Future Generation Computer Systems 78 (2018) 59–76

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Kuaa: A unified framework for design, deployment, execution, and recommendation of machine learning experiments

Rafael de Oliveira Werneck [a,*], Waldir Rodrigues de Almeida [a], Bernardo Vecchia Stein [a], Daniel Vatanabe Pazinato [a], Pedro Ribeiro Mendes Júnior [a], Otávio Augusto Bizetto Penatti [a,b], Anderson Rocha [a], Ricardo da Silva Torres [a]

[a] RECOD Lab., Institute of Computing (IC), University of Campinas (Unicamp), Av. Albert Einstein, 1251, Campinas, SP, 13083-852, Brazil
[b] Advanced Technologies Group, SAMSUNG Research Institute, Av. Cambacica, 1200, Building 1, Campinas, SP, 13097-160, Brazil

HIGHLIGHTS

- A framework for designing and deploying machine-learning experiments.
- Standardized environment for exploratory analysis of machine-learning solutions.
- The modeling of a machine-learning experiment as a workflow.
- A framework capable of recommending machine-learning workflows to the user.
- Evaluation of four similarity measures and a learning-to-rank method for recommending workflows.

ARTICLE INFO

Article history:
Received 7 February 2017
Received in revised form 20 April 2017
Accepted 13 June 2017
Available online 18 July 2017

Keywords:
Machine learning
Science - experiments
Workflow
Recommendation systems (Information filtering)

ABSTRACT

In this work, we propose Kuaa, a workflow-based framework that can be used for designing, deploying, and executing machine learning experiments in an automated fashion. This framework is able to provide a standardized environment for exploratory analysis of machine learning solutions, as it supports the evaluation of feature descriptors, normalizers, classifiers, and fusion approaches in a wide range of tasks involving machine learning. Kuaa also is capable of providing users with the recommendation of machine-learning workflows. The use of recommendations allows users to identify, evaluate, and possibly reuse previously defined successful solutions. We propose the use of similarity measures (e.g., Jaccard, Sørensen, and Jaro–Winkler) and learning-to-rank methods (LRAR) in the implementation of the recommendation service. Experimental results show that Jaro–Winkler yields the highest effectiveness performance with comparable results to those observed for LRAR, presenting the best alternative machine learning experiments to the user. In both cases, the recommendations performed are very promising and the developed framework might help users in different daily exploratory machine learning tasks.

1. Introduction

With data deluge becoming ever more commonplace and pervasive due to spectacular advances in hardware and software acquisition technologies, it becomes imperative to properly process such data for extracting information that can lead to knowledge generation. This knowledge extraction process is usually performed by means of data mining and machine-learning methods, with the ultimate goal being the improvement of the decision-making process in a target application [1].

A typical machine-learning solution comprises several steps, including, for example, feature extraction and normalization methods, and the definition of appropriate classifiers. Since there is no *silver bullet* that solves all machine learning problems, each technique has its own pros and cons when designed for specific applications. In this sense, one common strategy adopted for developers of machine learning systems consists in performing exploratory analysis often relying on running several experiments with the objective of identifying which techniques are more appropriate for a given application.

* Corresponding author.
E-mail addresses: rafael.werneck@ic.unicamp.br (R.de O. Werneck), waldir.r.almeida@gmail.com (W.R. de Almeida), bernardovstein@gmail.com (B.V. Stein), danielvatanabe92@gmail.com (D.V. Pazinato), pedrormjunior@gmail.com (P.R.M. Júnior), o.penatti@samsung.com (O.A.B. Penatti), anderson.rocha@ic.unicamp.br (A. Rocha), rtorres@ic.unicamp.br (R.da S. Torres).

http://dx.doi.org/10.1016/j.future.2017.06.013
0167-739X/

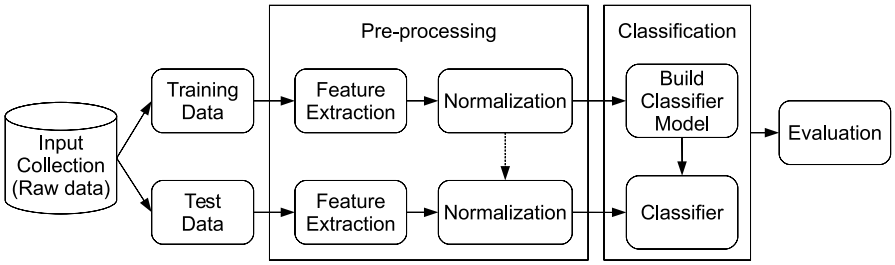


Fig. 1. Organization of a typical machine-learning experiment, composed of a collection input, a split of the collection into train and test sets, extraction of the feature descriptions of the collection, normalization of these features, classification of these features, and the evaluation of the classification results.

Several libraries and machine-learning frameworks have been proposed in the literature to support users in the process of defining the most appropriate methods for their applications. However, many frameworks have limitations including the lack of flexibility to include novel proposed descriptors and machine learning methods, and specially, the inability to reuse previous experiments and learn from them.

In this work, we address these issues by presenting Kuaa, a framework that can be used for designing, deploying, and executing machine learning experiments in an automated fashion. This framework is able to provide a standardized environment for exploratory analysis of machine-learning solutions, as it supports the evaluation of feature descriptors, normalizers, classifiers, and fusion approaches in a wide range of tasks involving machine learning. The Kuaa conceptual model relies on modeling machine-learning experiments as scientific workflows [2,3]. Workflow is the automation of a process, in which information is passed from one resource to another for action, according to a set of rules. The advantages of using workflows are that they are easily understandable, flexible, and reproducible, in which it is possible to redesign them and reproduce their results. The Kuaa's implementation relies on the use of *plugins*, which supports the incorporation of new machine-learning methods as workflow components into the framework, making it flexible to be used in different exploratory analysis.

We also empowered Kuaa with the capability of recommending machine-learning workflows. This service is useful even for experienced users, but specially for beginners in machine learning, as it may guide the user during the configuration of an experiment when facing new and challenging classification problems. The use of recommendations allows users to identify, evaluate, and possibly reuse previously defined successful machine-learning solutions. We also performed experiments in the recommendation system aiming at evaluating four similarity measures (Jaccard, Sørensen, Jaro–Winkler, and a TF–IDF-based measure) in order to define which one is more appropriate for ranking workflows. In addition, we performed experiments with the Learning to Rank using Association Rules (LRAR) method with the objective of comparing it with the methods that do not use any learning mechanism.

Along with this paper, we consider two scenarios in which Kuaa was used to support the identification of appropriate machine-learning solutions. The first one is related to the heart-views classification problem [4], which refers to the automatic recognition of heart view plane of 2D echocardiogram ultrasound images. The second one refers to the produce recognition problem [5,6], which refers to the automatic recognition of fruits and vegetables based on their visual properties. In both contexts, machine-learning-based system developers were interested in comparing several image descriptors (e.g., color, texture, and mid-level representations). By using Kuaa, researchers were able to include novel representations; reuse, based on recommendations, existing workflows previously defined in different contexts; and design, deploy, and perform experiments to assess the effectiveness of tested solutions.

This paper is organized as follows. Section 2 introduces background concepts and presents related work. Section 3 describes the proposed machine-learning framework. Case studies are presented in Section 4, in which we show the use of Kuaa. Section 4 also presents an overview of Kuaa's recommendation system and the experiments performed to evaluate different similarity measures used in the system. Finally, Section 5 presents our conclusions and proposes research directions for future work.

2. Related work and background concepts

This section introduces preliminary concepts related to our proposal, as well as related work on exploratory analysis using machine learning tools and workflow recommendation approaches.

2.1. Machine learning

Machine learning is the study of computational methods that extract useful knowledge from experience to improve performance of a target application [7]. Fig. 1 presents the typical six-step machine-learning classification experiment. Once the input collection is defined, the method selected for splitting it into train and test sets is executed and a feature descriptor is then employed to extract a feature vector from each object within the collection. If a normalization method is selected, the feature vectors of all data in the train and test sets are normalized accordingly. After that, a classification method is applied and the results are analyzed considering the chosen evaluation measure.

2.2. Exploratory analysis using machine learning tools

Several frameworks have been proposed to execute machine-learning experiments. Among the frameworks we may refer to PyML[1] , Accord.NET[2] , mlf [8], and Rattle [9]. PyML (see footnote 1) is an interactive object oriented framework for machine learning written in Python. The framework has implemented the most used classifiers, as Support Vector Machines [10] and Nearest Neighbor [11]. This framework allows combining classifiers and testing classifiers using a typical evaluation process (cross-validation, ROC curves). Accord.NET (see footnote 2) is a framework that provides several scientific computing related methods, such as machine learning, statistics, and computer vision, to the .NET environment. The machine learning framework for Mathematica [3] (mlf) [8] is a collection of machine-learning algorithms for intelligent data analysis, combining an optimized kernel with the manipulation, descriptive programming and graphical capabilities of Mathematica.

[1] http://pyml.sourceforge.net/ (December 2016).

[2] http://accord.googlecode.com (December 2016).

[3] Mathematica is a registered trademark of Wolfram Research Inc. www.wolfram.com (December 2016).

The R Analytical Tool To Learn Easily (Rattle) [9] is an R[4] package that provides a graphical user interface for data mining in the R programming language. GraphLab [12,13] is a parallel framework for machine learning that exploits the sparse structure and patterns of algorithms. It has a collection of applications for some tasks in large-scale graph computation, such as graph analytics, graphical models, computer vision, clustering, and collaborative filtering. Jubatus [14] is a distributed processing framework and streaming machine-learning library. It has a client–server architecture, in which the client side has two commands: `UPDATE`, which corresponds to the training phase of a machine-learning algorithm, and `ANALYZE`, which corresponds to the prediction phase of a machine-learning algorithm. The server side consists of a feature vector preprocessing module and an algorithm module, which supports classification, regression, recommendation of data, simple statistics, and graph analysis.

Different from our initiative, those solutions cannot be used for exploratory analysis concerning the execution of machine-learning experiments. Some of them [8,9,12,13], for example, do not support the inclusion of novel algorithms, while others do not support the possibility of reusing experiment designs defined previously, by means of recommendation [12–14]. Another common limitation refers to the lack of appropriate user interface to support the design of novel experiments modeled as workflows.

2.3. Recommendation-based exploratory analysis in workflow-based systems

Scientific workflows have been used to model complex data analysis procedures in different disciplines [2,3]. The objective is to support the design of experiments, their deployment and execution, and possibly the discovery of the best solutions/parameters/configurations. In some scenarios, scientists may be interested in determining *how* to process properly the available data, while in others, scientists may be interested in *which* possible knowledge can be discovered after processing the data. In both cases, the reuse of previous successful workflows is of paramount importance. In this paper, we propose the use of a recommendation service to support the identification of suitable workflows and possibly their reuse.

Recommendation is defined by Gonçalves [15] as: *given a collection and an actor, and a set of ratings for objects in that collection produced by others or the same actor, recommends (produces a subset of that collection) for that particular actor*. This kind of service is invaluable when the actor has little knowledge about the subject, or even if the actor is an expert in the application. Recommendation systems became an important research area since middle 1990s [16–18] and continues to grow, mainly because of the large volume of contents generated by users of social media. This is a problem-rich research area with several practical applications, such as recommendation of books [19], musics [20], CDs [21], movies [22], social network [23,24], restaurants [25], health insurance [26], e-learning [27], and news [28]. The recommendation problem is usually formulated as a problem of estimating ratings for objects that have not been rated by an actor.[5] Existing recommendation systems are divided into three categories [23,29,30]: content-based, collaborative filtering, and hybrid approaches.

