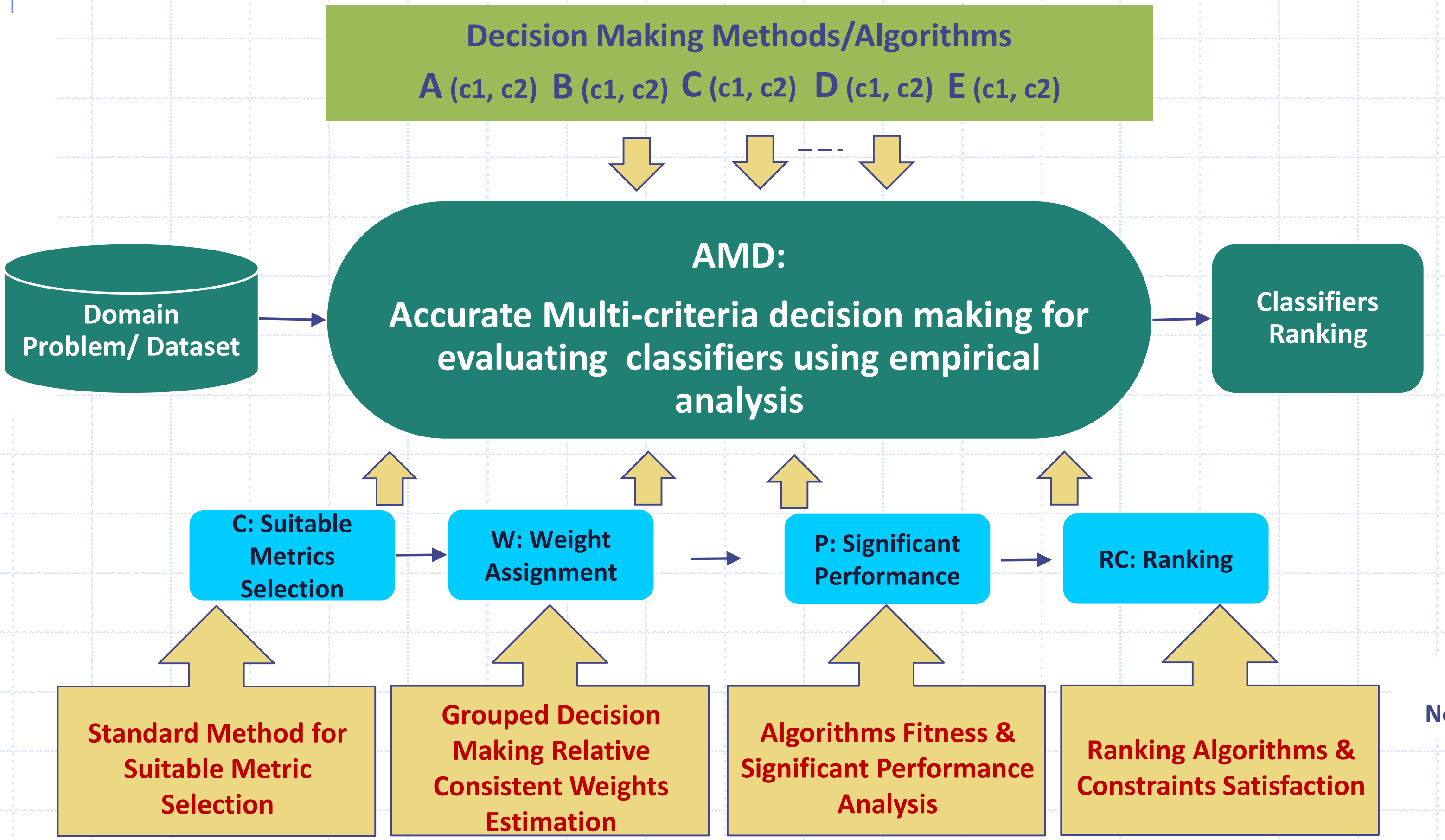
Dataset	Meta-features					Best Algorithm (s)
d1	mf1	mf2	mf3	…	mfn	A3
d2	…	…	…	…	…	A2
…	…	…	…	…	…	A1, A3

Algorithms: Logistic, DTNB, FT, J48

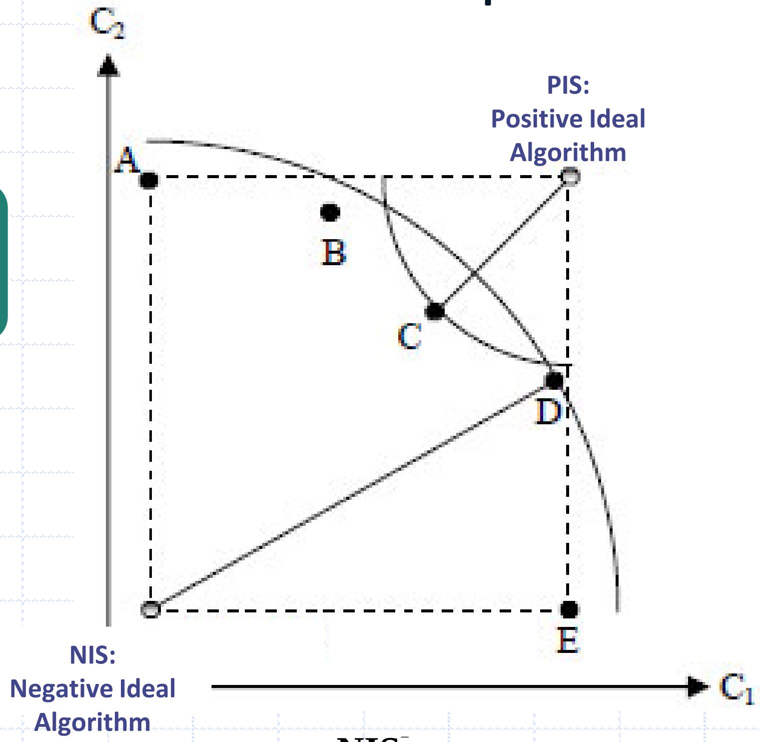


Performance Metrics: F-score, Testing Time, Training Time, Consistency (Stdev)

Empirical Performance Analysis of Classifiers (Solution B)



Basic Concept

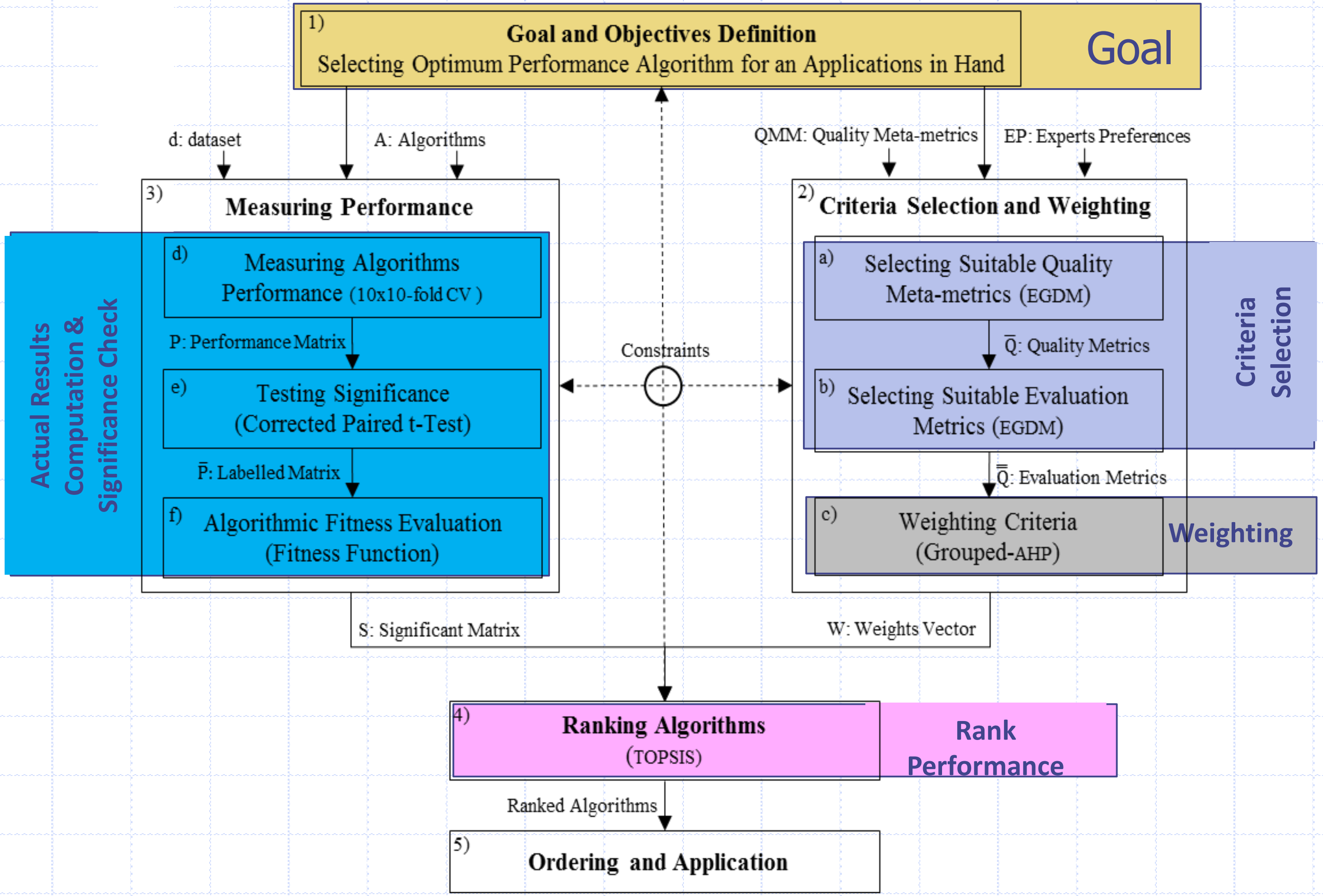


$$RC_i^* = \frac{NIS_i^-}{PIS_i^* + NIS_i^-},$$
$$where, i = 1,2,\ldots,m$$

Existing	Missing, support limited criteria	Absolute weighting	Only significance test	Limited constraints	Existing

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)



Algorithm 1. Selection of optimum performance algorithm

Begin

inputs: d – the given dataset
$A = \{a_1, a_2, \dots, a_n\}$ // n algorithms

output: R = top-k algorithms; where, $R \subseteq A$

Let QMM = Classifiers quality meta-metrics.

1 [Define Goal]
$G = \{o_1, o_2, \dots, o_n\}$; // where, G stands for goal.

2 [Select Suitable Quality Meta-metrics]
$\bar{Q}$ = selectSuitQuality(QMM, G); //where, $\bar{Q}$ quality metrics.

3 [Select Suitable Evaluation Metrics]
$\bar{\bar{Q}}$ = selectSuitEvalMetrics($\bar{Q}$, G); //where, $\bar{\bar{Q}} \subseteq \bar{Q}$, metrics.

4 [Estimate Relative Weight]
W = estimateRelativeWeights($\bar{\bar{Q}}$); //where W relative weight.

5 [Generate Performance Results of the Algorithms]
foreach algorithm **a** in **A** perform
a. P = algPerformanceEval(d, a, $\bar{\bar{Q}}$);
end for

6 [Perform Statistical Significance]
$\bar{P}$ = performStatSigTest(P); // $\bar{P}$ is significance matrix.

7 [Perform Algorithm Fitness]
S = Perform Algorithm Fitness Test ;

8 [Compute Relative Closeness to Ideal Algorithm]
RC^* = rankAlgorithms(S, W); //where, RC^* relative closeness.

9 [Rank the Algorithms]
RankedList = RANK.AVG(RC^*_1, RC^*_1: RC^*_n, 1);

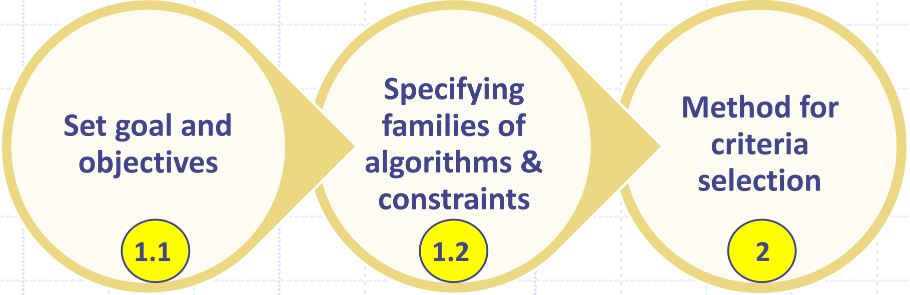
10 [Select Top-K Algorithms]
R = selectTopK(RankedList, k);

11 apply R to learn d

End

Empirical Performance Analysis of Classifiers (Solution B)

1-2(a, b) **Standard Method for Suitable Meta-metrics and Criteria Selection**



1.1 Selection of **optimum** performance **consistent classification algorithms**

1.2 Algorithms should come from the **heterogeneous** families of **multi-classification algorithms**

1.1 Optimum Performance Consistent Algorithm for Multi-class Classification

1.2

SNO	Classifier	SNO	Classifier
1	bayes.BayesNet	19	trees.J48
2	bayes.NaiveBayes	20	trees.J48graft
3	bayes.NaiveBayesUpdateable	21	trees.LADTree
4	functions.Logistic	22	trees.RandomForest
5	functions.RBFNetwork	23	trees.RandomTree
6	functions.SMO	24	trees.REPTree
7	misc.HyperPipes	25	trees.SimpleCart
8	misc.VFI	26	meta.AdaBoostM1
9	rules.ConjunctiveRule	27	meta.Bagging
10	rules.DecisionTable	28	meta.Dagging
11	rules.DTNB	29	meta.END
12	rules.JRip	30	meta.FilteredClassifier
13	rules.OneR	31	meta.LogitBoost
14	rules.PART	32	meta.RacedIncrementalLogitBoost
15	rules.Ridor	33	meta.RandomSubSpace
16	rules.ZeroR	34	meta.Stacking
17	trees.BFTree	35	meta.Vote
18	trees.FT		

Six Heterogeneous Families of Multi-Classification Classifiers from Weka Environment

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

2a Standard Method for Suitable Meta-metrics and Criteria Selection

(i) Building Classifiers Quality Meta-metrics (QMM) Classification Model

Classifiers Quality Meta-Metrics (QMM)

- ✓ Correctness (cor)
  - Accuracy ('+'cor) — **Examples**: percent correct, precision, recall, F measure etc.
  - Accuracy ('-'cor) - error metrics — **Examples**: percent incorrect, FPR, FP, TN etc.
- ✓ Complexity (complex)
  - Computational (ccom) — **Examples**: Elapsed Time training, User CPU Time training etc.
  - Memory/Space (scom) — **Examples**: Num Rules, Tree Size, Num Leaves etc.
- ✓ Responsiveness (res) — **Examples**: Elapsed time testing, UserCPU time testing
- ✓ Consistency (con) — **Examples**: It is a constant that can be used as a metric. Standard Deviation is used to measure it
- Comprehensibility (com) — **Examples**: Measures Interestingness and Interpretability, e.g., Num. Rules, Tree Size etc.
- Reliability (rel)
  - Information-Theoritic (irel) — **Examples**: Entropy, entropy gain etc.
  - Distance or Error Measure (erel) — **Examples**: MAR, RMSE etc.
- Robustness (rob) — **Examples**: Measure sensitivity in terms of True positive rate
- Separability (sep) — **Examples**: Graphical measures that best visualize the results in binary classification, e.g., ROC, AUC etc.

Advantage
- Physical Meaning of the Classifiers
- Easy understanding

(ii) Experts' Group-based Decision Making for QMM Selection

1. **Select Salient QMM** — Select salient qualities , compliant to Goal, from QMM by each ML expert
2. **Vote/Rate QMM** — Vote for salient QMM using rating method and select Top-k
3. **Repeat Vote, if Inconsistency** — Repeat step 2, if there is conflict or inconsistencies

Procedure 1. selectSuitQuality

Begin

inputs: QMM – the set of classifiers quality meta-metrics
G – the goal

output: Q'' – the set of highly rated/ranked quality meta-metrics

1 [Select key qualities by each expert]
$Q = \text{extractSalientQMM}(\text{QMM}, G)$; //where, $Q \subseteq \text{QMM}$

2 [Vote each selected quality by all the experts]
$Q' = \text{preliminaryVoteAggQuality}(Q')$; //where, Q' is initial list of selected QMM

a. If Q' contains Consistent qualities, then
i. $Q'' = \text{selectTopKQMM}(Q', k)$; // where, $k \subseteq Q$ and stands for number of qualities the experts are interested to select
ii. goto setp 3;

b. Else
i. Repeat step 2;

3 return $= Q''$;

End

Advantage
- Experts Consensus Will Satisfy Goal

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

2b Standard Method for Suitable Meta-metrics and Criteria Selection

(2b) Suitable Evaluation Metrics Selection

Classifiers Evaluation Metrics (e.g., 51 Metrics from Weka)

Id	Evaluation Metric	QMM	Sub-QMM	Id	Metric	QMM	Sub-QMM
1	Number_correct	cor	+cor	27	Elapsed_Time_training	complex	ccom
2	Percent_correct	cor	+cor	28	UserCPU_Time_training	complex	ccom
3	Kappa_statistic	cor	+cor	29	~~measureNumRules~~	~~complex, com~~	~~scom~~
4	True_positive_rate	cor	+cor	30	~~measurePercentAttsUsedByDT~~	~~complex, com~~	~~scom~~
5	Num_true_positives	cor	+cor	31	~~measureTreeSize~~	~~complex, com~~	~~scom~~
6	False_negative_rate	cor	+cor	32	~~measureNumLeaves~~	~~complex, com~~	~~scom~~
7	Num_false_negatives	cor	+cor	33	~~measureNumPredictionLeaves~~	~~complex, com~~	~~scom~~
8	IR_precision	cor	+cor	34	~~measureNodesExpanded~~	~~complex, com~~	~~scom~~
9	IR_recall	cor	+cor	35	Elapsed_Time_testing	res	ures
10	F_measure	cor	+cor	36	UserCPU_Time_testing	res	sres
11	Weighted_avg_true_positive_rate	cor	+cor	37	~~SF_prior_entropy~~	~~rel~~	~~irel~~
12	Weighted_avg_false_negative_rate	cor	+cor	38	~~SF_scheme_entropy~~	~~rel~~	~~irel~~
13	Weighted_avg_IR_precision	cor	+cor	39	~~SF_entropy_gain~~	~~rel~~	~~irel~~
14	Weighted_avg_IR_recall	cor	+cor	40	~~SF_mean_prior_entropy~~	~~rel~~	~~irel~~
15	Weighted_avg_F_measure	cor	+cor	41	~~SF_mean_scheme_entropy~~	~~rel~~	~~irel~~
16	Number_incorrect	cor	-cor	42	~~SF_mean_entropy_gain~~	~~rel~~	~~irel~~
~~17~~	~~Number_unclassified~~	~~cor~~	~~-cor~~	43	~~KB_information~~	~~rel~~	~~irel~~
~~18~~	~~Percent_incorrect~~	~~cor~~	~~-cor~~	44	~~KB_mean_information~~	~~rel~~	~~irel~~
~~19~~	~~Percent_unclassified~~	~~cor~~	~~-cor~~	45	~~KB_relative_information~~	~~rel~~	~~irel~~
~~20~~	~~False_positive_rate~~	~~cor~~	~~-cor~~	46	~~Mean_absolute_error~~	~~rel~~	~~erel~~
~~21~~	~~Num_false_positives~~	~~cor~~	~~-cor~~	47	~~Root_mean_squared_error~~	~~rel~~	~~erel~~
~~22~~	~~True_negative_rate~~	~~cor~~	~~-cor~~	48	~~Relative_absolute_error~~	~~rel~~	~~erel~~
~~23~~	~~Num_true_negatives~~	~~cor~~	~~-cor~~	49	~~Root_relative_squared_error~~	~~rel~~	~~erel~~
~~24~~	~~Weighted_avg_false_positive_rate~~	~~cor~~	~~-cor~~	50	Area_under_ROC	sep, cor	-
~~25~~	~~Weighted_avg_true_negative_rate~~	~~cor~~	~~-cor~~	51	Weighted_avg_area_under_ROC	sep, cor	-
26	True_positive_rate	cor, rob	+cor		--	--	--

(2b) Experts' Group-based Decision Making for Evaluation Metrics Selection

- Like QMM Selection, **Experts Group-based Consensus Method** is used
- General Guidelines
  - Goal' constraints must be satisfied
  - Conflicting and duplicate metrics should be avoided
- The selected criteria and the reasons behind are:
  - ***Wgt. Avg. F–score***
    - Satisfies multiclass constraints "weighted" accounts for class imbalance by computing the average of binary metrics in which each class's score is weighted by its presence in the true data sample
  - ***CPU_Time_training***
    - Satisfies the global applicability condition of classifiers and applicable to every algorithms. Shared among all families (heterogeneous) of classifiers
  - ***CPU_Time_testing***
    - Satisfies heterogeneity constraint of classifiers and measure the efficiency of algorithms in terms of response time
  - ***Standard deviation (Stdv)*** – ***Avg.Stdev*** of the above ***metrics***
    - Satisfies the obligatory constraint of consistency measure of each classifier

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

2c Group Decision-making for Relative Consistent Weighting

- Analytical hierarchy process [15] of relative weighting is used.
  1. Comparison matrix *E* with 4 experts' preferences is used as shown below.

Criteria	WgtAvgF-score	CPUTimeTesting	CPUTimeTraining	Consistency	Weights
WgtAvgF-score	1	5	7	4	0.60
CPUTimeTesting	0.20	1	4	1/2	0.14
CPUTimeTraining	0.14	0.25	1	1/5	0.05
Consistency	0.25	2.00	5	1	0.21
				CI:0.042	**1.00**

  2. Each value of the matrix is normalized as below

$$\bar{e}_{ij} = e_{ij} / \sum_{i=1}^{m} e_{ij}$$

  3. Criteria weight vector $W = w_j$ is computed using

$$w_j = \sum_{j=1}^{m} \bar{e}_{ij}/m = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix}$$

  4. Consistency of the estimated weights are checked using $CR = CI/RI$

where,

1. $\mathrm{CI} = (\lambda_{\max} - \mathrm{n})/(\mathrm{n}-1)$
2. $\lambda_{\max} = \left(\sum_{i=1}^{m} \mathrm{Cv_{ij}}\right)/\mathrm{m}$ (principal eigenvalue)
3. $\mathrm{Cv_{ij}} = \mathrm{E} * \mathrm{W}$ (consistency vector CV)
4. RI is taken from the Saaty's preference scale

1	2	3	4	5	6	7	8	9	10	11
0.00	0.00	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51

5. If CR < 0.10
   1. Weights are consistent and the judgments is correct
6. Else
7. Repeat Relative Weight Estimation Algorithmic and change the preferences

Grouped-based Relative Criteria Weighting

$$\mathrm{groupedWeight} = \sum_{e=1}^{m} \left(\prod_{dm=1}^{n} \mathrm{DMWeight} * \mathrm{EMWeight}\right)$$

Empirical Performance Analysis of Classifiers (Solution B)

2c Group Decision-making for Relative Consistent Weighting

Decision Makers' Decision Weightage

(a). Experts (decision makers) prioritization

DM Prioritization	DM#1	DM#2	DM#3	DM#4	DM decision weigh
DM#1	1	3	2	5	0.49
DM#2	0.33	1	1	3	0.21
DM#3	0.50	1.00	1	3	0.23
DM#4	0.20	0.33	0.33	1	0.08
				CI: 0.009	1.00

Individual Decision Maker's Assigned Weight

(b) DM#1 relative weighting

Criteria	WgtAvgF-score	CPUTimeTesting	CPUTimeTraining	Consistency	Weights
WgtAvgF-score	1	8	9	7	0.70
CPUTimeTesting	0.13	1	3	1/2	0.09
CPUTimeTraining	0.11	0.33	1	1/5	0.04
Consistency	0.142857143	2.00	5	1	0.16
				CI:0.050	1.00

(c) DM#2 relative weighting

Criteria	WgtAvgF-score	CPUTimeTesting	CPUTimeTraining	Consistency	Weights
WgtAvgF-score	1	7	9	5	0.68
CPUTimeTesting	0.14	1	2	1	0.12
CPUTimeTraining	0.11	0.50	1	1/3	0.06
Consistency	0.2	1.00	3	1	0.14
				CI:0.012	1.00

(d) DM#3 relative weighting

Criteria	WgtAvgF-score	CPUTimeTesting	CPUTimeTraining	Consistency	Weights
WgtAvgF-score	1	7	8	6	0.68
CPUTimeTesting	0.14	1	2	1/2	0.10
CPUTimeTraining	0.13	0.50	1	1/3	0.06
Consistency	0.17	2.00	3.00	1	0.16
				CI:0.021	1.00