WOrkflOw-based spatial Decision Support System (WOODSS) [31,32] is a computational tool implemented to be used in conjunction with Geographical Information System (GIS). This system is centered in monitoring the user activities in GIS and documenting them by means of scientific workflows. Kaster et al. [33] presented the use of Case-Based Reasoning (CBR) [34] as a retrieval mechanism in the WOODSS [31,32] to help users choose the most adequate models from those available in the database. CBR is a reasoning model, which consists in solving new problems by adapting solutions that were already used to solve previous problems [34]. The similarity retrieval applied with the CBR approach uses the metadata associated with each WOODSS workflow, which contains the problem focused by the workflow and its meaning. The process of similarity analysis employed by the CBR system is described as the following steps: (i) Find correspondences, aligning the input problem with the stored workflows; (ii) compute the degree of similarity of corresponding features; and (iii) assign importance values to features. The WOODSS' CBR mechanism uses city-block metrics to calculate the similarity evaluation between the input and the stored workflow.

Conforti et al. [35] proposed a recommendation system to present risk-informed decision to users in Business Process Management (BPM) when partaking in multiple process instances running concurrently. This recommendation system runs as a plugin to YAWL (Yet Another Workflow Language) BPM,[6] in which when an input of the user is required, the recommender determines the risk of the user input. Their recommendation uses a predicting-risks technique, and a technique to assign the best participants to the work items currently on offer. The provided recommendation significantly reduced the severity of faults in a simulation of the real life scenario.

Chong et al. [36] presented an adaptive workflow recommendation engine based on collaborative analytics matches with workflows stored in repositories. Then, top-*n* matched previously defined workflows are the input of a Genetic Programming framework, which is based on evolutionary operators (e.g., selection, crossover and mutation). The objective of this framework is the discovery of the workflow with the highest fitness. The resulting workflows were comparable in accuracy to the best benchmark workflows created by experts.

Zhou et al. [37], in turn, proposed the use of semantic similarities for scientific workflows using a layer-hierarchy representation. First, they describe the scientific workflow as a tuple, based on which they calculate the similarity between two workflows. Then the scientific workflows are clustered based on an workflow network model using a graph-skeleton-based method. Barycenters are determined to represent each cluster. Finally, to recommend a scientific workflow, the user-defined workflow is compared to each cluster representative. The top-*k* similar workflows are recommended to the user.

Zhang et al. [38] developed a plugin for VisTrails as a recommendation engine based on a social network. Workflows and services are modeled as social nodes in a Service Social Network, in which edges are defined based on their usage or authorship. Then various metrics are calculated to comprehend the interactions between workflows and services. To perform the recommendation, a user's query is assigned and matched with service categories in the repository. Then, the categories of the query and the categories of the services are compared using the rank-biased overlap algorithm to generate the candidate list, ranked from the most similar to the less similar.

Different from the above approaches, in this paper we propose a learning-to-rank-based recommendation service aiming to support the workflow-based design of machine-learning experiments. To the best of our knowledge, our work is innovative as we present

[4] http://www.r-project.org/ (December 2016).

[5] For further investigation on existing solutions for recommendation services, the reader may refer to the surveys of Almazro et al. [29] and Bobadilla et al. [30].

[6] http://yawlfoundation.org/ (December 2016).

for the first time the use of learning-to-rank methods in the recommendation of workflows.

2.4. Learning to rank: LRAR

As we shall explain shortly, to recommend a machine-learning experiment within our framework, we rely upon a learning-to-rank method based on association rules. We describe the concept of learning to rank and the learning-to-rank approach used in this work in the next sections.

2.4.1. Learning to rank

Ranking models and functions is an important research topic in many fields. In the literature, several empirical ranking methods are proposed, such as boolean, vector space, and probabilistic models [39]. However, it is difficult to empirically tune the parameters of ranking functions of the above methods, therefore, recently, learning-to-rank approaches have been proposed [39]. These methods exploit machine-learning methods to automatically learn effective ranking functions.

The task of learning to rank is defined as follows. We have a set D of training data consisting of tuples $\langle q, d, r\rangle$, where q is a query, d is a document, represented as a list of features $f_1, f_2, \ldots, f_n$, and r is the relevance of d to q in discrete value. D is used to create a model to relate the features of the document to the corresponding relevance. The test set T consists of tuples $\langle q, d, ?\rangle$, where the relevance of the document d for the query q is unknown. The model learned is used to produce a likelihood of relevance of such documents to the corresponding queries, which are used to generate the final ranking.

2.4.2. Learning to rank using association rules

Learning-to-rank methods in the literature rely on techniques such as Support Vector Machines [10,40], Neural Networks [41], and Genetic Programming [42] to learn effective ranking functions. Veloso et al. [43] proposed an alternative method using associative rules [44], that generates a model R, composed of rules of the form $f_i \cap \cdots \cap f_j \rightarrow r$, describing the training data by feature–relevance associations. Once the model is built, the rules are used to estimate the relevance of documents in the test set. There are two measures used to quantify the quality of a rule: the confidence θ (conditional probability of relevance r given $f_i \cap \cdots \cap f_j$) and the support σ (fraction of training examples containing features $f_i \cap \cdots \cap f_j$ and relevance r).

To explain the Learning to rank using association rules, consider the rules shown in Table 1 for the use of a tennis court, taking into account the weather (outlook, temperature, humidity, and windy).

We can generate rules from each information about the weather to estimate if that can be a game or not. However, generating every rule for the training data is costly. The method of Veloso et al. [43] generates the rules on a demand-driven basis, making it more efficient. Using the id 10 as an example, only a few training data will be used to generate the rules, as Table 2 shows.

We can generate the following rules considering the information present in each training data from Table 2, ordered by its confidence:

- overcast → yes ($\theta = 1.00, \sigma = 0.22$)
- mild → yes ($\theta = 0.50, \sigma = 0.11$)
- mild → no ($\theta = 0.50, \sigma = 0.11$)
- high → yes ($\theta = 0.40, \sigma = 0.22$)
- high → no ($\theta = 0.60, \sigma = 0.33$)
- true → yes ($\theta = 0.33, \sigma = 0.11$)
- true → no ($\theta = 0.77, \sigma = 0.22$)
- overcast & high → yes ($\theta = 1.00, \sigma = 0.11$)
- overcast & true → yes ($\theta = 1.00, \sigma = 0.11$)
- mild & high → yes ($\theta = 0.50, \sigma = 0.11$)
- mild & high → no ($\theta = 0.50, \sigma = 0.11$)
- high & true → no ($\theta = 1.00, \sigma = 0.11$).

These rules are combined to estimate the relevance of test id 10, and the score of each rule is weighted according to its confidence defined by

$$s(r_i) = \frac{\sum_{r_i \in R_d} \theta(r_i)}{|R_d|}, \tag{1}$$

where R_d is the total number of rules generated and $\theta(r_i)$ are the rules confidence with relevance r_i.

Therefore, the rank of a test is estimated by the linear combination of the normalized score for each relevance:

$$rank = \sum_{i=0}^{k} r_i \frac{s(r_i)}{\sum_{j=0}^{k} s(r_j)}. \tag{2}$$

Therefore, we have that the rank of test id 10 is: $s(yes) = 0.394$ and $s(no) = 0.198$. Considering $yes = 1$ and $no = 0$, we calculate the rank, obtaining $rank = 0.666$. With these results, we can confirm the relevance of the test id 10 as 1, i.e., the game can be played, as the weather will not interfere with it.

3. Kuaa

In this section, we introduce Kuaa, a framework for designing, deploying, executing, and recommending machine-learning experiments. We selected this name because "kuaa" is a suffix in the Guarani language, which means "learn", just like our framework proposes to help the deployment and execution of machine-learning experiments. In the next sections, we describe how the proposed machine-learning framework is structured and how Kuaa was implemented, describing its architecture and modules, as well as the employed plugin scheme.

3.1. Modeling a machine-learning experiment as a workflow

In the proposed framework, we model each machine-learning experiment step (e.g., feature extraction, normalization, and classification) as a workflow activity, allowing the user to construct a machine-learning experiment as a workflow. The workflow representation is modeled as a directed acyclic graph (DAG), in which its activities are depth-first traversed, usually beginning with a collection and ending with an evaluation measure. Fig. 2 shows a representation workflow of a typical machine-learning experiment, with all the typical six steps. The first component (in red) refers to the selection of a collection (dataset). In purple, we highlight the component related to the selection of an approach for splitting the collection into train and test sets. The selection of a descriptor (feature extractor) is represented in dark green. The next module refers to the optional selection of a normalization method (blue). In brown, we represent the component that encodes the selection of a classification method. Finally, in light green, we present the module associated with the selection of an evaluation measure to assess the effectiveness performance of the specified machine-learning experiment. Those colors are also used in the Kuaa prototype created to refer to the same workflow component.

3.2. Architecture and overview

The objective of the proposed machine-learning framework is to facilitate the automation of classification experiments. It is responsible for managing the steps of machine-learning experiments. Each step consists of a module in the framework (e.g., collection, train and test set definition, feature extraction, normalization, classification, fusion, and evaluation), that runs independently of others. Efficiency aspects on the workflow executions

Table 1
Weather data relative to the occurrence or not of a tennis game. Outdoor tennis games do only occur when the weather has a little impact on the game. Weather data extracted from WEKA library [45].

	Weather					Game
	id	Outlook	Temperature	Humidity	Windy	
Training data	1	Sunny	Hot	High	False	No
	2	Sunny	Hot	High	True	No
	3	Overcast	Hot	High	False	Yes
	4	Rainy	Mild	High	False	Yes
	5	Rainy	Cool	Normal	False	Yes
	6	Rainy	Cool	Normal	True	No
	7	Overcast	Cool	Normal	True	Yes
	8	Sunny	Mild	High	False	No
	9	Sunny	Cool	Normal	False	Yes
Test data	10	Overcast	Mild	High	True	Yes
	11	Overcast	Hot	Normal	False	Yes
	12	Rainy	Mild	High	True	No

Table 2
Learning to Rank using Association Rules for the document id 10. In this test data, Veloso et al. [43] proposes to select only the rules related to the test data. Training data id 5 and 9 were ignored as they do not have any information equal to the test data id 10. Then the rest of the data is used to generate the rules.

	Weather					Game
	id	Outlook	Temperature	Humidity	Windy	
Training data	1	Sunny	Hot	High	False	No
	2	Sunny	Hot	High	True	No
	3	Overcast	Hot	High	False	Yes
	4	Rainy	Mild	High	False	Yes
	5	–	–	–	–	Yes
	6	Rainy	Cool	Normal	True	No
	7	Overcast	Cool	Normal	True	Yes
	8	Sunny	Mild	High	False	No
	9	–	–	–	–	Yes
Test data	10	Overcast	Mild	High	True	Yes

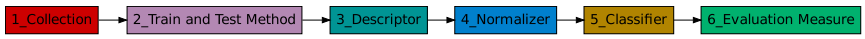


Fig. 2. Workflow representation of a typical machine-learning experiment, containing the typical six steps: Collection selection, train and test split, feature extraction, normalization, classification, and evaluation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are addressed by exploiting multiple cores in the extraction and normalization modules.

Fig. 3 depicts the architecture of Kuaa. It consists of three layers: an interface, the core of the framework, and a set of repositories. The interface is responsible for the communication with the user. Using the interface modules, users can design, call the execution of a workflow, and receive the results of the execution. In the design of an experiment, the core of the framework is in charge of designing the machine learning experiments using the modules of the repository and executing the experiments, processing the communication between the modules in execution. Once the execution is done, the workflow and its results are stored in a repository, which can be used later for making recommendations. The recommendation service can help users in their task of building a workflow experiment, by avoiding common errors and providing best practices from the past (for more details, see Sections 3.4 and 4.3).