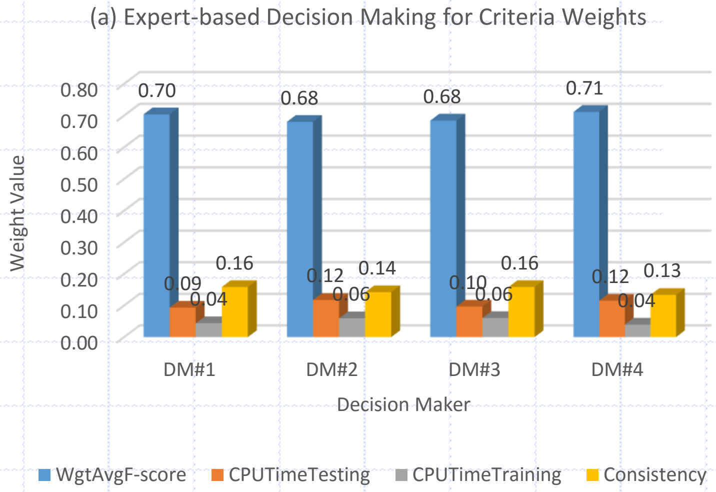
(e) DM#4 relative weighting

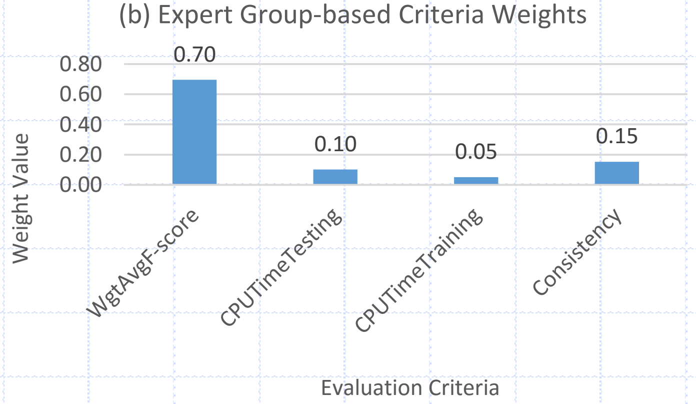
Criteria	WgtAvgF-score	CPUTimeTesting	CPUTimeTraining	Consistency	Weights
WgtAvgF-score	1	8	9	8	0.71
CPUTimeTesting	0.13	1	4	1	0.12
CPUTimeTraining	0.11	0.25	1	1/6	0.04
Consistency	0.13	1.00	6	1	0.13
				CI:0.073	1.00

(f) Criteria weights based on group decision making

DM Decision Prior	0.49	0.21	0.23	0.08	
Criteria\DM	DM#1	DM#2	DM#3	DM#4	Weight
WgtAvgF-score	0.70	0.68	0.68	0.71	0.70
CPUTimeTesting	0.09	0.12	0.10	0.12	0.10
CPUTimeTraining	0.04	0.06	0.06	0.04	0.05
Consistency	0.16	0.14	0.16	0.13	0.15
					1.00

Group Decision of all the Decision Makers





[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

3d Algorithms Performance

- Real performance results for criteria
- Weka, 10x10-fold cross-validation for stable performance

Procedure 4. algPerformanceEval.

Begin
inputs: d – the given dataset
a – the given classification algorithm
$\overline{\overline{Q}} = \{e_1, e_2, \ldots, e_m\}$ – the set of evaluation metrics
output: p– performance matrix of algorithm a on dataset d for the evaluation metrics $\overline{\overline{Q}}$;
Let ITER = number of iteration
F = number of folds
Performance= 1*m matrix for storing the performance results of algorithm a on dataset d for the metrics $\overline{\overline{Q}}$
1. $\mathrm{ITER} = 10; \mathrm{F} = 10;\ \mathrm{Performance} = 0;$
2. **for** i = 1 **to** ITER **perform**
3. generate F FOLD from d; //generate 10-fold from dataset d
4. **for** f = 1 **to** F **perform**
   a. TestData = FOLD [f]; //create test dataset
   b. TrainData = d − TestData; //create train dataset
   c. Model = buildModel(TrainData, a); // build the classification model
   d. $\mathrm{Performanance} = \mathrm{Performance} + \mathrm{addPerformance}\left(\mathrm{testModel}\left(\mathrm{TestData}, \mathrm{Model}, \overline{\overline{Q}}\right)\right);$
   end for
   end for
1. $p = {}^{\mathrm{Performanance}}/_{(\mathrm{ITER}*\mathrm{F})}$
2. return (p)
End

Algorithms	F-score	CPUTimeTraining	CPUTimeTesting	Consistency
bayes.BayesNet	0.78*	0.027*	0.002	0.013
bayes.NaiveBayes*	0.825*	0.013*	0.008*	0.010
bayes.NaiveBayesUpdateable*	0.825*	0.011*	0.01*	0.011
functions.Logistic	0.836	0.229*	0.000	0.012
functions.RBFNetwork	0.733*	0.232*	0.004	0.043
functions.SMO	0.830	1.99*	(ref) **0.000**	0.041
misc.HyperPipes	0.66*	(ref) **0.001**	0.000	0.005
misc.VFI	0.716*	0.008*	0.004	0.012
rules.ConjunctiveRule	0.645*	0.043*	0.000	0.006
rules.DecisionTable	0.829	1.086*	0.000	0.043
rules.DTNB	0.832	88.16*	0.004	2.611
rules.JRip	0.825*	0.648*	0.000	0.067
rules.OneR	0.739*	0.014*	0.000	0.007
rules.PART	0.819*	1.161*	0.001	0.057
rules.Ridor	0.795*	0.453*	0.000	0.034
rules.ZeroR	0.645*	0.000	0.000	0.001
trees.BFTree	0.838	0.79*	0.000	0.024
trees.FT	0.827	1.38*	0.161*	0.044
trees.J48	0.828	0.221*	0.000	0.014
trees.J48graft	0.829	0.29*	0.000	0.014
trees.LADTree	0.833	1.676*	0.000	0.020
trees.RandomForest	0.837	2.304*	0.022*	0.022
trees.RandomTree	0.791*	0.028*	0.001	0.009
trees.REPTree	0.835	0.084*	0.000	0.012
trees.SimpleCart	0.836	0.713*	0.000	0.021
meta.AdaBoostM1	0.822*	1.074*	0.001	0.021
meta.Bagging	(ref) **0.842**	0.753*	0.000	0.013
meta.Dagging*	0.824*	0.013*	0.107*	0.010
meta.END	0.828	0.215*	0.003	0.013
…	…	…	…	…

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

3(e-f) Significance Test and Fitness Evaluation

- Significance test
  - **Corrected paired t-test** method [16] is used for checking statistical significance
  - Reference classifier
    - Highest score (for **benefit criteria**)
    - Lowest score (for **cost criteria**)
  - Algorithms labeled as **either best, poor or equal**
- Significance fitness evaluation function
  - Input: significance matrix
  - Output: **significantly fit algorithms**

	Algo. A	Algo. B
F-score	A*	B*
Train-time	A*	B
Test-time	A*	B v

A is significantly unfit

Algorithm Statistical Significance Test

Procedure 5. performStatSigTest

Begin
inputs: P – performance matrix
output: $\bar{p}$ – m*n performance matrix, where m is the number of evaluation metrics and n is the number of algorithms;
Let d – given dataset
$\underline{A} = \{a_1, a_2, .., a_n\}$ – set of classification algorithms
$\bar{\bar{Q}} = \{e_1, e_2, .., e_m\}$ – set of evaluation metrics
1. **foreach e** $\in \bar{Q}$ in the performance matrix **P** for a dataset **d**
  a. **if e** $\in$ benefit metric
    i. **referenceAlg** = selectReferenceAlg(maxPerformValue(**e**));
  b. **else**
    i. **referenceAlg** = selectReferenceAlg(minPerformValue(**e**));
  c. $\bar{p}$ = performCorrectedPairedtTest(**referenceAlg**, **P**, **e**);
2. **end for**
3. **Return** ($\bar{P} = \bar{p}$)
End

Algorithm fitness evaluation function

$$S = \{\forall_{a \in A} : a \in \bar{P} \mid \forall_e : e \in \bar{\bar{Q}}.\ {\sim}\mathrm{nonSignificant}(e)\}$$

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

4 Ranking

TOPSIS Method for Ranking [17]

Procedure 6. rankAlgorithms

Begin

inputs: S – n*m matrix containing significant algorithms
W – 1*m (single row) weight vector

output: RC – n*1 (single column) matrix of the relative closeness score

Let d – given dataset
$A = \{a_1, a_2, .., a_n\}$ – set of classification algorithms
$\bar{\bar{Q}} = \{e_1, e_2, .., e_m\}$ – set of evaluation metrics

1 [create the evaluation matrix from the significant matrix S]
$S = (s_{ij})_{m*n}$; //where, s_{ij} represents the value of algorithm i for the evaluation metric j

2 [normalize the evaluation matrix S]
Define local/implicit constraints on $\bar{\bar{Q}} = \{e_1, e_2, .., e_m\} \in S$;

3 [normalize the evaluation matrix S]
$\bar{S} = r_{ij} = s_{ij} / \sqrt{\sum_{i=1}^{m} s_{ij}^2}$; //where, i =1, 2, ..., n and j = 1, 2, ..., m with n is the number of algorithm and m is the number of evaluation metrics

4 [compute weighted normalized decision matrix V with each value v_{ij}]
$V = (r_{ij})_{m*n} = r_{ij} * W_j$; //where, W_j is the weight vector

5 [compute positive ideal (PIS) and negative ideal (NIS) solutions]

a. $PIS = \{(\max_i v_{ij} \mid j \in C_b), (\min_i v_{ij} \mid j \in C_c)\} = \{v_j^* \mid j = 1,2,\ldots,n\}$

b. $NIS = \{(\min_i v_{ij} \mid j \in C_b), (\max_i v_{ij} \mid j \in C_c)\} = \{v_j^- \mid j = 1,2,\ldots,n\}$

6 [compute the separation measures using the m-dimensional Euclidean distance]

a. $PIS_i^* = \sqrt{\sum_{j=1}^{m} (v_{ij} - v_j^*)^2}, \; j = 1,2,\ldots,m$

b. $NIS_i^- = \sqrt{\sum_{j=1}^{m} (v_{ij} - v_j^-)^2}, \; j = 1,2,\ldots,m$

7 [compute relative closeness RC of the algorithm to the ideal algorithm]

Empirical Performance Analysis of Classifiers (Solution B)

Results and Evaluation

- Experimental setup
  - Dataset
    - **Fifteen (15) OpenML [18] UCI Library [19]**
  - Tools and Library
    - Weka [20], DAME AHP [21], SANNA 2014
  - Environment
    - Win. PC CPU(3.3 GHz) and RAM 8GB.
  - Algorithms
    - **Thirty five (35)** Weka classifiers

Datasets

Datasets	Characteristics of Datasets							Domain
	Attributes	NominalAtts	NumericAtts	BinaryAtts	Classes	InstanceCount	Missing	
abalone-3class	9	1	7	0	3	4177	0	Biology
rabe-148	9	1	7	0	3	4177	0	Synthetic
acute-inflammations-nephr	6	0	5	0	2	66	0	Medical
ADA_Agnostic	7	5	1	5	2	120	0	Business
ADA_Prior	49	0	48	0	2	4562	0	Business
adult-4000	15	8	6	1	2	4562	88	Social Studies
adult-8000	15	8	6	1	2	3983	0	Social Studies
aileron	15	8	6	1	2	8000	0	nil
analcatdata-AIDS	41	0	40	0	2	5795	0	AIDS
analcatdata-apnea2	5	2	2	0	2	50	0	book
analcatdata-apnea2	4	2	1	0	2	475	0	book
analcatdata-asbestos	4	2	1	0	2	475	0	book
analcatdata-authorship	4	2	1	1	2	83	0	Research
analcatdata-bankruptcy	71	0	70	0	4	841	0	Finance
analcatdata-birthday	7	1	5	0	2	50	0	Social Studies

Algorithms

SNO	Classifier	SNO	Classifier
1	bayes.BayesNet	19	trees.J48
2	bayes.NaiveBayes	20	trees.J48graft
3	bayes.NaiveBayesUpdateable	21	trees.LADTree
4	functions.Logistic	22	trees.RandomForest
5	functions.RBFNetwork	23	trees.RandomTree
6	functions.SMO	24	trees.REPTree
7	misc.HyperPipes	25	trees.SimpleCart
8	misc.VFI	26	meta.AdaBoostM1
9	rules.ConjunctiveRule	27	meta.Bagging
10	rules.DecisionTable	28	meta.Dagging
11	rules.DTNB	29	meta.END
12	rules.JRip	30	meta.FilteredClassifier
13	rules.OneR	31	meta.LogitBoost
14	rules.PART	32	meta.RacedIncrementalLogitBoost
15	rules.Ridor	33	meta.RandomSubSpace
16	rules.ZeroR	34	meta.Stacking
17	trees.BFTree	35	meta.Vote
18	trees.FT		

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

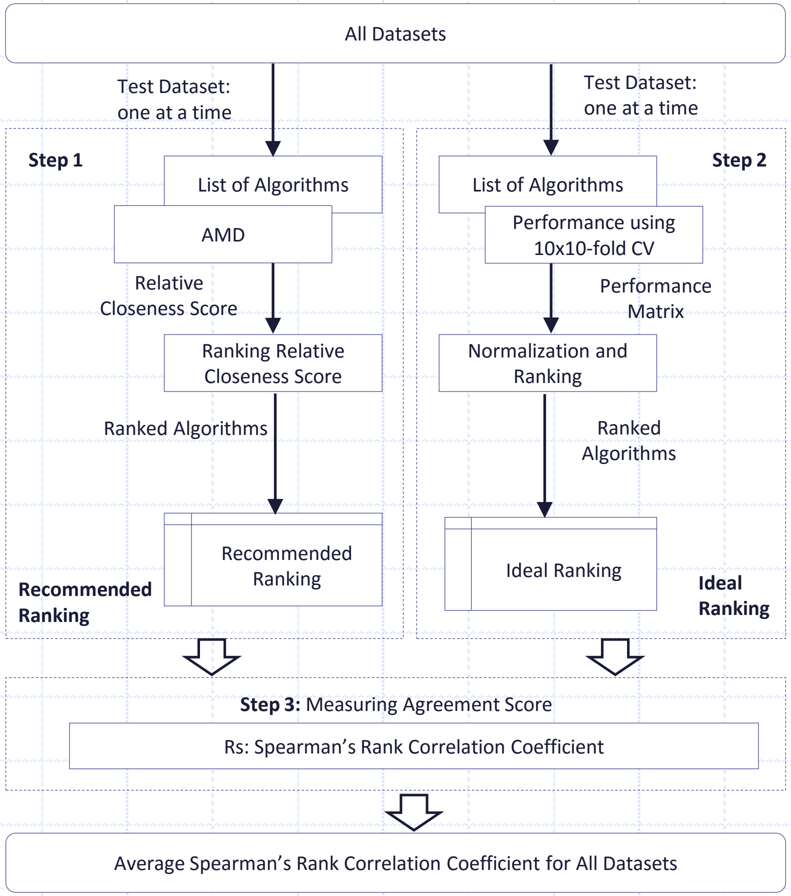
Empirical Performance Analysis of Classifiers (Solution B)

Evaluation methodology, criteria and Experiments

Experiments

- Experiment 1
  - **Correctness** average Spearman's correlation coefficient with Ideal Ranking
- Experiment 2
  - **Sensitivity** and Consistency **analysis**
- Experiment 3
  - **Significance fitness** evaluation

Evaluation Methodology



Evaluation Criteria

(Spearman's Rank Correlation Coefficient)

$$R_s = 1 - \frac{6(ir - rr)^2}{n * (n^2 - 1))}$$

Where, *ir* and *rr* are the ideal and recommended ranking value values while *n* is the number of algorithms used in the comparison

(Neave & Worthington, 1992)[24]

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

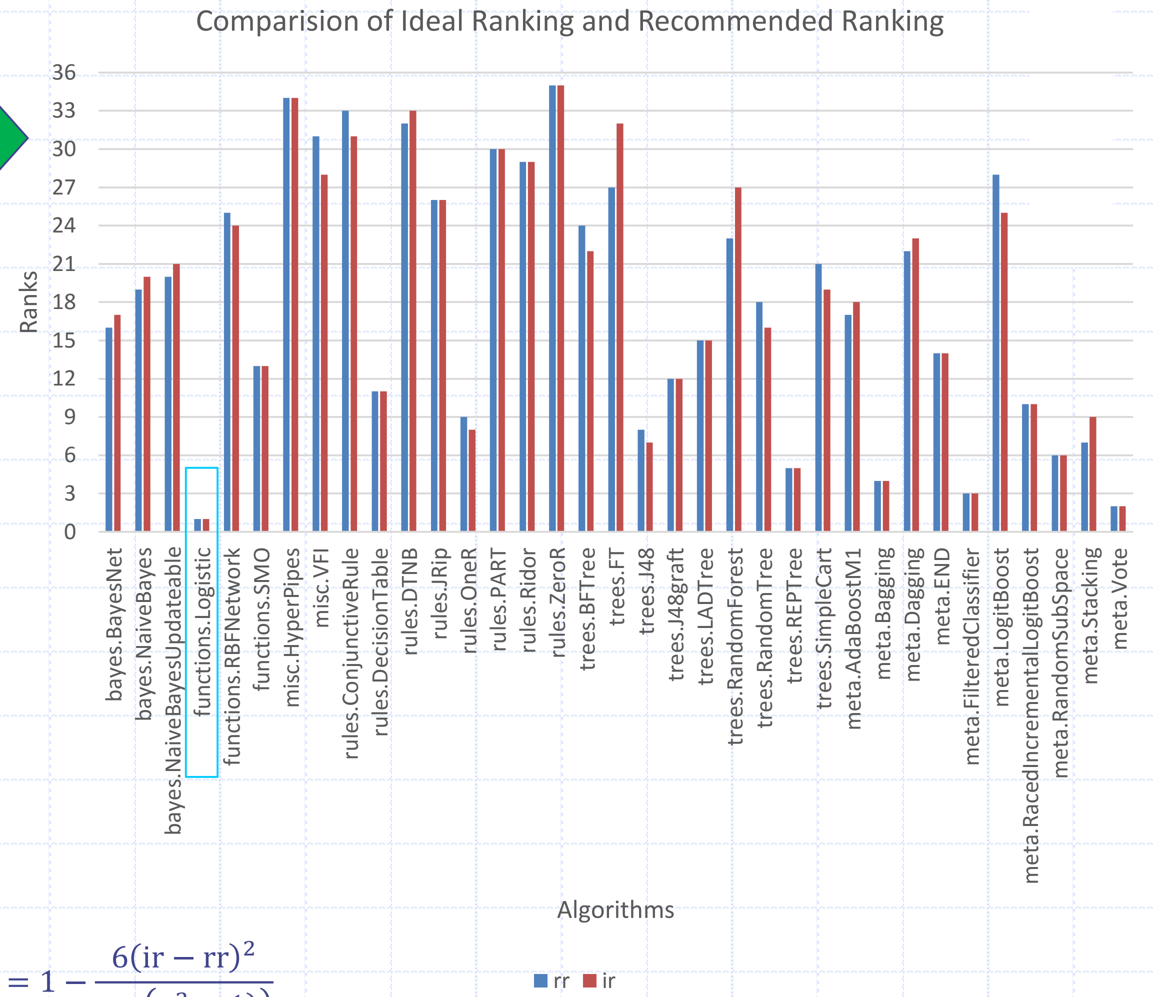
Empirical Performance Analysis of Classifiers (Solution B)

Experiment 1 (correctness)

Dataset ID	Dataset Name	Rs (Spearman's Rank Corelation Coeffecient
1	abalone-3class	0.988
2	rabe-148	0.985
3	acute-inflammations-nephr	0.994
4	ADA_Agnostic	0.990
5	ADA_Prior	0.991
6	adult-4000	0.983
7	adult-8000	0.975
8	aileron	0.979
9	analcatdata-AIDS	0.983
10	analcatdata-apnea2	0.932
11	analcatdata-apnea2	0.963
12	analcatdata-asbestos	0.973
13	analcatdata-authorship	0.999
14	analcatdata-bankruptcy	0.983
15	analcatdata-birthday	0.969
Average Spearman's Rank Correlation Coefficient		0.979

Algorithm	rr	ir	(ir-rr)	(ir-rr)^2
bayes.BayesNet	16	17	1	1
bayes.NaiveBayes	19	20	1	1
bayes.NaiveBayesUpdateable	20	21	1	1
functions.Logistic	1	1	0	0
functions.RBFNetwork	25	24	-1	1
functions.SMO	13	13	0	0
misc.HyperPipes	34	34	0	0
misc.VFI	31	28	-3	9
rules.ConjunctiveRule	33	31	-2	4
rules.DecisionTable	11	11	0	0
rules.DTNB	32	33	1	1
rules.JRip	26	26	0	0
rules.OneR	9	8	-1	1
rules.PART	30	30	0	0
rules.Ridor	29	29	0	0
rules.ZeroR	35	35	0	0
trees.BFTree	24	22	-2	4
trees.BFTree	27	32	5	25
trees.J48	8	7	-1	1
trees.J48graft	12	12	0	0
trees.LADTree	15	15	0	0
trees.RandomForest	23	27	4	16
trees.RandomTree	18	16	-2	4
trees.REPTree	5	5	0	0
trees.SimpleCart	21	19	-2	4
meta.AdaBoostM1	17	18	1	1
meta.Bagging	4	4	0	0
meta.Dagging	22	23	1	1
meta.END	14	14	0	0
meta.FilteredClassifier	3	3	0	0
meta.LogitBoost	28	25	-3	9
meta.RacedIncrementalLogitBoost	10	10	0	0
meta.RandomSubSpace	6	6	0	0
meta.Stacking	7	9	2	4
meta.Vote	2	2	0	0
			sum	88
			Rs (k=35)	0.987

Weights: F-score (0.70), TrainTime (0.05), TestTime (0.10), Consistency (0.15)



$$Rs = 1 - \frac{6(\mathrm{ir} - \mathrm{rr})^2}{n * (n^2 - 1)}$$

[24] Ali R, et. al.. “An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)”. Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

Experiment 2

- Sensitivity and Consistency Analysis
- To **determine consistency** with **varying parameters**'
- **Changed weight one at a time**, i.e., interchange highest weight 0.70 with each criteria

		Sensitivity Analysis			
Dataset ID	Dataset\Weights, k=35	Rs. Wgt.Avg F-score (0.70,0.05,0.10,0.15)	Rs. CPUTimeTraining (0.05,0.70,0.10,0.15)	Rs. CPUTimeTesting (0.05,0.10,0.70, 0.15)	Rs. Consistency (0.05,0.10,0.15,0.70)
1	abalone-3class	0.454	0.913	0.523	0.999
2	rabe-148	0.904	0.758	0.500	0.992
3	acute-inflammations-nephr	0.858	0.798	0.501	0.979
4	ADA_Agnostic	0.880	0.368	0.819	0.433
5	ADA_Prior	0.295	0.943	0.565	0.985
6	adult-4000	0.276	0.890	0.599	0.979
7	adult-8000	0.488	0.792	0.670	0.943
8	aileron	0.946	0.223	0.806	0.563
9	analcatdata-AIDS	0.654	0.766	0.500	0.995
10	analcatdata-apnea2	0.107	0.844	0.652	0.986
11	analcatdata-apnea2	0.158	0.936	0.618	0.972
12	analcatdata-asbestos	0.508	0.838	0.500	0.999
13	analcatdata-authorship	0.880	-0.265	0.738	-0.074
14	analcatdata-bankruptcy	0.945	0.863	0.543	0.998
15	analcatdata-birthday	-0.506	0.777	0.618	0.990
	Average Spearman's Rank Correlation (Rs)	**0.523**	**0.696**	**0.610**	**0.849**

Results Interpretation

- The highlighted values shows negative/very weak correlation and are not significant
- On average, the correlation is positive showing consistent results with significance of α=0.005-0.002

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

Experiment 3

- Significance fitness evaluation function
- Purpose of the experiment
  - To find out algorithms that are non-significant on all criteria
- Interpretation
  - Three probabilistic, two decision tree and two meta-leaning algorithms performed poorly on all three criteria
  - If they are not get excluded prior to ranking, the results are affected

Algorithm	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 13
bayes.BayesNet	26	4	2	7	27	**4**
bayes.NaiveBayes	**19**	11	12	21	**30**	**7**
bayes.NaiveBayesUpdateable	**20**	10	15	20	**31**	8
trees.FT	30	**32**	**32**	**32**	25	2
trees.RandomForest	17	**25**	23	**24**	**17**	6
meta.Dagging	**27**	18	21	26	**32**	**30**

- bays.BayesNet is ranked 4th and bayes.NaiveBayes as 7th.
- Same is for other

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

Empirical Performance Analysis of Classifiers (Solution B)