Each one of these modules in the repository layer is responsible for a step in a machine-learning experiment. The *Collections* module is responsible for gathering the objects of the collection for the framework. The *Train-and-Test* module splits the objects of the collection into two sets, a training set, which is used to generate a classifier model and a test set, used to test the built model. Different partition strategies may be chosen, such as K-Fold cross-validation and randomized samples. The *Feature Extraction* module extracts feature vectors from the objects in the collection (e.g., BIC, HOG). The normalization of the feature vectors of all data is responsibility of the *Normalization* module (e.g., TF–IDF), and the *Classification* module performs the classification of the test set using the model learned with the training set (e.g., SVM). Finally, the results of the classification are obtained in the *Evaluation* module (e.g., Global Accuracy Score).

While there might be some drawbacks implementing a new framework and a new workflow rather than using an existing one (e.g., lack of interaction with the workflow community and potential bugs of connections among old and novel modules), it has several advantages. First, a new workflow has the connections between its modules defined a priori, avoiding errors introduced by users. Second, using the defined modules, a specialist is more productive on designing and deploying experiments. Last, the insertion and implementation of recommendation tools is easier in a new framework, as we are aware of their specifications.

3.3. Implementation aspects

In this section, we present the main decisions for the implementation of the machine-learning framework, such as the plugin scheme and the use of XML-based documents representing a workflow experiment.

3.3.1. Plugin scheme

In order to provide an extensibility functionality in the build framework, we decided to use a plugin scheme inspired by the design used in Eva tool [46]. Plugins consist of components that implement or encapsulate new features to an existing software, developed according to defined standards and interfaces [47].

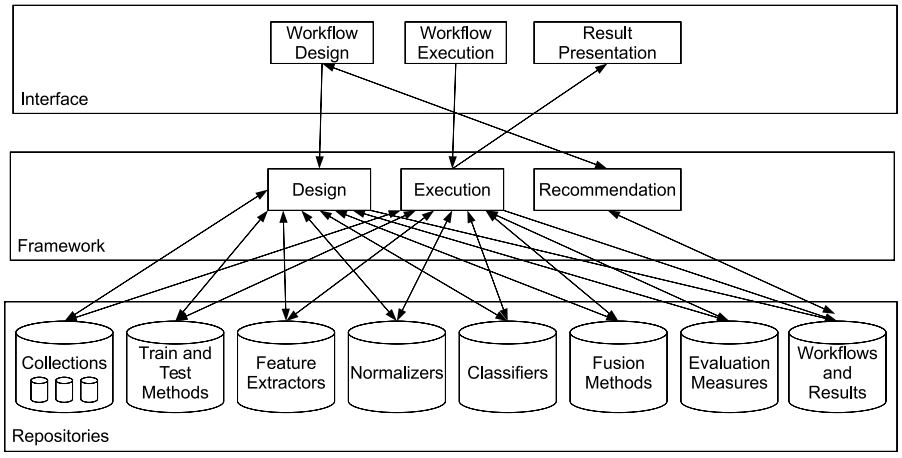


Fig. 3. Architecture of the framework to automate machine-learning experiments. It is divided into three layers: Interface, the core of the framework, and repositories. The interface module is responsible of presenting the design and result of the experiment. The core of the framework is responsible of connecting the methods and executing them. The repository layer contains a set of repositories for each module of the framework, and stores previous executed experiments.

Each module of the machine-learning framework is composed of plugins of different implemented methods. This plugin scheme makes the framework more flexible and easily extensible, i.e., it is possible to define methods in any programming language and add them to the framework using a Python wrapper. The plugins are organized according to the step of a machine learning experiment (module). The majority of ML algorithms/components present in Kuaa were implemented based on plugins created using functions available in the Scikit-Learn library [48]. Table 3 summarizes the plugins available in the Kuaa framework.

3.3.2. XML documents

For organizing experiments, we store the information using the eXtensible Markup Language (XML)[7] format. The XML documents store the values of parameters of the methods implemented in each plugin and the setup of a machine learning experiment. The XML file format was chosen due to its flexibility, portability, and ease management by computers and, in some cases, by humans.

One use of an XML document in the framework is in the representation of a workflow experiment, which describes each module and each method in the built workflow. The workflow is represented as a graph, with a list of modules and a list of links between the modules (nodes and edges, respectively). A root tag "experiment" contains a name for the experiment, the name of the author, the number of iterations in the experiment and a date and hour control. Child tags of the root are: the modules present in the workflow, with an identification for the module, the plugin selected and its parameters; and a "link" tag representing the input and output links for each module id. Fig. 4 shows the schema of an experiment XML file.

With the XML document of the machine learning experiment, the framework traverses the workflow as in a deep-first traversing, beginning in the Collections module and following the output links present in the XML file until there is no more modules to be visited. This method favors the execution of a whole branch in the machine learning experiment, saving the result of the branch, and avoiding possible conflicts between the results of two different branches on a parallel execution of modules.

3.3.3. Open-set scenario

Kuaa framework was designed to execute machine-learning experiments. A typical classification assigns a test sample to one or more known classes (e.g., classifying the image of a digit), however, in case a classifier cannot be trained with all classes because they are ill-sampled or unknown, typical classification methods do not work, as they classify an unknown sample wrongly as a known class. This scenario is called open-set scenario [87,88].

Kuaa framework deals with the open-set scenario providing an option to modify the experiment to the open-set scenario. Therefore, the Collection module is modified to allow the user to select the known and unknown classes for the experiment, and allows that exclusive methods which deal with open-set scenario to be selected and executed in the framework.

3.4. Recommendation system

The objective of the recommendation system is to support the reuse of experiments and activities successfully used in the past, avoiding common mistakes in workflow design. To make the recommendations, we implemented four similarities measures to estimate the proximity of two workflow experiments and a *learning-to-rank* method that "learns" how to rank workflows according to the users interests.

3.4.1. Similarity measures

The similarity measures implemented in the framework consider that an experiment workflow is a textual sentence, in which each module is represented by a word in the sentence. To calculate the distance between two sentences, we implemented the Jaccard [89], Sørensen [90], and Jaro–Winkler [91–93] distances and a measure based on Term Frequency–Inverse Document Frequency (TF–IDF) [70] to represent an experiment workflow. We use similarity measures to find wokflows with similar characteristics to each other, in order to recommend workflows similar to the one build by the user.

Jaccard Distance: Let A and B be two sequences. The Jaccard index calculates the similarity between these two sequences using Eq. (3). Eq. (4) measures the distance between A and B.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

$$d_J(A, B) = 1 - J(A, B). \tag{4}$$

[7] http://www.w3.org/XML/ — as of November 2016.

Table 3
List of the plugins implemented in the framework.

Modules	Number of plugins	List of plugins
Train and Test	5	K-fold [49], Leave Video Out [49], Number of Images, Percentage of Images, Read Files
Extraction	18	ACC [50], Bag of Visual Words, BIC [51], CCOM [52], CCV [53], CEDD [54], GCH [55], Gist [56], HOG [57], HTD [58,59], JAC [60], LAS [61], LBP [62], M-SPyd [63], QCCH [64], SASI [65,66], SMD [67], Unser [68]
Normalization	4	Min–Max [69], Term Frequency [70], TF–IDF [70], Z-Score [69,71]
Classification	8	DecisionTree [72], kNN [73], LDA [74,75], lib-SVM [76], LogisticRegression [77], OPF [78], SVM [76], SVMDBC [79,80]
Fusion methods	2	Concatenation [81], Majority Voting [82]
Evaluation measures	10	Confusion Matrix [83], False Negative, False Positive, F-measure, Global Accuracy Score, Cohen's Kappa [84], Normalized Accuracy Score, ROC curve [85,86], True Negative, True Positive

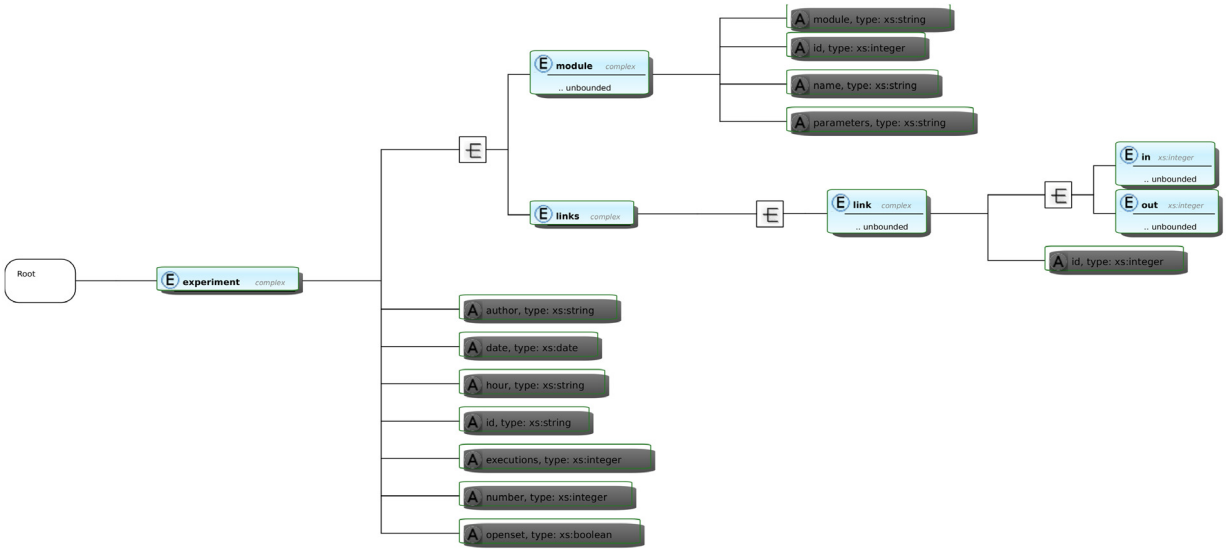


Fig. 4. XML Schema of an experiment in XML document. The XML tree is composed of the root "experiment", which describes metadata of the experiment, such as the name of the author, name of the experiment, date and hour of execution, number of executions, and if it is an open-set experiment. Child elements of the root are the "modules" of the experiment, in which their attributes detail the selected plugin and its parameters, and "links" tag, which represents the link between the output and input modules.

Sørensen Distance This measure, represented by Eq. (5), was originally used to be applied to presence/absence of data [90]. Eq. (6) represents the distance between two samples A and B.

$$QS = \frac{2|A \cap B|}{|A| + |B|} \tag{5}$$

$$d_{QS} = 1 - QS. \tag{6}$$

Jaro–Winkler Distance The Jaro–Winkler similarity measure is composed of two algorithms. The similarity between two strings is defined as

$$d_j(A, B) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|A|} + \frac{m}{|B|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{7}$$

where m is the number of matching characters, if they are the same and not farther than

$$\left\lfloor \frac{\max(|A|, |B|)}{2} - 1 \right\rfloor,$$

and t is the number of matches in different sequence order, divided by 2. Eq. (8) is an extension of the Jaro distance that gives favorable ratings to strings that match from the beginning.

$$d_w(A, B) = d_j(A, B) + (lp(1 - d_j(A, B))), \tag{8}$$

where l is the length of common prefix at the start of the string, and p is a constant scaling factor ($p = 0.1$).