Comparison

- Ranking Learning Algorithms on Accuracy and Time [6]

$$ARR = \frac{\dfrac{SR_{ap}^{di}}{SR_{aq}^{di}}}{1 + \propto * \log\left(\dfrac{T_{ap}^{di}}{T_{aq}^{di}}\right)}$$

- Recommendation of classification algorithms eristics [7]

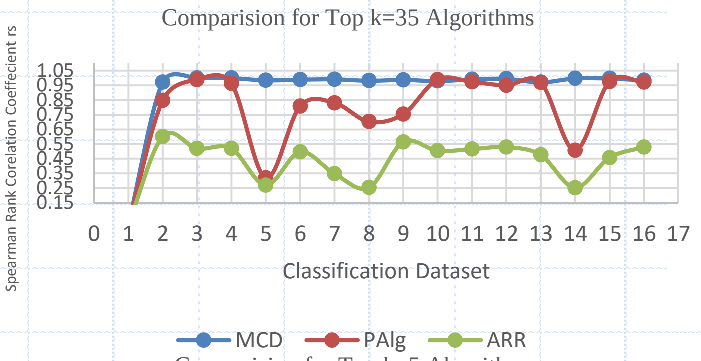
$$P_{Alg,D} = \frac{Accuracy_{Alg,D}}{1 + \propto * \log(RTime_{Alg,D})}$$

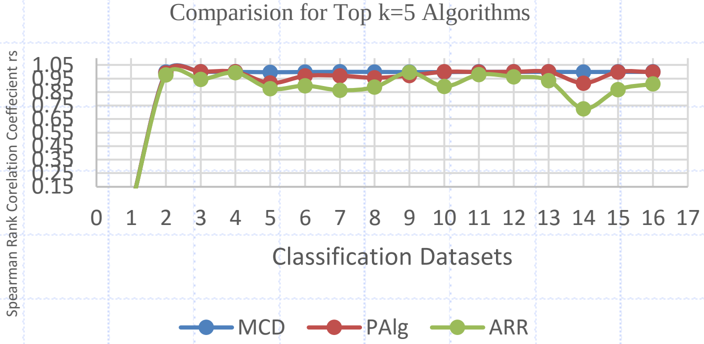
- Where, ∝ = 0.1, 1, 10 for 10% preference of accuracy, equal preferences for both and 10% preference of time
- Proposed AMD [24]

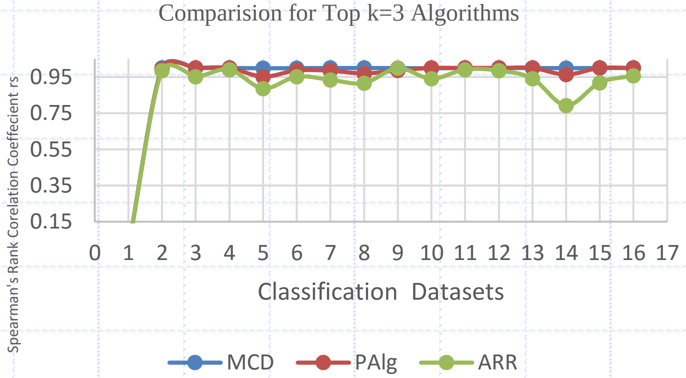
$$AMD(RC_i^{*}) = \frac{\overline{NISi}}{PISi^{*} + \overline{NISi}},$$

$where, i = 1,2,...,m$

Dataset/ Method	AMD			PAlg			ARR		
	Rs with α=0.1 (Wgt.F-Score=0.55, Rtime=0.45)			Rs with α=0.1 (Wgt.F-Score=0.55, Rtime=0.45)			Rs with α=0.1 (Wgt.F-Score=0.55, Rtime=0.45)		
Dataset	k=35	k=5	k=3	k=35	k=5	k=3	k=35	k=5	k=3
d1	0.9720	0.9978	1.0000	0.8473	0.9926	0.9944	0.6012	0.9769	0.9842
d2	1.0000	1.0000	1.0000	0.9900	1.0000	1.0000	0.5200	0.9450	0.9520
d3	1.0000	1.0000	1.0000	0.9641	1.0000	1.0000	0.5199	0.9940	0.9908
d4	0.9852	0.9974	0.9989	0.3187	0.9171	0.9521	0.2696	0.8752	0.8865
d5	0.9899	0.9992	0.9993	0.8081	0.9699	0.9863	0.4966	0.8975	0.9515
d6	0.9922	1.0000	1.0000	0.8314	0.9715	0.9851	0.3482	0.8641	0.9342
d7	0.9824	0.9997	1.0000	0.7028	0.9556	0.9697	0.2529	0.8871	0.9158
d8	0.9882	0.9986	0.9997	0.7541	0.9724	0.9869	0.5646	0.9956	0.9987
d9	0.9801	0.9985	0.9987	0.9908	1.0000	1.0000	0.5039	0.8929	0.9399
d10	0.9916	1.0000	1.0000	0.9748	0.9987	1.0000	0.5162	0.9799	0.9910
d11	0.9955	1.0000	1.0000	0.9501	1.0000	1.0000	0.5292	0.9636	0.9854
d12	0.9711	1.0000	1.0000	0.9706	1.0000	1.0000	0.4764	0.9359	0.9410
d13	0.9980	0.9992	0.9993	0.5070	0.9164	0.9637	0.2524	0.7271	0.7921
d14	0.9975	1.0000	1.0000	0.9756	0.9997	1.0000	0.4574	0.8694	0.9185
d15	0.9854	1.0000	1.0000	0.9728	0.9977	1.0000	0.5298	0.9107	0.9567
Average	0.9886	0.9993	0.9997	0.8372	0.9794	0.9892	0.4559	0.9143	0.9426







[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.

경희대학교
KYUNG HEE UNIVERSITY

Uniqueness and Contributions

Accurate classification and recommendation models development for real-world applications

- Semantics-preserved accurate Rough set classification model (**avg. accuracy 95.91%,)** and a precise hybrid-CBR model **(accuracy 94.0%)** are developed that *utilize semantics-enabled discretization* and accurate *case matching and retrieval similarity functions*
- Guidelines-enabled rule-based methods for correct data and datasets/case base creation

Accurately evaluating classifiers performance for optimum performance classifier selection

- A standard *expert group-based* method for selecting quality metrics and evaluation criteria
- A group-based decision making method for relative criteria weighting
- Significance and fitness functions with constraints satisfaction methods for accurate ranking and selection of consistent performance algorithm **(Spearman's ranked correlation coefficient (Rs) 0.979**)

Conclusion and Future Work

- This thesis proposed

Accurate rough set and CBR models creation and dataset/case base preparation

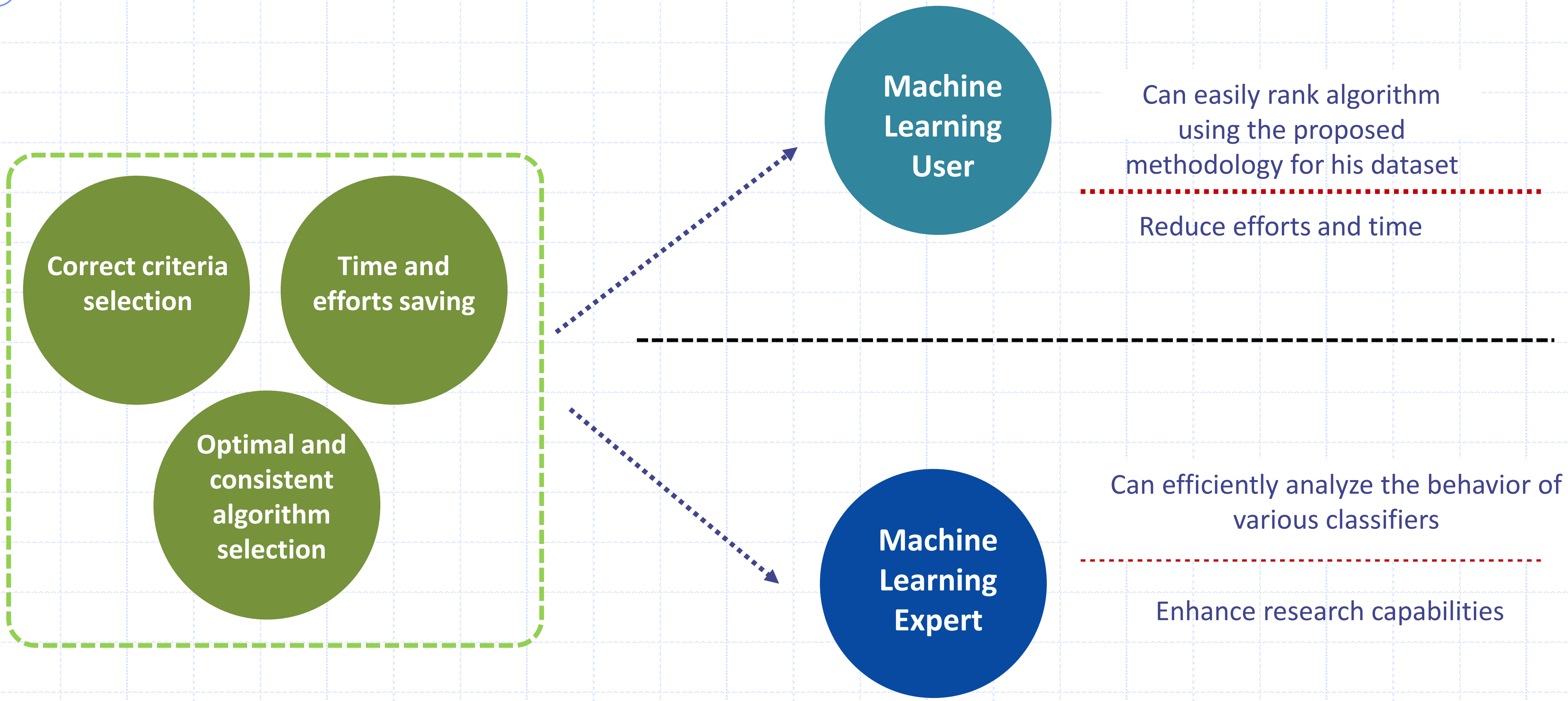
- A semantic-enabled discretization for accuracy preference
- Accurate case retrieval & reasoning using similarity functions
- Guidelines-enable dataset/case base creation for rough set and CBR models

Accurate recommendation/selection of classification algorithm using multiple criteria

- A standard experts' group-based criteria selection
- A consistent group-based relative criteria weighting
- Recommendation of consistent performance algorithm using consistency criteria and implicit and explicit constraints

- Future Research
  - Automatic **algorithm selectio**n using **meta-learning**
  - Finding optimum and most important **meta-features** for automatic algorithm selection
  - **Extending** the current multi-criteria decision making method to **other criteria and families of algorithms**

Benefits


Correct criteria selection
Time and efforts saving
Optimal and consistent algorithm selection
Machine Learning User
Can easily rank algorithm using the proposed methodology for his dataset
Reduce efforts and time
Machine Learning Expert
Can efficiently analyze the behavior of various classifiers
Enhance research capabilities

경희대학교
KYUNG HEE UNIVERSITY

Publications

Published papers

- Patents (02)
  - Two Korean
- SCI/ SCIE Journals (09)
  - SCI (02)
  - SCIE (02)
  - Co-author (05)
- Conferences (09)
  - International (02)
  - Domestic (03)
  - Co-author (4)

Total Publications (20)

First Author Publication s 11

Papers in progress

- SCI/ SCIE Journals (02)
  - Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21.
  - Ali. R et al A knowledge-based decision support system for inducing healthy lifestyle, Expert System, Under review, 2016

경희대학교 KYUNG HEE UNIVERSITY

Selected References

[1] Wang G, Song Q, Zhang X, Zhang K. A Generic Multilabel Learning-Based Classification Algorithm Recommendation Method. ACM Transactions on Knowledge Discovery from Data (TKDD). 2014 Oct 28;9(1):7.

[2] Mannor S, Meir R, Zhang T. Greedy algorithms for classification—consistency, convergence rates, and adaptivity. The Journal of Machine Learning Research. 2003 Dec 1;4:713-41.

[3] Liu Hongjiu and Hu Yanrong, "An Evaluating Method with Combined Assigning-Weight Based on Maximizing Variance," Scientific Programming, vol. 2015, Article ID 290379, 8 pages, 2015. doi:10.1155/2015/290379

[4] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. ACM SIGKDD explorations newsletter. 2009 Nov 16;11(1):10-8.

[5] Luo G. A review of automatic selection methods for machine learning algorithms and hyper-parameter values.

[6] Brazdil PB, Soares C, Da Costa JP. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning. 2003 Mar 1;50(3):251-77.