TF–IDF-based Distance This distance measure relies on the fact that, if a workflow is often used, it should be recommended. This measure calculates the Inverse Document Frequency (IDF) of each item of the sequences in the previous experiments, measuring whether the item is common or rare in the sequences. Let p be an item of a experiment workflow, s a experiment workflow, and S the set of previous experiments of the framework. The IDF of each item is obtained by dividing the total number of sequences N_S by the number of sequences that contains the item, and taking the logarithm of the quotient as

$$\text{idf}(p, S) = \log \frac{N_S}{|s \in S : p \in s|}. \tag{9}$$

The Term Frequency of an item in a experiment workflow ($\text{tf}(p, s)$) is represented by the raw frequency of the item in the sequence. Therefore, the feature vector of a sequence is the product of this two statistics for each item of the sequence, defined as

$$\text{tf–idf}(p, s, S) = \text{tf}(p, s) \times \text{idf}(p, S). \tag{10}$$

With the feature vector of two sequences, the distance between them is calculated using the Euclidean distance.

3.4.2. Learning to rank using association rules

Consider the training and test sets shown in Table 4, with three queries for the training data and one query for test data, and each query having three documents associated, represented by the workflows. We can generate rules from each step of the workflow experiment to estimate the relevance of a new document.

By using the method of Veloso et al. [43] to generate the rules on a demand-driven basis for id 11, only a few documents will be used, as Table 5 shows.

We can generate the following rules considering the steps present in each document from Table 5, ordered by its confidence:

- Classification (MCOCSVM) $\rightarrow r = 0\,(\theta = 1.00, \sigma = 0.11)$
- Feature extraction (CCV) $\cap$ Classification (MCOCSVM) $\rightarrow r = 0\,(\theta = 1.00, \sigma = 0.11)$
- Normalization (Min–Max) $\cap$ Classification (MCOCSVM) $\rightarrow r = 0\,(\theta = 1.00, \sigma = 0.11)$
- Feature extraction (CCV) $\cap$ Normalization (Min–Max) $\cap$ Classification (MCOCSVM) $\rightarrow r = 0\,(\theta = 1.00, \sigma = 0.11)$
- Feature extraction (CCV) $\rightarrow r = 0\,(\theta = 0.75, \sigma = 0.33)$
- Normalization (Min–Max) $\rightarrow r = 1\,(\theta = 0.66, \sigma = 0.44)$
- Feature extraction (CCV) $\cap$ Normalization (Min–Max) $\rightarrow r = 0\,(\theta = 0.66, \sigma = 0.22)$
- Normalization (Min–Max) $\rightarrow r = 0\,(\theta = 0.33, \sigma = 0.22)$
- Feature extraction (CCV) $\cap$ Normalization (Min–Max) $\rightarrow r = 1\,(\theta = 0.33, \sigma = 0.11)$
- Feature extraction (CCV) $\rightarrow r = 1\,(\theta = 0.25, \sigma = 0.11)$

These rules are combined to estimate the relevance of document id 11, and the score of each rule is weighted according to its confidence defined by

$$s(r_i) = \frac{\sum_{r_i \in R_d} \theta(r_i)}{|R_d|}, \tag{11}$$

where R_d is the total number of rules generated and $\theta(r_i)$ are the rules confidence with relevance r_i.

Therefore, the rank of a document is estimated by the linear combination of the normalized score for each relevance:

$$rank = \sum_{i=0}^{k} r_i \frac{s(r_i)}{\sum_{j=0}^{k} s(r_j)}. \tag{12}$$

Therefore, we have that the rank of document with id 11 is: $s(0) = 0.574$ and $s(1) = 0.124$, and $rank = 0.178$. With these results, we can confirm the relevance of the document id 11 for the test query as 0.

4. Validation

In this section, we present case studies (Section 4.1), in which we create workflow experiments. Next, in Section 4.2, we show the use of the framework in a real-world setup concerning the evaluation of machine learning algorithms in the context of the produce recognition problem. Finally, Section 4.3 presents an overview of the Recommendation module and the conducted experiments related to its validation.

4.1. Case studies

Kuaa framework is capable of designing different scientific wokflows for different applications. Fig. 5 shows two different applications designed in the Kuaa framework. The first is the heart view plane classification of echocardiogram [4] using two descriptors (Fig. 5a), and the second is the produce identification [5,94] (Fig. 5b).

The heart view plane classification workflow split the dataset selecting one of the echocardiograms video frames as the test set, and using the rest of the frames as training. Then these frames are described using two feature extractors, a Bag of Visual Words and SASI, which are fed in a SVM classifier. The evaluation plugins will present the global accuracy score and the confusion matrix of the classification. The produce identification workflow, in turn, has the same modules used in the previous application, except for the addition of a normalization module. The differences between the workflows for produce identification and for heart view classification are only the plugins and the collection selected for the scientific experiment. This shows that the Kuaa framework is robust to different applications, and the plugin scheme makes it flexible.

From now on, the usage of the Kuaa system will be illustrated in the context of the produce identification problem. Suppose that we want to compare the results of machine-learning solutions using two different methods of feature extraction. For exemplifying this experiment, we selected a fruit and vegetable identification problem, a recurrent task in supermarkets. This problem is defined as following: given a produce image, identify its species (e.g., apple, potatoes, oranges) and its variety (e.g., Gala and Fuji apples) with the objective of determining its price [5,6]. Fig. 5b shows this experiment designed in Kuaa framework.

For this experiment, we selected a representative collection of fruits [5], with 15 classes and 2633 images. The K-Fold plugin was selected to split this collection into training and test sets. The plugin splits the collection into 3 folds, in which each fold is used as test, and the other two as the training set. For the Feature Extraction module, we select two descriptor plugins, Border/Interior Pixel Classification (BIC) [51] and Local Activity Spectrum (LAS) [61]. These descriptors extract the feature vectors based on color and texture visual properties, respectively. With those two methods, we can compare the results with the objective of determining which descriptor performs better in this collection. For fair comparison, the steps following the Feature Extraction module have to be identical. The selected plugin for the Normalization module was the Min–Max [69] method, with parameters min $=$ 0.0 and max $=$ 1.0. For the Classification module, the libSVM [76] plugin was selected with linear kernel with a grid-search for the other parameters. To present the results of the classification, the Global Accuracy Score and Confusion Matrix evaluation measures were selected. The Global Accuracy Score shows the mean accuracy of the classification for each fold, and the Confusion Matrix plugin plot the mean confusion matrix of the folds, showing the percentage of images classified as each class.

4.2. Interactive exploratory analysis using Kuaa

For designing workflow related to the produce recognition problem, users can select a module from a circle-shaped selection menu with all of the modules of the framework, as shown in Fig. 6. By selecting one of the modules, it will be added to the workflow design area. For example, we selected the Collections module. Upon right clicking on this module box, it opens a window with a list of available plugins in the framework (Fig. 7). In this example, the user selects the dataset *tropical fruits*, originally presented in [5]. With two or more modules built in the framework, it is possible to connect them, creating a relationship between the modules, as shown in Fig. 8. In the example showed in the figure, after selecting the dataset, the users opted for a k-fold experimental protocol. Fig. 9 presents the experiment described using two descriptors for comparison of the results: BIC and LAS. With the workflow completely designed, the user can execute the experiment.[8]

[8] A video example of the design of this workflow at Kuaa is present in the supplementary materials.

Table 4
Queries, documents and relevance for three training queries and one test query. Each step of the experiment will generate rules to estimate the relevance of a test document.

	Query	Documents		Relevance
		id	Workflow	
Training data	Query 1	1	1_scene_categories → 2_number_of_images → 3_ccv → 4_min_max → 5_LDA → 6_global_accuracy_score	1
		2	1_scene_categories → 2_number_of_images → 3_las → 4_min_max → 5_MCSVMSH → 6_global_accuracy_score	1
		3	1_scene_categories → 2_percentage → 3_ccv → 4_min_max → 5_LogisticRegression → 6_global_accuracy_score	0
	Query 2	4	1_auslan → 2_k_fold → 3_ccv → 4_min_max → 5_MCOCSVM → 6_global_accuracy_score	0
		5	1_auslan → 2_k_fold → 3_sasi → 4_min_max → 5_kNN → 6_global_accuracy_score	1
		6	1_auslan → 2_k_fold → 3_las → 4_tfidf → 5_SVM → 6_global_accuracy_score	0
	Query 3	7	1_tropical_fruits → 2_number_of_images → 3_ccv → 4_tfidf → 5_MCSVMexternal → 6_global_accuracy_score	0
		8	1_scene_categories → 2_k_fold → 3_ccv → 4_tfidf → 5_SVM → 6_global_accuracy_score	0
		9	1_tropical_fruits → 2_number_of_images → 3_gch → 4_min_max → 5_SVM → 6_global_accuracy_score	1
Test data	Query test	10	1_aloi → 2_number_of_images → 3_sasi → 4_min_max → 5_SVM → 6_global_accuracy_score	1
		11	1_aloi → 2_number_of_images → 3_ccv → 4_min_max → 5_MCOCSVM → 6_global_accuracy_score	0
		12	1_aloi → 2_number_of_images → 3_sasi → 4_min_max → 5_OPF → 6_global_accuracy_score	1

Table 5
Learning to Rank using Association Rules for the document id 11. In this test query, Veloso et al. [43] proposes to select only the rules related to the test query (specifically the CCV feature extraction, Min–Max normalization, and MCOCSVM classification). Document id 6 was ignored as it does not have any step equal to the document id 11 (as the train and test split and evaluation steps do not interfere with the results, they were not considered).

Documents		Relevance
id	Workflow	
1	1_scene_categories → 2_number_of_images → 3_ccv → 4_min_max → 5_LDA → 6_global_accuracy_score	1
2	1_scene_categories → 2_number_of_images → 3_las → 4_min_max → 5_MCSVMSH → 6_global_accuracy_score	1
3	1_scene_categories → 2_percentage → 3_ccv → 4_min_max → 5_LogisticRegression → 6_global_accuracy_score	0
4	1_auslan → 2_k_fold → 3_ccv → 4_min_max → 5_MCOCSVM → 6_global_accuracy_score	0
5	1_auslan → 2_k_fold → 3_sasi → 4_min_max → 5_kNN → 6_global_accuracy_score	1
6	–	0
7	1_tropical_fruits → 2_number_of_images → 3_ccv → 4_tfidf → 5_MCSVMexternal → 6_global_accuracy_score	0
8	1_scene_categories → 2_k_fold → 3_ccv → 4_tfidf → 5_SVM → 6_global_accuracy_score	0
9	1_tropical_fruits → 2_number_of_images → 3_gch → 4_min_max → 5_SVM → 6_global_accuracy_score	1
11	1_aloi → 2_number_of_images → 3_ccv → 4_min_max → 5_MCOCSVM → 6_global_accuracy_score	0

At the end of the execution of the workflow, the framework opens a PDF file containing the results of the experiments. In this case, the BIC method had 98.48% of global accuracy against 74.86% on the branch with the LAS descriptor. To view the PDF generated at the end of the execution of this experiment, the reader must refer to the supplementary materials.

4.3. Workflow recommendations

In this section, we present an overview of the recommendation system execution, with the methods used to perform the recommendation and the experiments made to evaluate the system.

4.3.1. Overview

To make a recommendation, the framework uses the workflow that is being built in the ongoing experiment configuration (the *query* workflow) to search for similar workflows in previous experiments. Fig. 10 shows the "Recommend" button, which initiates the recommendation system, based on the simple workflow experiment built in the framework.