[7] Song Q, Wang G, Wang C. Automatic recommendation of classification algorithms based on data set characteristics. Pattern recognition. 2012 Jul 31;45(7):2672-89.

[8] Reif M, Shafait F, Goldstein M, Breuel T, Dengel A. Automatic classifier selection for non-experts. Pattern Analysis and Applications. 2014 Feb 1;17(1):83-96.

Wang G, Song Q, Zhang X, Zhang K. A generic multilabel learning-based classification algorithm recommendation method. ACM Transactions on Knowledge Discovery from Data (TKDD). 2014 Oct 9;9(1):7.

[9] Khanmohammadi S, Rezaeiahari M. AHP based Classification Algorithm Selection for Clinical Decision Support System Development. Procedia Computer Science. 2014 Dec 31;36:328-34.

[10] Smith KA, Woo F, Ciesielski V, Ibrahim R. Modelling the relationship between problem characteristics and data mining algorithm performance using neural networks, in: C. Dagli, et al. InEds.), Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining, and Complex Systems 2001.

[11] Ali R, Hussain J, Siddiqi MH, Hussain M, Lee S. H2RM: A hybrid rough set reasoning model for prediction and management of diabetes mellitus. Sensors. 2015 Jul 3;15(7):15921-51.

[11a] Zierler-Brown, S.; Brown, T.R.; Chen, D.; Blackburn, R.W. Clinical documentation for patient care: Models, concepts, and liability considerations for pharmacists. Am. J. Health-Syst. Pharm. 2007, 64, 1851–1858.

[12] Ali R, Siddiqi MH, Lee S. Rough set-based approaches for discretization: A compact review. Artificial Intelligence Review. 2015 Aug 1;44(2):235-63.

[13] Prędki B, Wilk S. Rough set based data exploration using ROSE system. InFoundations of Intelligent Systems 1999 Jun 8 (pp. 172-180). Springer Berlin Heidelberg.

[14] Ali R, Afzal M, Hussain M, Ali M, Siddiqi MH, Lee S, Kang BH. Multimodal hybrid reasoning methodology for personalized wellbeing services. Computers in biology and medicine. 2016 Feb 1;69:10-28.

[15] Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European journal of operational research, 48*, 9-26.

[16] Witten, I. H., & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann.

[17] García-Cascales MS, Lamata MT. On rank reversal and TOPSIS method. Mathematical and Computer Modelling. 2012 Sep 30;56(5):123-32.

[18] Van Rijn JN, Bischl B, Torgo L, Gao B, Umaashankar V, Fischer S, Winter P, Wiswedel B, Berthold MR, Vanschoren J. OpenML: A collaborative science platform. InMachine learning and knowledge discovery in databases 2013 Sep 23 (pp. 645-649). Springer Berlin Heidelberg.

[21] Bouckaert, R. R., Frank, E., Hall, M. A., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2010). WEKA---Experiences with a Java Open-Source Project. *The Journal of Machine Learning Research, 11*, 2533-2541.

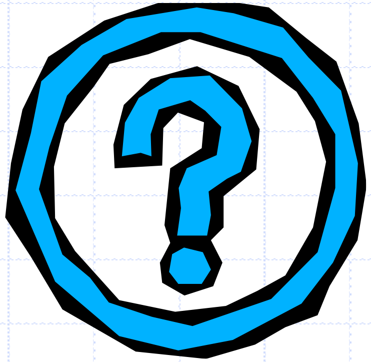
[22] Perzina R, Ramik J. DAME-Microsoft Excel add-in for solving multicriteria decision problems with scenarios. InProceedings of the 30th International Conference Mathematical Methods in Economics 2012 (pp. 697-702).

[23] SANNA 2014 - MS Excel based system for multicriteria evaluation of alternatives.

[24] Ali R, et. al.. "An accurate multi-criteria decision making methodology for recommending optimum performance machine learning algorithm(s)". Entropy. Reviews Completed, 2016 April 21..

THANK YOU!

Any questions or comments?



Appendix

- Solution 1(A-1)

Solution 1(A-1): Rough Set-based Prediction Model for Classification

2 Data Mining (rules extraction) using Rough Set Theory

2.1 Preprocessing Clinical Data

- **Features filtration**: irrelevant features
- **Filling missing values:** dataset level (missing >=20%), and patient encounter level (missing <=2 encounters; <20% encounters; >=20% of the encounters)

2.2 Data Reduction

- **Discretization**:
  - **Limitation: Existing methods [12]** (statistical, entropy, genetic, fuzzy theory and Boolean) lakes semantics splitting
  - **Solution**: semantic –enabled discretization

Attributes	# Cut-points: Cut-points Description	# Intervals: Interval Description	Discrete Value for Interval
BMI	3: 18.5;25;30	4: (-∞, 18.5),[18.5, 24.9],[25, 30),[30, ∞)	0,1,2,3
Gender	NA	NA	NA
Age	2: 30;50	3: (-∞, 30),[30, 50],(50, ∞)	0,1,2
SBP	4: 120;140;160;181	5: (-∞, 120),[120, 139],[140, 159],[160, 180],[181, ∞)	0,1,2,3,4

Example: SBP Values
Existing (Boolean Method) -- >
- (SBP < 110), (SBP 110-116), (SBP ≥117)

Proposed Method -- >
- (-∞, 120),[120, 139],[140, 159],[160, 180],[181, ∞)

- **Features Selection:** Reduct and Core methods for final attributes.

Reduct #	# Attributes	Reduct (Attributes)
1	10	{BMI, Gender, Age, SBP, DBP, FBS, Hba1c, HDL, LDL, PT}
2	10	{BMI, Age, SBP, DBP, FBS, Hba1c, TG, HDL, LDL, PT}
3	10	{BMI, Gender, Age, SBP, FBS, Hba1c, HDL, LDL, OT, PT}
4	10	{BMI, Age, SBP, FBS, Hba1c, TG, HDL, LDL, OT, PT}

Eight (08) Selected Features

Core(DIS)=Intersection (RED(DIS))=
{BMI, Age, SBP, FBS, Hba1c, HDL, LDL, PT}

2.3 Rules Mining & Validation

- **Rules Mining**: basic minimal covering criteria of LEM2 algorithm is used

Rule #	Prediction for TDM	Prediction Rule	Significance
1	(T1DM)	(BMI = [18.5, 24.9]) & (Age = (50, ∞)) & (SBP = [120, 139]) & (Hba1c = (7.4, ∞)) & (TC = (-∞, 200)) & (SGPT = [7, 56])	20 (17.70%)

[12] Ali R, Siddiqi MH, Lee S. Rough set-based approaches for discretization: A compact review. Artificial Intelligence Review. 2015 Aug 1;44(2):235-63.

Solution 1(A-1): Rough Set-based Classification Model

3 Classification/Prediction for Decision

3.1 Rough Set-based Classification

- Rough Set-based Classification Algorithm for accurately predicting new cases

3.1 Reference Range-based Classification

- Use Reference range rules, extracted from guidelines, to find risky prediction

4 Correlation-based Trend Analysis for Prognosis

- Supporting prediction decisions with insights of the past observations of patients
- Future trend predicted using correlation-based polynomial trend line of order 3 is computed using the equation.

$$y = -\propto x^3 + \beta x^2 - \gamma x + c$$
$$R^2 = \text{residue}$$

- Residue R2 is computed for accuracy of the future prediction

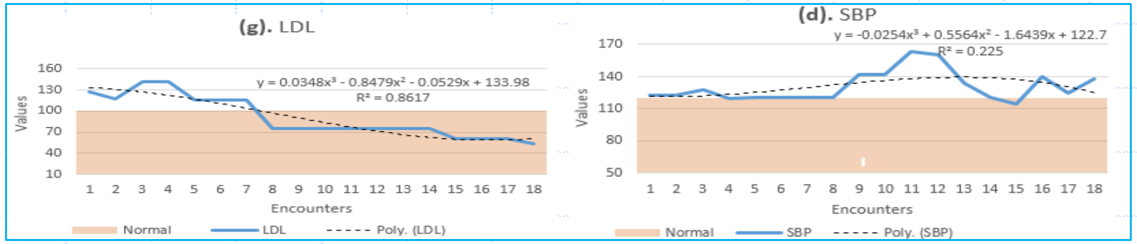
Algorithm. Rough Set-based Classification & Trend Analysis

Input: KB: Knowledge Base, I: New Instance
Output: classType, INTERPRETATION

Begin
ApplyRBR (**I**), where {I|I is New Instance, I: = {Iid, Cond}}
A. PerformRSR(**I**) // **Rough Set Reasoning**
[Load Classification Rules From Knowledge Base]
1. DMPR: = LoadRulesFromKB(RULES that contain classType as CONC); where CONC: = { classType 1, . . , classType n [Execute Rules For Classification of CONC]
2. **Foreach** RULE in DMPR
   a. **Foreach** CA in RULE //CA: = {Cond} are conditions of the rule
   b. **If** CA. values ≠ E. Cond. value
      THEN Try next RULE
      EndIf
   c. classType ≔ CONC of the RULE;
   d. **Goto** Step **B**
   e. **EndFor**
   EndFor
3. classsType = Message(UNDEFINED);

B. Perform3R (**I**) // **Reference Range-based Reasoning**
[Load Reference Range Rules From Knowledge Base]
4. ATAR: = LoadRulesFromKB(RULES that contain INTERPRETATION as CONC); where CONC: = { Table 2. INTERPRETATION. Value}
   [Execute Rules For Finding Current Status of Each Observations]
5. **Foreach** RULE in ATAR
   a. **Foreach** CA in RULE
   b. **If** CA. values ≠ I. OBS. value
      THEN Try next RULE
      EndIf
   c. INTERPRETATION [] ≔ CONC of the RULE;
      EndFor
   EndFor

C. Expert ≔ **ProvideResults** (**Iid, classType, INTERPRETATION**)
End



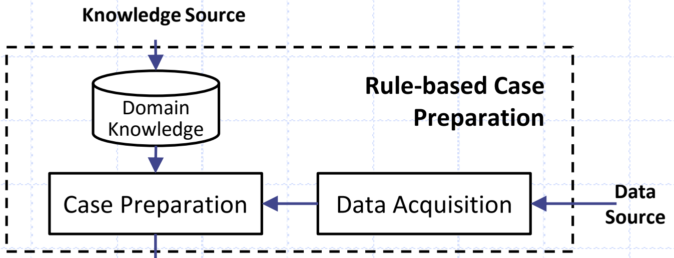
Appendix

- Solution 1(A-2)

Solution 1(A-2): CBR Recommendation Model

1-2 **Rule-based Case Preparation**

- Acquire/receive data
- Acquire/extract domain knowledge
- Transform data to case using RBR methodology
- Define case structure, conditions, data types and conclusion
- Persist case into case base



Condition	Data type	Possible value	Description
C1	Symbol	{v1,v2,..}	…
…	…	…	…
Cn	Float	Min, Max	…
Conclusion	String	{r1,r2}	…

3 **Accurate case retrieval & reasoning methodology using similarity functions**

3.1 **Local Similarity Function Definition**

[continuous value] Closest match similarity function

$$\mathbf{METSim_l(nC, eC)} = \frac{\mathbf{d_g(Max_{MET}, Min_{MET}) - d_l(nC_{MET}, eC_{MET}) - 1}}{\mathbf{d_g(Max_{MET}, Min_{MET})}}$$

[nominal value] Exact match similarity function

$$\mathbf{AGSim_l(nC, eC)} = \begin{Bmatrix} \mathbf{AG_{ij} = 1} & \mathbf{for\ \forall\ (i \geq j) OR (i = 0\ OR\ j = 1)} \\ \mathbf{0} & \mathbf{otherwise} \end{Bmatrix}$$

3.2 **Global Similarity Function Definition**

$$\mathbf{Sim_g(nC, eC) = \beta(AGSim_l(nC, eC)) + \gamma(METSim_l(nC, eC))}$$

where $\beta = 0.1$ and $\gamma = 0.9$ are the weight values of age and MET attributes,

4 **Select Top-k relevant recommendations**

$$\mathrm{R:= ApplyKNN}(\mathbf{Sim_g}]);\ //\mathrm{where\ k = 3}$$

Solution 1(A-2): CBR Recommendation/ Classification Model

3 Design and Implementation of Case-based Reasoning

3.1 Local Similarity Function Definition

$$\mathbf{METSim_l(nC, eC)} = \frac{\mathbf{d_g(Max_{MET}, Min_{MET}) - d_l(nC_{MET}, eC_{MET}) - 1}}{\mathbf{d_g(Max_{MET}, Min_{MET})}} \quad (11)$$

$$\mathbf{AGSim_l(nC, eC)} = \begin{Bmatrix} \mathbf{AG_{ij} = 1} & \mathbf{for\ \forall\ (i \geq j) OR (i = 0\ OR\ j = 1)} \\ \mathbf{0} & \mathbf{otherwise} \end{Bmatrix} \quad (12)$$

Age Group	All Age	Young	Older Adults	Adults
All Age	1	1	1	1
Young	1	1	0	0
Older Adults	1	1	1	0
Adults	1	1	1	1

3.2 Global Similarity Function Definition

$$\mathbf{Sim_g(nC, eC) = \beta(AGSim_l(nC, eC)) + \gamma(METSim_l(nC, eC))} \quad (13)$$

where $\beta = 0.1$ and $\gamma = 0.9$ are the weight values of age and MET attributes,

3.3 Retrieve and Retain Existing/New Cases

Algorithm 4. Case-based reasoning methodology for generating accurate recommendations decisions

Input: UID:uid, METCBurl, nC:= new Case

Output: List PAR <Recommendations>

Begin

Let PAR:= A set of top 3 relevant existing cases as the proposed recommendations

Sim_g[]:= Array of global similarities of existing cases

$METCB_r :=$ ReteriveCaseBaseFromKB(METCBurl), Where $METCB_r$ is the matrix $eC_m x A_n$, eC_m is the set of existing cases, i.e., $eC = eC_1, eC_2, eC_3, \ldots, eC_m$. Similarly, A_n is the set of attributes, i.e., $A_n = A_1, A_2, A_3, \ldots, A_n$

1. **For** i = 1 to SizeOfCases($METCB_r$)

   Let Sim_l[]:=Array of local similarities of attributes of individual cases

   a. **For** j = 1 to SizeOfAttributes($METCB_r$)

   b. $Sim_l[A_j] := \text{ComputeLocSim}(nC.A_j, METCB_r[i, j])$; // use eq. 11 and eq. 12

   c. **End for**

   d. $\mathbf{Sim_g}[eC_i] := \text{ComputeGlobSim}\ (\mathbf{Sim_l})$; // weighted sum method (eq.13)

2 **End for**

3. PAR: = ApplyKNN($\mathbf{Sim_g}$]); //where k = 3

4. PropgateCBRResults (uid, PAR);

5. FCB ≔ RetainCBRPAR(uid, PAR);

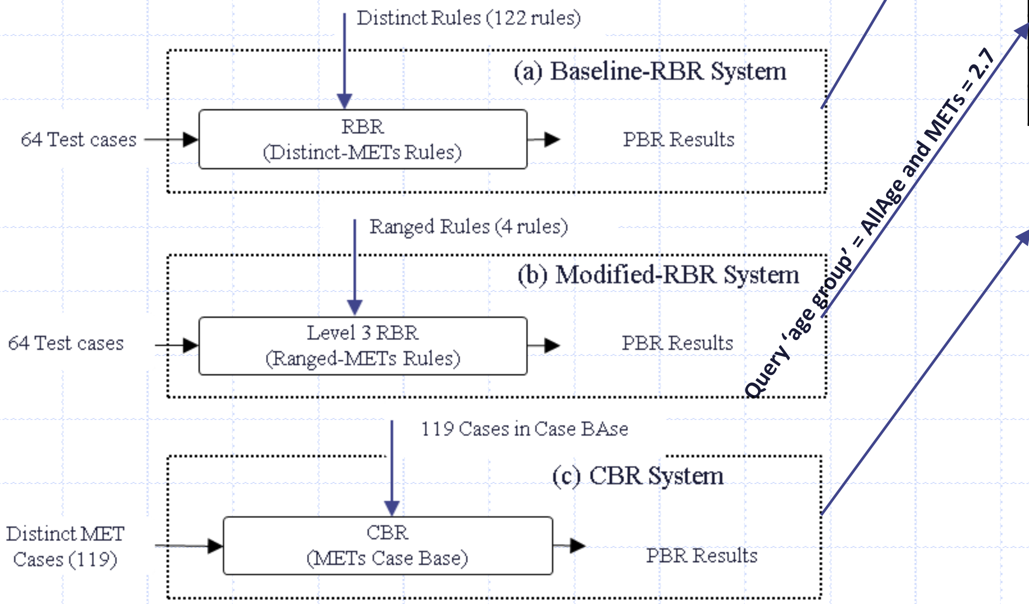
6. **Exit;**

End

Ranking Recommendations

Solution 1(A-2): Evaluation and Comparison

- **Comparison**
  - 64 input cases randomly generated from original cases
- Experiments and Analysis of Results
  - Experiment 1 (Baseline Experiment):
    - RBR with distinct MET
  - Experiment 2 (Modified-RBR Experiment):
    - RBR with ranged-MET
  - Experiment 3 (CBR Experiment):
    - CBR with Test Case Base



UID	Age	METs	Recommendations
1	Young	6.5	i. Climbing hills with 0 to 9 lb load. ii. Race walking; rock or mountain climbing
2	Adult	7.6	X
3	Older Adult	7.8	i. backpacking; hiking or organized walking with a daypack
4	Adult	8.1	X
…	…	…	…
n	Older Adult	7.6	X

Recommendation #	METs	Suggested physical activity recommendations
1	1.3	riding in a car or truck
2	1.3	riding in a bus or train
3	1.5	sitting; meeting; general; and/or with talking involved
…	…	…
17	2.8	walking; 2.0 mph; level; slow pace; firm surface

UID	METs value)	Retrieved cases (METs value)	Recommendations decision
1	6.5	6.5	climbing hills with 0 to 9 lb load.
		6.5	race walking; rock or mountain climbing
		6.3	climbing hills; no load
2	7.6	7.3	climbing hills with 10 to 20 lb load
		7.5	bicycling; general
		7.8	backpacking; hiking or organized walking with a daypack
3	7.8	7.8	backpacking; hiking or organized walking with a daypack
		8	running; training; pushing a wheelchair or baby carrier
		8	running; marathon
4	8.1	8	running; training; pushing a wheelchair or baby carrier
		8	running; marathon

Solution 1(A-2): CBR Recommendation Model

Characteristics of Case Base

S.No	Type of activity	Distribution
1	Running	25
2	Walking	56
3	Cycling	18
4	Standing	5
5	Sitting	4
6	Transportation	4
7	Volunteer	7
Total instances		119

Appendix

- Solution 2 (B)

Guidelines for criteria selection and algorithms performance analysis

1. Define an unambiguous goal for which the algorithm(s) need to be selected
2. Analyze and specify goal as either single-objective or multi-objectives and specify the corresponding quality meta-metrics (QMM)
   a. Categorize objective(s) as cost and benefit criteria
   b. Define essential constraints on the objective(s), reflecting goal's constraints
3. Analyze the specified objective(s) and constraints against existing criteria
   a. If existing criteria work, then go to step 4.
   b. If existing criteria do not fit well, then go to step 5.
4. Evaluate the algorithms performances using the available criterion under the constraints, defined in step 2(b), and rank them for the best selection
5. Define a generic multi-metrics evaluation criteria using the following steps
   a. Analyze QMM for conflict among evaluation criteria (interdependence/fuzziness)
   b. Select suitable QMM, defining the objectives.
   c. Select suitable evaluation metrics for the selected QMM (objectives)
   d. Prioritize the selected evaluation metrics
   e. Rank algorithms based on the aggregate value of the weighted metrics
   f. Repeat step 5, if any of the constraints, defined in step 2(b), is not satisfied

Selecting Quality Meta-metrics

Table 6. Experts' group-based rating of quality metrics for heterogeneous classifiers

Quality Metrics	DM#1	DM#2	DM#3	DM#4	Total
Correctness (cor)	60	50	55	70	**235**
Computational Complexity (ccom)	5	20	15		**40**
Responsiveness (res)	15		20	20	**55**
Consistency (con)	10	15			**25**
Comprehensibility (com)		15		7	**23**
Reliability (rel)	5				**5**
Robustness (rob)			10	3	**13**
Separability (sep)	5				**5**
Total	**100**	**100**	**100**	**100**	**400**

Consistency measure

$$\mathrm{Consistency}_{a \in A} = \frac{\sum_{i=1}^{m} \mathrm{Stdev}_i}{m}$$

Table 6.9. A partial list of the average standard deviation of classifiers

Algorithms	**Wgt.Avg.F-score (Stdev)**	**CPUTimeTraining (Stdev)**	**CPUTimeTesting (Stdev)**	**Average (Stdev) - Consistency**
bayes.BayesNet	0.018	0.015	0.005	**0.013**
bayes.NaiveBayes	0.017	0.006	0.008	**0.010**
bayes.NaiveBayesUpdateable	0.017	0.007	0.008	**0.011**
functions.Logistic	0.015	0.019	0.002	**0.012**
…	…	…	…	**…**
meta.Vote	0.017	0.010	0.000	**0.009**

Saaty's preference scale

Table 6.2. Saaty's preference scale for pair-wise comparison of the performance metrics

Definition	Intensity of importance	Definition	Intensity of importance
Equally important	1	Equally important	1/1
Equally or slightly more important	2	Equally or slightly less important	1/2
Slightly more important	3	Slightly less important	1/3
Slightly to much more important	4	Slightly to way less important	1/4
Much more important	5	Way less important	1/5
Much to far more important	6	Way to far less important	1/6
Far more important	7	Far less important	1/7
Far more important to extremely more important	8	Far less important to extremely less important	1/8
Extremely more important	9	Extremely less important	1/9

Relative consistent weight algorithm

Procedure 3. estimatRelativeWeight

Begin

inputs: $\overline{\overline{Q}} = \{e_1, e_2, \dots, e_m\}$ // the list of m selected evaluation metric (e)

output: W – weight vector of the set of evaluation metrics $\overline{\overline{Q}}$

Let $DM = \{dm_1, dm_2, \dots, dm_n\}$ // Group of Experts

SPS = Saaty's preference scale (see Table 6.2)

GDMM = m*n grouped decision making matrix of the weight of evaluation metrics by assigned by decision makers

1. [Design comparison matrix for decision makers]

   $DMM = dm_{ij}$; //where, DMM is an n*n comparison matrix of the decision makers with dm_{ij} is the preference value of the i^{th} decision maker relative to the j^{th} decision maker

1. [Estimate decision makers decisions weight]

   a. $DMWeight = estimateDMWgt(SPS, DMM)$; //where DMDWeight is a column weights vector of the decision makers' weights. // See equations 2,3

   b. Check consistency of DMWeight; // See equations 4-7

2. [Estimate evaluation metrics weights]

   For $dm = 1$ to n **do**

   a. $EM = e_{ij}$; //where, EM is an m*m comparison matrix of the evaluation metrics entered by the decision maker. Each value e_{ij} is the preference value of the i^{th} evaluation metric relative to the j^{th} evaluation metric

   b. $EMWeight = estimateEvalMetricsWgt(SPS, EM)$; //where, EMWeight is a column weights vector for evaluation metrics $\overline{\overline{Q}}$ estimated by decision maker dm. // See equations 2,3

   c. Check consistency of EMWeight; // See equations 4-7

   d. Insert $< EMWeight >$ into GDMM;

   End for

1. [Grouped decision making]

   $groupedWeight = \sum_{e=1}^{m}(\prod_{dm=1}^{n} DMWeight * EMWeight)$; //where, groupedWeight is a column weights vector for the weights of all the evaluation metrics $\overline{\overline{Q}}$ estimated by estimated by DM

1. $W = groupedWeightDM$;
2. return W;

End

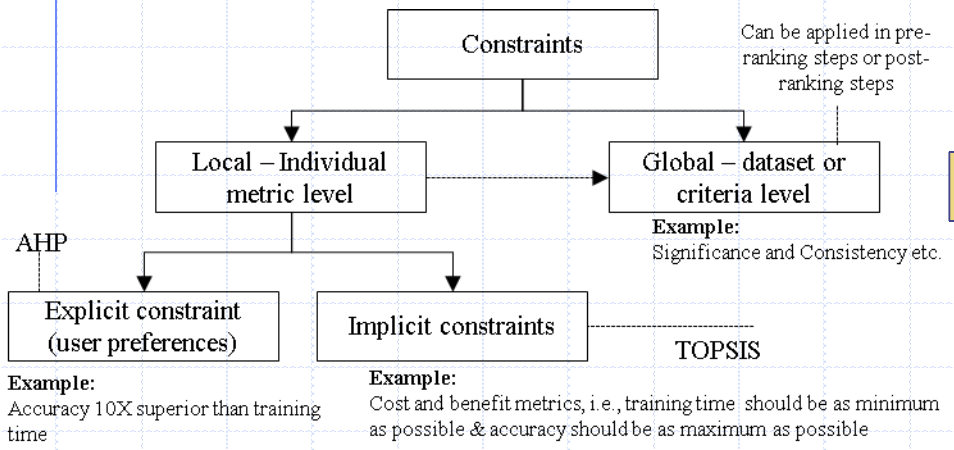
Solution 2: Experiments and Results

- **Experiment 3** (Significance fitness evaluation function)
- Purpose of the experiment
  - To find out algorithms that are non-significant on all criteria
- Interpretation
  - Three probabilistic, two decision tree and two meta-leaning algorithms performed poorly on all three criteria
  - If they are not get excluded prior to ranking, the results are affected