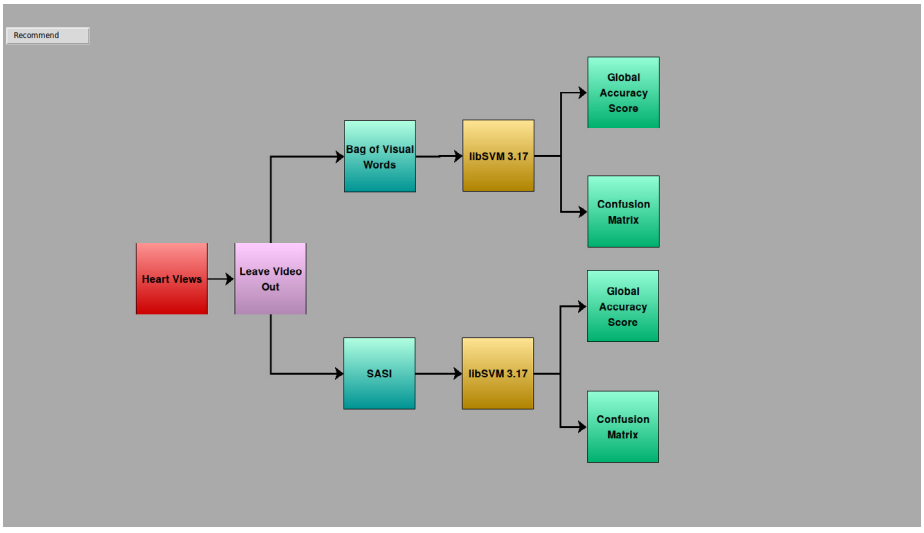
When the user clicks on the "Recommend" button, the framework opens a window with a list of recommendation plugins implemented in the framework. These plugins are responsible for ranking the previous workflows of the framework according to the query, and show them to the user. Fig. 11 shows the list of plugins in the framework.

When the user selects the plugin and clicks in the "Recommend" button in the new window, the framework starts the execution of the Recommendation module. At the end of the execution, the framework lists five workflow experiments that are similar to the workflow that is being built in the framework, according to the selected method. These similar workflows provide to the users ideas of other plugins to use in their experiments. Fig. 12 depicts the recommended workflow experiments. The most similar workflows retrieved are expected to be the most relevant ones, given the user needs represented by means of the query workflow.

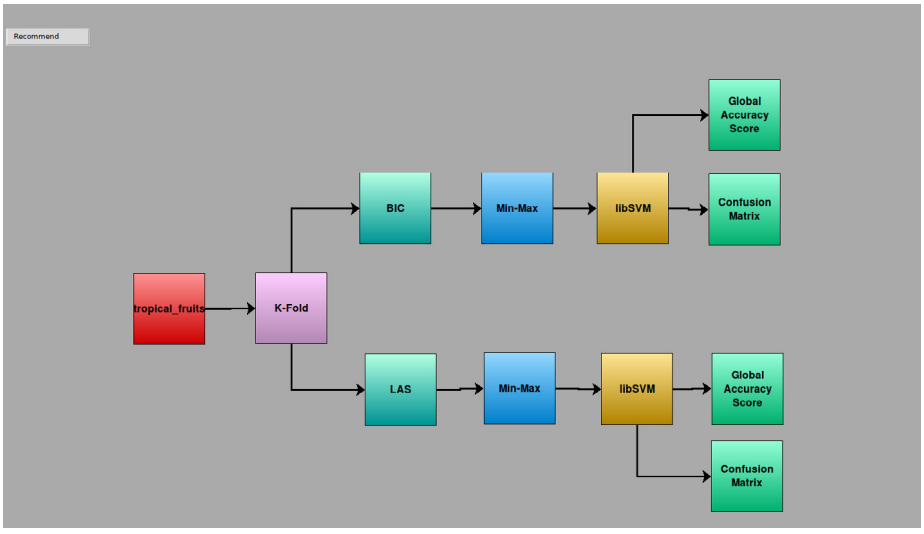
4.3.2. Experiments

Our experiments aim at addressing two different research questions: (i) *Which similarity measure is more appropriate for ranking workflows modeled as a sequence of activities?* and (ii) *Is the use of learning-to-rank methods a suitable research venue for ranking workflows?*

In order to address the first question, we performed experiments in the recommendation system using four similarity measures (Jaccard, Sørensen, Jaro–Winkler, and a TF–IDF-based measure). For the second question, we performed experiments with the Learning to Rank using Association Rules (LRAR) with the objective



(a) Heart view plane classification of echocardiograms using two different descriptors, Bag of Visual Words and SASI.



(b) Produce identification using two different descriptors, BIC and LAS.

Fig. 5. Comparison between the design of two different applications in the Kuaa framework.

of comparing its accuracy performance with the methods that do not use any learning mechanism.

To perform these experiments, we had to define a ground-truth that indicates the relevance of experiments (i.e., the relevance of workflows). To obtain this ground-truth, we invited five specialists in machine-learning experiments to label workflows as relevant or not for some queries. For this, we randomly created 1000 workflow experiments, and selected 18 of those workflows as queries.

For each query workflow, we applied the four similarity measures (Jaccard, Sørensen, Jaro–Winkler, and a TF–IDF-based measure) with the 1000 entries. For each measure, we ranked and selected the closest 20 workflows to the query, and presented these ranked workflows to all those specialists, so that they could label which ones are relevant for the query, as shown in Fig. 13.

To compare the results of different similarity methods with the LRAR method, we applied a majority voting scheme based on the labels provided by each specialist. One workflow is labeled as relevant if most of the specialists agree on that. In the end, we have the ground truth for 18 queries merging the results of the four similarity methods. These labels are then used to train the LRAR method.

With 18 queries, we split these queries into five folds, where four folds contain the training set of the LRAR method, and the fifth fold contains the test queries. The training set of the LRAR is composed of the queries of four folds, and each query has no more than 80 workflows (the 20 closest workflows for each similarity method). For each workflow, its features in the learning-to-rank method are the distance between the workflow to its query calculated by the similarity methods, and the relevance of the workflow is the majority voting of the relevance labeled by each user for each query. The LRAR method implemented uses a minimum confidence value θ and a minimum support value σ to limit the amount of rules created. We executed the LRAR method with seven values of confidence (0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5) and five values of support (0.01, 0.05, 0.1, 0.15, 0.2). The precision values on the top five ranked workflows ($P@5$) for the variation of the confidence and support are shown in Fig. 14.

We can see in Fig. 14 that the variation of the support value did not affect the result of the precision. However, an increase in the value of the minimum confidence limits the power of the LRAR method, and smaller values add noise from weak rules to the

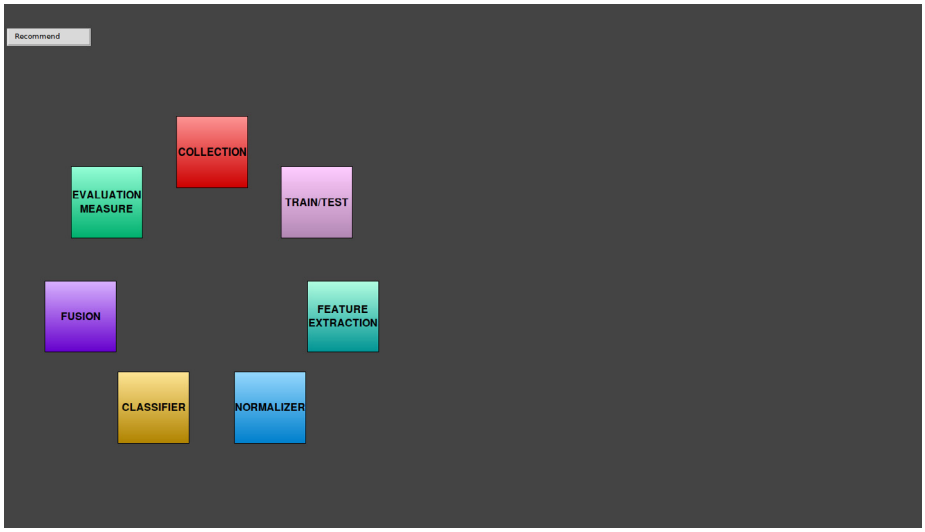


Fig. 6. Modules of the framework as presented to the user. To include a module/step of the workflow into the Kuaa framework, the user clicks the design area, and the modules are presented as in the figure. Selecting a module, it will be added in the workflow design area.

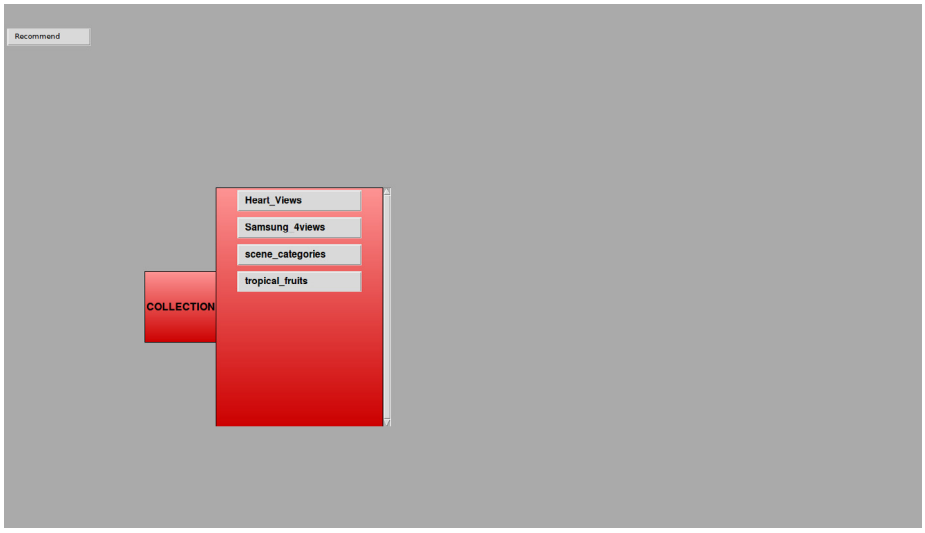


Fig. 7. List of available collections of the Collection module. Other modules present the plugins available in them. Selecting a plugin allows the settings of its parameters.

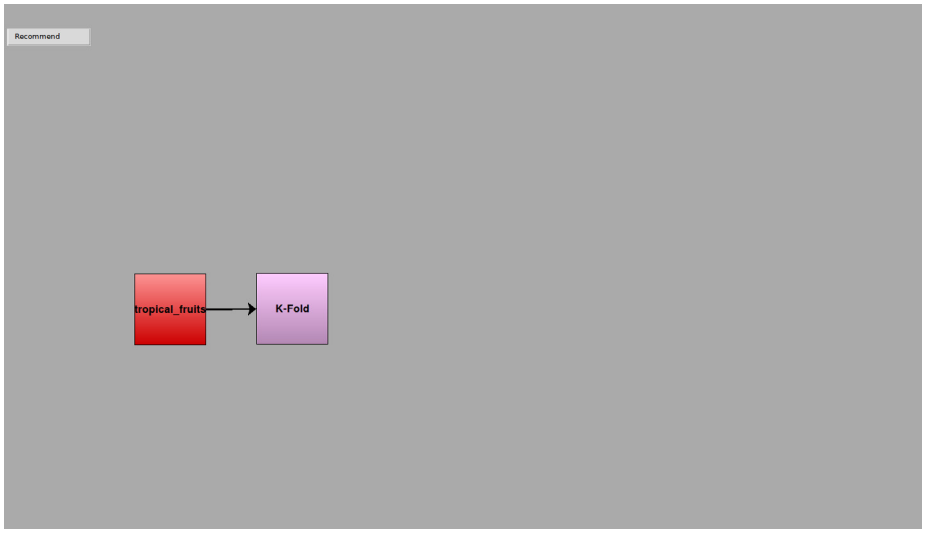


Fig. 8. Two modules of the framework linked. The Collections module (Tropical Fruits) is used as input for the Train-and-Test module (K-Fold) in the execution.

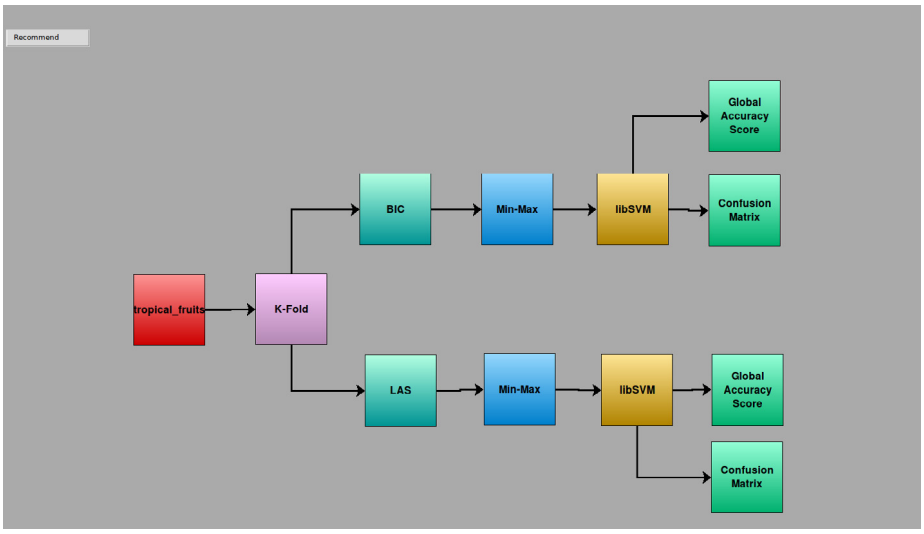


Fig. 9. Complete machine learning experiment comparing the classification results (global accuracy score and confusion matrices) when using two different feature extractors (BIC and LAS).

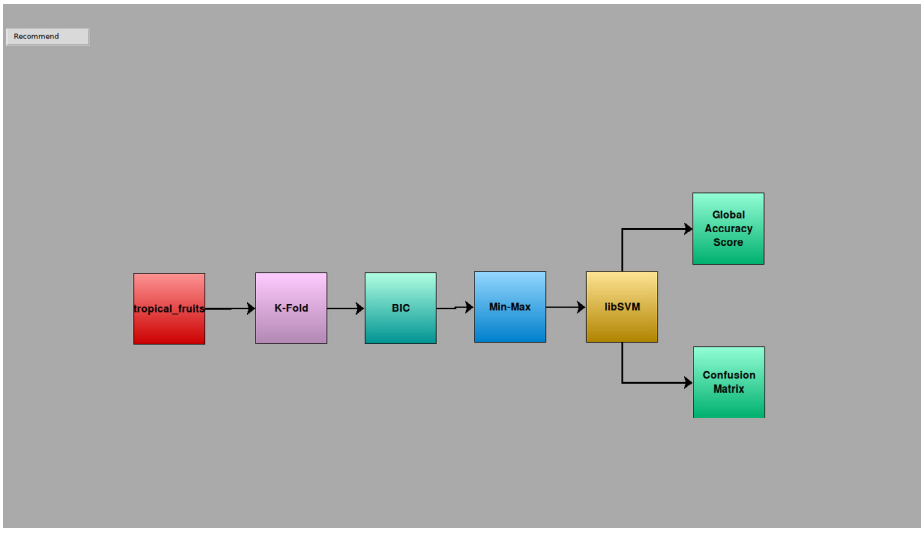


Fig. 10. "Recommend" button must be used to begin the Recommendation module of the framework. The Recommendation module will use the experiment under configuration (the workflow in the center of the screen) as a query for searching for existing similar experiments.

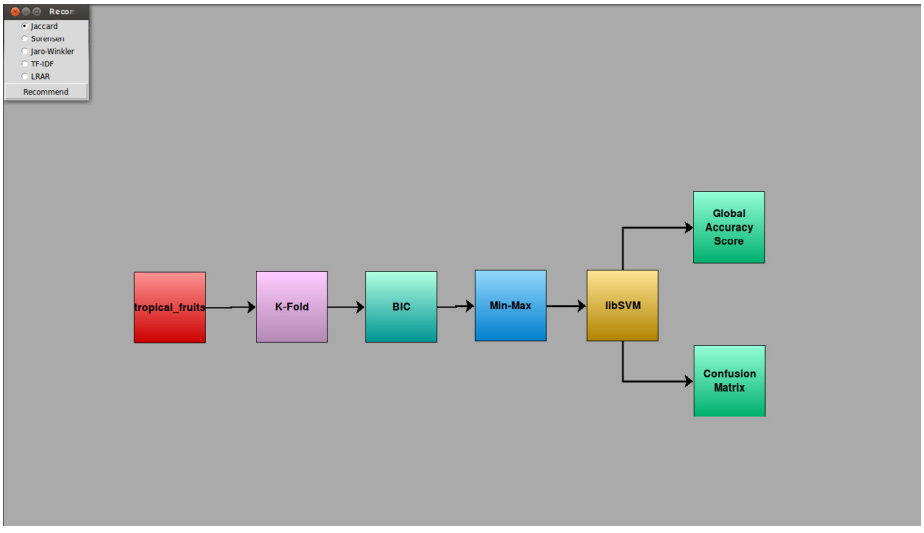


Fig. 11. "Recommender" window showing the plugin options that are implemented in the Recommendation module. Selecting one of these plugins, each previous workflow designed in the framework are ranked according to the selected recommendation method.

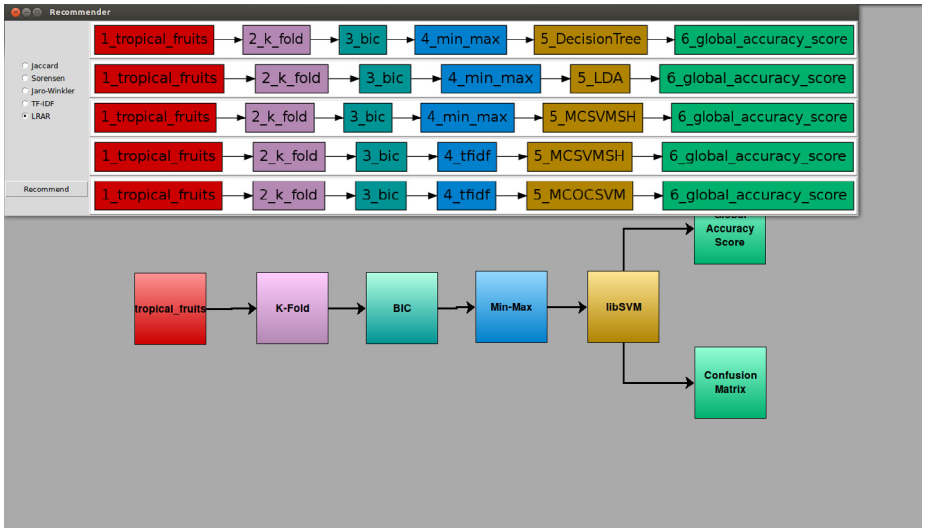


Fig. 12. The five most similar existing experiments ranked according to the recommendation method selected on the left panel. In this example, we can see that the LRAR method recommends very similar experiments, only changing the normalizer and the classifier.

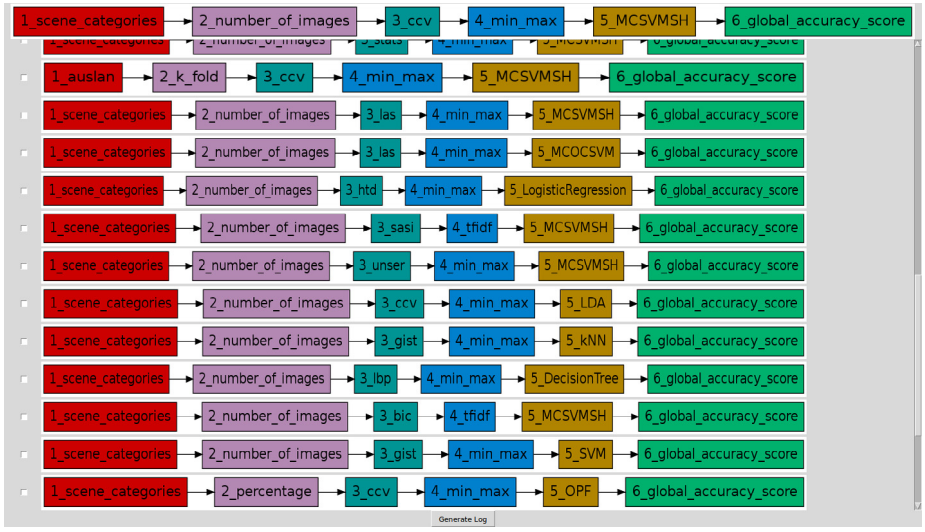


Fig. 13. Interface shown to the specialists. On the top of the window we have the query workflow, and below it, we have a list of workflows to be labeled. On the bottom, there is a button to generate a file with the labels given by the specialists.

precision. To use the LRAR method, we selected the best confidence and support ($\min_\theta = 0.1$ and $\min_\sigma = 0.1$).

With the best result of the LRAR method, we can compare the precision ($P@5$, $P@10$, and $P@20$) of each query for the five implemented methods (Jaccard, Sørensen, Jaro–Winkler, TF–IDF-based, and LRAR), grouping them according to the folds. Table 6 shows the result for the precision measure.

We can conclude, by Table 6, that the Learning to Rank using Association Rules (LRAR) has the best performance, with precision $P@5$ of 81%, followed by the Jaro–Winkler similarity measure with $P@5$ of 80%. We noted that the relevant workflows are those that are very similar to the query, specially maintaining the early steps of the machine-learning experiments (database, feature extraction). These early steps are selected as they represent the core of the experiment according to the specialists.

Once we have the precision of all methods, we applied the Student's t-test and the Wilcoxon test to confirm if the results of the methods are significantly different for each other. We compared the methods in pairs, using a confidence of 95%. Fig. 15 shows these results.

The Student's t test in Fig. 15 shows that the two methods with the best performance (LRAR and Jaro–Winkler) are not significantly different from each other, but significantly different from the others. The Wilcoxon test confirms the Student's results, with p-values above 0.05 in the Jaro–Winkler and LRAR comparison.

It is worth to mention that the LRAR method is consistent with the user-generated ground-truth. It recommends relevant workflows. The accuracy measures shown in Table 7 highlight this property of LRAR.

5. Conclusions

Nowadays, we often have to handle large and complex datasets that are difficult to process using some of the existing data analysis tools. To extract knowledge from this data, we usually perform machine-learning experiments. There are several libraries and machine-learning frameworks in the literature, however, they have some flaws, as they are not amenable to further extension with novel methods, and often they do not identify and reuse successful solutions devised in the past. In this work, we addressed

Table 6
Precision of each similarity measure and LRAR for each fold. Jaro–Winkler and LRAR achieved the best result for Precision@5 with 0.80 and 0.81, respectively.