Algorithm	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8	Dataset 13
bayes.BayesNet	26	4	2	7	27	**4**
bayes.NaiveBayes	**19**	11	12	21	**30**	**7**
bayes.NaiveBayesUpdateable	**20**	10	15	20	**31**	8
trees.FT	30	**32**	**32**	**32**	25	2
trees.RandomForest	17	**25**	23	**24**	**17**	6
meta.Dagging	**27**	18	21	26	**32**	**30**

- bays.BayesNet is ranked 4th and bayes.NaiveBayes as 7th.
- Same is for other

Solution 2: Ranking & Constraints



Constraints

Table 11. Performance results of classification algorithms on ADA_Agnostic dataset and their ranking with respect to relative distance from the ideal algorithm

	Constraints							
	Max	Min	Min	Min				
Algorithms	F-score	TimeTraining	TimeTesting	Consistency	PIS+	NIS-	RC	Ranking
bayes.BayesNet	0.78*	0.027*	0.002	0.013	0.00906	0.03830	**0.80874**	26
bayes.NaiveBayes*	0.825*	0.013*	0.008*	0.010	0.00264	0.04180	**0.94068**	19
bayes.NaiveBayesUpdateable*	0.825*	0.011*	0.01*	0.011	0.00272	0.04171	**0.93882**	20
functions.Logistic	0.836	0.229*	0.000	0.012	0.00088	0.04317	**0.97995**	4
functions.RBFNetwork	0.733*	0.232*	0.004	0.043	0.01593	0.03492	**0.68672**	29
functions.SMO	0.830	1.99*	(ref) **0.000**	0.041	0.00181	0.04239	**0.95905**	12
misc.HyperPipes	0.66*	(ref) **0.001**	0.000	0.005	0.02658	0.03309	**0.55457**	32
misc.VFI	0.716*	0.008*	0.004	0.012	0.01841	0.03433	**0.65097**	31
rules.ConjunctiveRule	0.645*	0.043*	0.000	0.006	0.02877	0.03301	**0.53432**	35
rules.DecisionTable	0.829	1.086*	0.000	0.043	0.00195	0.04231	**0.95597**	14
rules.DTNB	0.832	88.16*	0.004	2.611	0.02792	0.03234	**0.53668**	33
rules.JRip	0.825*	0.648*	0.000	0.067	0.00257	0.04180	**0.94203**	18
rules.OneR	0.739*	0.014*	0.000	0.007	0.01504	0.03574	**0.70380**	28
rules.PART	0.819*	1.161*	0.001	0.057	0.00341	0.04126	**0.92367**	23
rules.Ridor	0.795*	0.453*	0.000	0.034	0.00687	0.03942	**0.85156**	24
rules.ZeroR	0.645*	0.000	0.000	0.001	0.02877	0.03305	**0.53463**	34
trees.BFTree	0.838	0.79*	0.000	0.024	0.00063	0.04328	**0.98557**	**2**
trees.FT	0.827	1.38*	0.161*	0.044	0.01790	0.03819	**0.68088**	30
trees.J48	0.828	0.221*	0.000	0.014	0.00205	0.04241	**0.95392**	15
trees.J48graft	0.829	0.29*	0.000	0.014	0.00190	0.04251	**0.95715**	13
trees.LADTree	0.833	1.676*	0.000	0.020	0.00134	0.04281	**0.96967**	10
trees.RandomForest	0.837	2.304*	0.022*	0.022	0.00255	0.04223	**0.94299**	17
trees.RandomTree	0.791*	0.028*	0.001	0.009	0.00745	0.03923	**0.84041**	25
trees.REPTree	0.835	0.084*	0.000	0.012	0.00103	0.04308	**0.97669**	7
trees.SimpleCart	0.836	0.713*	0.000	0.021	0.00090	0.04311	**0.97950**	5
meta.AdaBoostM1	0.822*	1.074*	0.001	0.021	0.00293	0.04176	**0.93440**	21
meta.Bagging	(ref) **0.842**	0.753*	0.000	0.013	0.00014	0.04373	**0.99681**	**1**
meta.Dagging*	0.824*	0.013*	0.107*	0.010	0.01209	0.03861	**0.76154**	27
meta.END	0.828	0.215*	0.003	0.013	0.00207	0.04228	**0.95323**	16
meta.FilteredClassifier	0.832	0.065*	0.000	0.009	0.00146	0.04282	**0.96697**	11
meta.LogitBoost	0.835	1.948*	0.002	0.058	0.00121	0.04267	**0.97245**	9
meta.RacedIncr.LogitBoost	0.82*	0.062*	0.001	0.012	0.00322	0.04166	**0.92833**	22
meta.RandomSubSpace	0.837	0.412*	0.001	0.012	0.00075	0.04322	**0.98299**	**3**
meta.Stacking	0.834	0.724*	0.001	0.014	0.00118	0.04292	**0.97318**	8
meta.Vote	0.835	0.076*	0.000	0.009	0.00103	0.04310	**0.97676**	6
Relative Weights	**0.69520**	**0.05067**	**0.10097**	**0.15315**				
Positive Ideal Solution (PIS)	**0.12296**	**0.00874**	**0.01776**	**0.02647**				
Negative Ideal Solution (NIS)	**0.09419**	**0.00000**	**0.00000**	**0.00000**				

$$\text{RankedList} = \text{RANK.AVG}(\text{RC}^{*}{}_{1}, \text{RC}^{*}{}_{1} : \text{RC}^{*}{}_{n}, 1)$$

Solution 2: Comparison with State-of-the-Art Methods

- Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results [6]

$$\mathrm{ARR} = \frac{\frac{\mathrm{SR}_{ap}^{di}}{\mathrm{SR}_{aq}^{di}}}{1 + \propto * \log\left(\frac{\mathrm{T}_{ap}^{di}}{\mathrm{T}_{aq}^{di}}\right)}$$

- Automatic recommendation of classification algorithms based on data set characteristics [7]

$$\mathrm{P}_{\mathrm{Alg,D}} = \frac{\mathrm{Accuracy}_{\mathrm{Alg,D}}}{1 + \propto * \log\left(\mathrm{RTime}_{\mathrm{Alg,D}}\right)}$$

- Where, $\propto$ = 0.1, 1, 10 for 10% for specifying 10% preference of the accuracy, equal preferences for both accuracy and 10% preference of the total time (execution/training)

- Proposed Method Setting

$$\mathrm{RC\ (Relative\ Closness)} = \frac{\mathrm{NIS}_i^-}{\mathrm{NIS}_i^- + \mathrm{PIS}_i^+}$$

  - We averaged CPUTimeTraining and CPUTimeTesting to get uniform value for Total/Rtime
  - We dropped the fourth consistency criterion from out method
  - For simplicity, we performed experiments only for $\propto$ = 0.1 with three different sitting (k=35 algorithms, k=5 and k=3)
  - The weight for Accuracy and Total/Rtime were taken as 0.55 and 0.45

[1] Brazdil PB, Soares C, Da Costa JP. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. Machine Learning. 2003 Mar 1;50(3):251-77. (Cited by 284)
[2] Song Q, Wang G, Wang C. Automatic recommendation of classification algorithms based on data set characteristics. Pattern recognition. 2012 Jul 31;45(7):2672-89. (Cited by 25)

Freidman Test (Statistical Significance)

(a) Friedman's test steps for comparing ranking methods with k=35

Dataset	d1		d2		d3		d4		d5		d6		d7		d8		d9		d10		d11		d12		d13		d14		d15			
Method\Rs	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	RRj	(RRj-mR)^2
AMD	0.9720	1.0	1.0000	1.0	1.0000	1.0	0.9852	1.0	0.9899	1.0	0.9922	1.0	0.9824	1.0	0.9882	1.0	0.9801	2.0	0.9916	1.0	0.9955	1.0	0.9711	1.0	0.9980	1.0	0.9975	1.0	0.9854	1.0	1.1	0.871111111
PAlg	0.8473	2.0	0.9900	2.0	0.9641	2.0	0.3187	2.0	0.8081	2.0	0.8314	2.0	0.7028	2.0	0.7541	2.0	0.9908	1.0	0.9748	2.0	0.9501	2.0	0.9706	2.0	0.5070	2.0	0.9756	2.0	0.9728	2.0	1.9	0.004444444
ARR	0.6012	3.0	0.5200	3.0	0.5199	3.0	0.2696	3.0	0.4966	3.0	0.3482	3.0	0.2529	3.0	0.5646	3.0	0.5039	3.0	0.5162	3.0	0.5292	3.0	0.4764	3.0	0.2524	3.0	0.4574	3.0	0.5298	3.0	3.0	1
																															S	1.875555556

(b) Friedman's test steps for comparing ranking methods with k=5

Dataset	d1		d2		d3		d4		d5		d6		d7		d8		d9		d10		d11		d12		d13		d14		d15			
Method\Rs	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	RRj	(RRj-mR)^2
AMD	0.9978	1.0	1.0000	1.5	1.0000	1.5	0.9974	1.0	0.9992	1.0	1.0000	1.0	0.9997	1.0	0.9986	1.0	0.9985	2.0	1.0000	1.0	1.0000	1.5	1.0000	1.5	0.9992	1.0	1.0000	1.0	1.0000	1.0	1.2	0.64
PAlg	0.9926	2.0	1.0000	1.5	1.0000	1.5	0.9171	2.0	0.9699	2.0	0.9715	2.0	0.9556	2.0	0.9724	3.0	1.0000	1.0	0.9987	2.0	1.0000	1.5	1.0000	1.5	0.9164	2.0	0.9997	2.0	0.9977	2.0	1.9	0.017777778
ARR	0.9769	3.0	0.9450	3.0	0.9940	3.0	0.8752	3.0	0.8975	3.0	0.8641	3.0	0.8871	3.0	0.9956	2.0	0.8929	3.0	0.9799	3.0	0.9636	3.0	0.9359	3.0	0.7271	3.0	0.8694	3.0	0.9107	3.0	2.9	0.871111111
																															S	1.528888889

(c) Friedman's test steps for comparing ranking methods with k=3

Dataset	d1		d2		d3		d4		d5		d6		d7		d8		d9		d10		d11		d12		d13		d14		d15			
Method\Rs	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	Rs	RR	RRj	(RRj-mR)^2
AMD	1.0000	1.0	1.0000	1.5	1.0000	1.5	0.9989	1.0	0.9993	1.0	1.0000	1.0	1.0000	1.0	0.9997	1.0	0.9987	2.0	1.0000	1.5	1.0000	1.5	1.0000	1.5	0.9993	1.0	1.0000	1.5	1.0000	1.5	1.3	0.49
PAlg	0.9944	2.0	1.0000	1.5	1.0000	1.5	0.9521	2.0	0.9863	2.0	0.9851	2.0	0.9697	2.0	0.9869	3.0	1.0000	1.0	1.0000	1.5	1.0000	1.5	1.0000	1.5	0.9637	2.0	1.0000	1.5	1.0000	1.5	1.8	0.054444444
ARR	0.9842	3.0	0.9520	3.0	0.9908	3.0	0.8865	3.0	0.9515	3.0	0.9342	3.0	0.9158	3.0	0.9987	2.0	0.9399	3.0	0.9910	3.0	0.9854	3.0	0.9410	3.0	0.7921	3.0	0.9185	3.0	0.9567	3.0	2.9	0.871111111
																															S	1.415555556

Friedman's Test	S	M	C	M vs. C	Interpretation
Top-K=35	1.876	28.133	10.99	M > C	M(28.13) > C(10.99) → null hypothesis is rejected at the confidence level α = 0.001
Top-K=5	1.529	22.933	10.99	M > C	M(22.93) > C(10.99) → null hypothesis is rejected at the confidence level α = 0.001
Top-K=3	1.416	21.233	10.99	M > C	M(21.23) > C(10.99) → null hypothesis is rejected at the confidence level α = 0.001