	Jaccard	Jaro–Winkler	Sørensen	TF–IDF-based	LRAR
	Fold 1				
mean_P@5	0.60	0.70	0.60	0.25	**0.75**
mean_P@10	**0.42**	**0.42**	**0.42**	0.25	0.40
mean_P@20	0.23	**0.29**	0.23	0.21	0.20
	Fold 2				
mean_P@5	0.75	**0.90**	0.75	0.50	**0.90**
mean_P@10	**0.60**	0.57	**0.60**	0.43	0.55
mean_P@20	0.44	**0.52**	0.44	0.30	0.28
	Fold 3				
mean_P@5	0.60	**0.85**	0.60	0.20	**0.85**
mean_P@10	0.50	**0.55**	0.50	0.25	0.48
mean_P@20	0.30	**0.42**	0.30	0.25	0.24
	Fold 4				
mean_P@5	0.53	**0.67**	0.53	0.13	**0.67**
mean_P@10	**0.40**	**0.40**	**0.40**	0.27	**0.40**
mean_P@20	0.22	**0.27**	0.22	0.22	0.20
	Fold 5				
mean_P@5	0.60	**0.87**	0.60	0.33	**0.87**
mean_P@10	**0.57**	**0.57**	**0.57**	0.37	0.53
mean_P@20	0.32	**0.38**	0.32	0.27	0.27
	Mean of folds				
mean_P@5	0.62	0.80	0.62	0.28	**0.81**
mean_P@10	**0.50**	**0.50**	**0.50**	0.31	0.47
mean_P@20	0.30	**0.38**	0.30	0.25	0.24

Table 7
Accuracy of the LRAR method. The accuracy for the LRAR method was calculated as the relevance prediction of the LRAR was the same as the specialists ground truth.

Folds	Accuracy
Fold-1	0.88
Fold-2	0.81
Fold-3	0.80
Fold-4	0.86
Fold-5	0.89
Mean	0.85

these two flaws directly by providing a plugin-based framework capable of recommending previous solutions.

We have proposed a workflow-based framework for designing, deploying, executing, and recommending machine-learning experiments. An important contribution of this work is the implementation of Kuaa, a tool that implements the proposed framework. This tool, as explained in the previous sections, is able to provide a standardized environment for exploratory analysis of machine-learning solutions. Kuaa makes it easy to evaluate different feature descriptors, normalizers, classifiers, fusion approaches in a wide range of tasks involving machine learning. The Kuaa code is freely available to download at GitHub.[9]

Another contribution is the evaluation of similarity measures and a learning-to-rank method in a recommendation setup, in which it makes the recommendation of machine-learning experiments modeled as a sequence of activities. We compared the performance of four similarity measures (Jaccard, Sørensen, Jaro–Winkler, and a TF–IDF-based measure) and the learning-to-rank method using Association Rules. Among the similarity measures, Jaro–Winkler had the best performance, with *P*@5 of 80%, and the LRAR method obtained *P*@5 of 81%, that is, 80% of the recommended workflow experiments were marked relevant according to machine-learning specialists. With these precision values, we

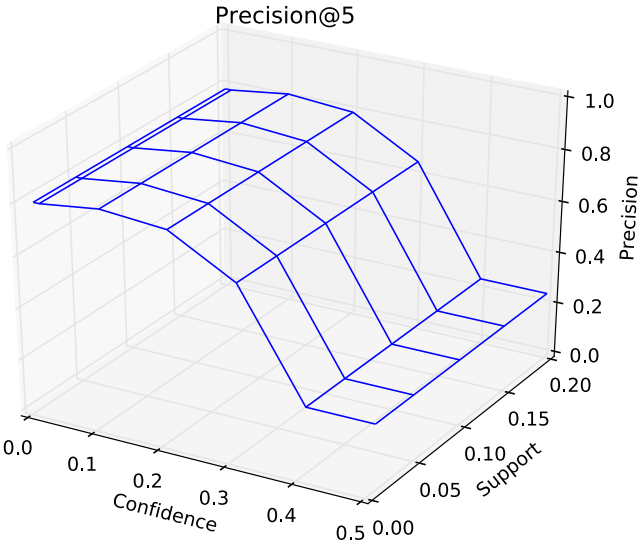


Fig. 14. Precision of the LRAR method for each combination of confidence and support. The precision is only affected by the confidence threshold, and a high value of precision limits the effectiveness of LRAR.

applied the Student's t test and the Wilcoxon test to confirm that these two methods are not significantly different from each other. A good recommendation can help beginner users, and also experienced ones, to design more effective machine-learning experiments, presenting workflows used previously in the framework that can help new ideas to use different algorithms.

Several research venues can be addressed in future work. We propose the study of other *learning-to-rank* techniques, such as RankSVM [40,95], AdaRank [96] or RankBoost [97]. Another strategy concerns the use of rank aggregation approaches to combine ranked lists defined by different similarity functions [98]. We also plan to incorporate other plugins so that the tool can be used

[9] Kuaa code: https://github.com/rafaelwerneck/kuaa.

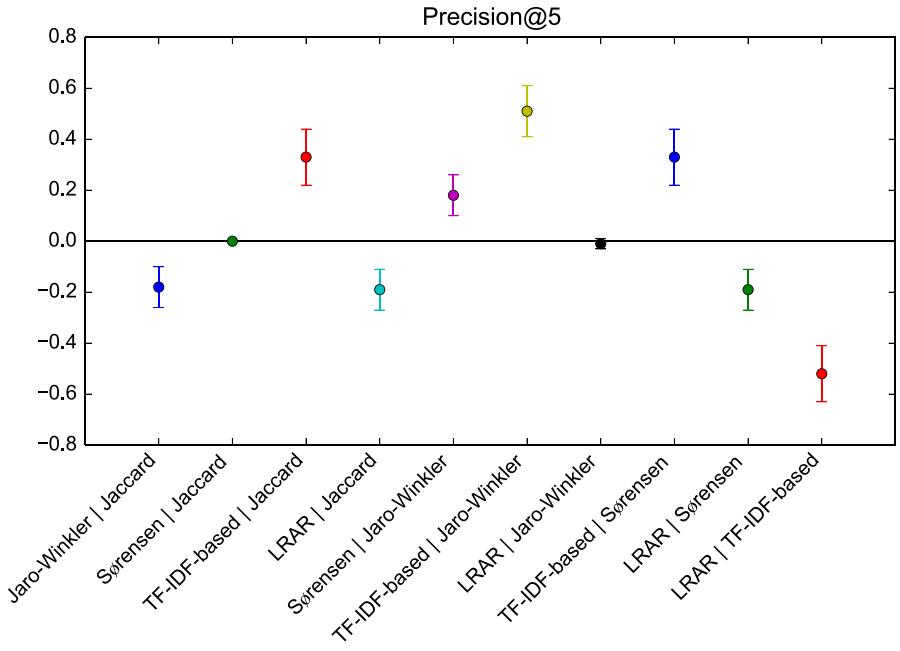


Fig. 15. Student's t-test for the Precision@5, comparing all recommendation methods. Dots below the horizontal line indicate that the first method in the corresponding pair of the x-axis is better. Dots above the line indicate the opposite. If the error bar touches the horizontal line, there is no statistical difference between the two methods being compared. We also execute a non-parametric Wilcoxon test and obtained similar results.

for designing and executing more complex machine learning experiments. We also would like to incorporate the possibility to run experiments for deep learning, like for instance, providing a visual interface for configuring experiments of Caffe or TensorFlow. In special, we would like to incorporate meta-recognition methods [99], expanding the range of tasks to be performed in the Kuaa framework.

Acknowledgments

Part of the results presented in this paper were obtained through the project "Pattern recognition and classification by feature engineering, *-fusion, open-set recognition, and meta-recognition", sponsored by Samsung Eletrônica da Amazônia Ltda., in the framework of law No. 8248/91. The authors also thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (grants #141584/2016-5 and #307560/2016-3), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) through the DeepEyes project, and Microsoft Research for the financial support. Authors are also grateful to FAPESP (grants #2014/12236-1, #2016/18429-1, and #2015/19222-9), and the FAPESP-Microsoft Virtual Institute (grants #2013/50169-1 and #2013/50155-0).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.future.2017.06.013.

References

[1] F. Provost, T. Fawcett, Data science and its relationship to big data and data-driven decision making, Big Data 1 (1) (2013) 51–59.
[2] J. Wainer, M. Weske, G. Vossen, C.B. Medeiros, Scientific workflow systems. in: Proceedings of the NSF Workshop on Workflow and Process Automation Information Systems, 1996.
[3] M. Mattoso, C. Werner, G.H. Travassos, V. Braganholo, E.S. Ogasawara, D. de Oliveira, S.M.S. da Cruz, W. Martinho, L. Murta, Towards supporting the life cycle of large scale scientific experiments, IJBPIM 5 (1) (2010) 79–92 URL http://dx.doi.org/10.1504/IJBPIM.2010.033176.
[4] O.A. Penatti, R.de O. Werneck, W.R. de Almeida, B.V. Stein, D.V. Pazinato, P.R.M. Júnior, R.da S. Torres, A. Rocha, Mid-level image representations for real-time heart view plane classification of echocardiograms, Comput. Biol. Med. 66 (2015) 66–81. http://dx.doi.org/10.1016/j.compbiomed.2015.08.004. URL http://www.sciencedirect.com/science/article/pii/S0010482515002814.
[5] A. Rocha, D.C. Hauagge, J. Wainer, S. Goldenstein, Automatic fruit and vegetable classification from images, Comput. Electron. Agric. 70 (1) (2010) 96–104.
[6] F.A. Faria, J.A. dos Santos, A. Rocha, R. da Silva Torres, Automatic classifier fusion for produce recognition, in: Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images, 2012, pp. 252–259.
[7] P. Langley, H.A. Simon, Applications of machine learning and rule induction, Commun. ACM 38 (11) (1995) 54–64.
[8] T. Natschläger, F. Kossak, M. Drobics, Extracting knowledge and computable models from data-needs, expectations, and experience, in: Proceedings of the IEEE International Conference on Fuzzy Systems, Vol. 1, IEEE, 2004, pp. 493–498.
[9] G.J. Williams, Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery, Use R!, Springer, 2011.
[10] Y. Yue, T. Finley, F. Radlinski, T. Joachims, A support vector method for optimizing average precision, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 271–278.
[11] T. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Trans. Inform. Theory 13 (1) (1967) 21–27 URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1053964.
[12] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, J.M. Hellerstein, Graphlab: A new framework for parallel machine learning, in: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2010, pp. 340–349.
[13] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, J.M. Hellerstein, Distributed graphlab: A framework for machine learning in the cloud, Proc. VLDB Endow. 5 (8) (2012) 716–727.
[14] S. Hido, S. Tokui, S. Oda, Jubatus: An open source platform for distributed online machine learning, in: NIPS Workshop on Big Learning, 2013.
[15] M.A. Gonçalves, E.A. Fox, L.T. Watson, N.A. Kipp, Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries, ACM Trans. Inf. Syst. 22 (2) (2004) 270–312.
[16] W. Hill, L. Stead, M. Rosenstein, G. Furnas, Recommending and evaluating choices in a virtual community of use, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'95, ACM Press/Addison-Wesley Publishing Co., ACM/Addison-Wesley, 1995, pp. 194–201.
[17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: An open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW'94, ACM, New York, NY, USA, 1994, pp. 175–186.

[18] U. Shardanand, P. Maes, Social information filtering: Algorithms for automating word of mouth, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'95, ACM Press/Addison-Wesley Publishing Co., ACM/Addison-Wesley, 1995, pp. 210–217.
[19] Z. Huang, W. Chung, T.-H. Ong, H. Chen, A graph-based recommender system for digital library, in: Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, ACM, ACM, 2002, pp. 65–73.
[20] H.-C. Chen, A.L. Chen, A music recommendation system based on music data grouping and user interests, in: Proceedings of the 10th ACM CIKM International Conference on Information and Knowledge Management, Vol. 1, ACM, 2001, pp. 231–238.
[21] G. Linden, B. Smith, J. York, Amazon Com Recommendations: Item-To-Item Collaborative Filtering, IEEE Internet Comput. 7 (1) (2003) 76–80.
[22] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, J. Riedl, Movielens unplugged: Experiences with an occasionally connected recommender system, in: Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI'03, ACM, ACM, New York, NY, USA, 2003, pp. 263–266.
[23] A. Sapountzi, K.E. Psannis, Social networking data analysis tools & challenges, Future Gener. Comput. Syst. (2016). http://dx.doi.org/10.1016/j.future.2016.10.019. URL http://www.sciencedirect.com/science/article/pii/S0167739X1630423X.
[24] S. Lo, C. Lin, Wmr–a graph-based algorithm for friend recommendation, in: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI'06, IEEE Computer Society, IEEE Computer Society, Washington, DC, USA, 2006, pp. 121–128.
[25] R. Burke, Hybrid recommender systems: Survey and experiments, User Model. User-Adapt. Interact. 12 (4) (2002) 331–370.
[26] A. Abbas, K. Bilal, L. Zhang, S.U. Khan, A cloud based health insurance plan recommendation system: A user centered approach, Future Gener. Comput. Syst. 43–44 (2015) 99–109. http://dx.doi.org/10.1016/j.future.2014.08.010. URL http://www.sciencedirect.com/science/article/pii/S0167739X14001587.
[27] J.K. Tarus, Z. Niu, A. Yousif, A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining, Future Gener. Comput. Syst. 72 (2017) 37–48. http://dx.doi.org/10.1016/j.future.2017.02.049. URL http://www.sciencedirect.com/science/article/pii/S0167739X17303254.
[28] D. Billsus, C.A. Brunk, C. Evans, B. Gladish, M. Pazzani, Adaptive interfaces for ubiquitous web access, Commun. ACM 45 (5) (2002) 34–38.
[29] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kherees, R. Martinez, W. Nzoukou, A survey paper on recommender systems, Comput. Res. Repository (2010).
[30] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (0) (2013) 109–132.
[31] L.A. Seffino, C. Bauzer Medeiros, J.V. Rocha, B. Yi, WOODSS - a spatial decision support system based on workflows, Decis. Support Syst. 27 (1–2) (1999) 105–123.
[32] C.B. Medeiros, J. de Jesús Pérez Alcázar, L.A. Digiampietri, G.Z. Pastorello, Jr., A. Santanchè, R. da Silva Torres, E.R.M. Madeira, E. Bacarin, Woodss and the web: Annotating and reusing scientific workflows, ACM SIGMOD Rec. 34 (3) (2005) 18–23.
[33] D.S. Kaster, C.B. Medeiros, H.V. Rocha, Supporting modeling and problem solving from precedent experiences: the role of workflows and case-based reasoning, Environ. Model. Softw. 20 (6) (2005) 689–704.
[34] C.K. Riesbeck, R.C. Schank, Inside Case-Based Reasoning, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1989.
[35] R. Conforti, M. de Leoni, M.L. Rosa, W.M. van der Aalst, A.H. ter Hofstede, A recommendation system for predicting risks across multiple business process instances, Decis. Support Syst. 69 (2015) 1–19. http://dx.doi.org/10.1016/j.dss.2014.10.006. URL http://www.sciencedirect.com/science/article/pii/S0167923614002516.
[36] C.S. Chong, T. Zhang, K.K. Lee, G.G. Hung, Terence, B.S. Lee, Collaborative analytics with genetic programming for workflow recommendation, in: 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 657–662. http://dx.doi.org/10.1109/SMC.2013.117.
[37] Z. Zhou, Z. Cheng, L.J. Zhang, W. Gaaloul, K. Ning, Scientific workflow clustering and recommendation leveraging layer hierarchical analysis, IEEE Trans. Serv. Comput. PP (99) (2016) 1–14. http://dx.doi.org/10.1109/TSC.2016.2542805.
[38] J. Zhang, C. Lee, S. Xiao, P. Votava, T.J. Lee, R. Nemani, I. Foster, 2014. A community-driven workflow recommendations and reuse infrastructure, in: 2014 IEEE 8th International Symposium on Service Oriented System Engineering, pp. 162–172. http://dx.doi.org/10.1109/SOSE.2014.23.
[39] R.A. Baeza-Yates, B.A. Ribeiro-Neto, Modern Information Retrieval - the Concepts and Technology behind Search, second ed, Pearson Education Ltd., Harlow, England, 2010.
[40] R. Herbrich, T. Graepel, K. Obermayer, Large Margin rank boundaries for ordinal regression, in: Advances in Large-Margin Classifiers, MIT Press, 2000, pp. 115–132 (Chapter 7).
[41] C.J.C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G.N. Hullender, Learning to rank using gradient descent, in: Proceedings of the 22nd International Conference on Machine Learning, in: ACM International Conference Proceeding Series, vol. 119, ACM, New York, NY, USA, 2005, pp. 89–96.
[42] W. Fan, M.D. Gordon, P. Pathak, Genetic programming-based discovery of ranking functions for effective web search, J. Manage. Inf. Syst. 21 (4) (2005) 37–56.
[43] A. Veloso, H.M. de Almeida, M.A. Gonçalves, W.M. Jr, Learning to rank at query-time using association rules, in: S.-H. Myaeng, D.W. Oard, F. Sebastiani, T.-S. Chua, M.-K. Leong (Eds.), Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, ACM, 2008, pp. 267–274.
[44] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, in: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Vol. 22, no. 2, 1993, pp. 207–216.
[45] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM Spec. Interest Group Knowl. Discov. Data Min. Explor. Newsl. 11 (1) (2009) 10–18.
[46] O.A.B. Penatti, R.d.S. Torres, Eva - an evaluation tool for comparing descriptors in content-based image retrieval tasks, in: International Conference on Multimedia Information Retrieval, 2010, pp. 413–416.
[47] D.C.G. Pedronette, Uma plataforma de serviços de recomendação para bibliotecas digitais, Master's thesis, Universidade Estadual de Campinas, (Mar. 2008).
[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[49] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: International Joint Conference on Artificial Intelligence, ACM, New York, NY, USA, 1995, pp. 1137–1143.
[50] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, in: Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, pp. 762–768.
[51] R.O. Stehling, M.A. Nascimento, A.X. Falcão, A compact and efficient image retrieval approach based on border/interior pixel classification, in: Proceedings of the 11th ACM CIKM International Conference on Information and Knowledge Management, CIKM'02, ACM, 2002, pp. 102–109.
[52] V. Kovalev, S. Volmer, Color co-occurrence descriptors for querying-by-example, in: Proceedings of the International Conference on Multimedia Modeling, 1998, pp. 32–38.
[53] G. Pass, R. Zabih, J. Miller, Comparing images using color coherence vectors, in: Proceedings of the 4th ACM International Conference on Multimedia, MULTIMEDIA'96, ACM, New York, NY, USA, 1996, pp. 65–73.
[54] S. Chatzichristofis, Y. Boutalis, Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval, in: Computer Vision Systems, Springer, 2008, pp. 312–322.
[55] M.J. Swain, D.H. Ballard, Color indexing, Int. J. Comput. Vis. 7 (1) (1991) 11–32.
[56] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (3) (2001) 145–175.
[57] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.
[58] P. Wu, B.S. Manjunath, S. Newsam, H. Shin, A texture descriptor for browsing and similarity retrieval, Signal Process., Image Commun. 16 (1) (2000) 33–43.
[59] F. Mahmoudi, J. Shanbehzadeh, A.-M. Eftekhari-Moghadam, H. Soltanian-Zadeh, Image retrieval based on shape similarity by edge orientation autocorrelogram, Pattern Recognit. 36 (8) (2003) 1725–1736.
[60] A. Williams, P. Yoon, Content-based image retrieval using joint correlograms, Multimedia Tools Appl. 34 (2) (2007) 239–248.
[61] B. Tao, B.W. Dickinson, Texture recognition and image retrieval using gradient indexing, J. Vis. Commun. Image Represent. 11 (3) (2000) 327–342.
[62] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[63] J.A. Montoya-Zegarra, N.J. Leite, R.d.S. Torres, Rotation-invariant and scale-invariant steerable pyramid decomposition for texture image retrieval, in: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing, 2007, pp. 121–128.
[64] C.-B. Huang, Q. Liu, An orientation independent texture descriptor for image retrieval, in: Proceedings of the International Conference on Communications, Circuits and Systems, 2007, pp. 772–776.
[65] A. Çarkacıoğlu, F. Yarman-Vural, Sasi: a new texture descriptor for content based image retrieval, in: Proceedings of the International Conference on Image Processing, Vol. 2, 2001, pp. 137–140.
[66] A. Çarkacıoğlu, F. Yarman-Vural, Sasi: a generic texture descriptor for image retrieval, Pattern Recognit. 36 (11) (2003) 2615–2633.
[67] D.V. Pazinato, B.V. Stein, W.R. de Almeida, R.de O. Werneck, P.R.M. Júnior, O.A.B. Penatti, R.d.S. Torres, F.H. Menezes, A. Rocha, Pixel-level tissue classification for ultrasound images, IEEE J. Biomed. Health Inform. 20 (1) (2016) 256–267. http://dx.doi.org/10.1109/JBHI.2014.2386796.
[68] M. Unser, Sum and difference histograms for texture classification, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (1) (1986) 118–125.

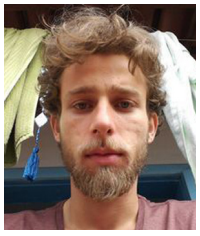
[69] S. Aksoy, R.M. Haralick, Feature normalization and likelihood-based similarity measures for image retrieval, Pattern Recognit. Lett. 22 (5) (2001) 563–582.
[70] G. Salton, A. Wong, C.-S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.
[71] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
[72] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, Wadsworth, 1984.
[73] N.S. Altman, An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, Amer. Statist. 46 (3) (1992) 175–185.
[74] R.A. Fisher, The statistical utilization of multiple measurements, Ann. Eugenics 8 (4) (1938) 376–386.
[75] K. Fukunaga, Introduction To Statistical Pattern Recognition, second ed, Academic Press Professional, Inc., San Diego, CA, USA, 1990.
[76] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27.
[77] C.R. Boyd, M.A. Tolson, W.S. Copes, Evaluating trauma care: the triss method, J. Trauma-Inj. Infect. Crit. Care 27 (4) (1987) 370–378.
[78] J.P. Papa, A.X. Falcão, P.A. Miranda, C.T. Suzuki, N.D. Mascarenhas, Design of robust pattern classifiers based on optimum-path forests, in: Mathematical Morphology and its Applications to Signal and Image Processing, ISMM, MCT/INPE, 2007, pp. 337–348.
[79] F.de O. Costa, M. Eckmann, W.J. Scheirer, A. Rocha, Open set source camera attribution, in: Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images, 2012, pp. 71–78.
[80] F.de O. Costa, E. Silva, M. Eckmann, W.J. Scheirer, A. Rocha, Open set source camera attribution and device linking, Pattern Recognit. Lett. 39 (0) (2014) 92–101.
[81] A.A. Ross, R. Govindarajan, Feature level fusion of hand and face biometrics, in: Defense and Security, Int. Soc. Opt. Photonics (2005) 196–204.
[82] L.I. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2) (2002) 281–286.
[83] S.V. Stehman, Selecting and interpreting measures of thematic classification accuracy, Remote Sens. Environ. 62 (1) (1997) 77–89.
[84] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. 20 (1) (1960) 37–46.
[85] K.A. Spackman, Signal detection theory: Valuable tools for evaluating inductive learning, in: Proceedings of the 6th International Workshop on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989, pp. 160–163.
[86] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.
[87] P.R. Mendes Júnior, R.M. de Souza, R.d.O. Werneck, B.V. Stein, D.V. Pazinato, W.R. de Almeida, O.A.B. Penatti, R.d.S. Torres, A. Rocha, Nearest neighbors distance ratio open-set classifier, Mach. Learn. 106 (3) (2017) 359–386 URL http://dx.doi.org/10.1007/s10994-016-5610-8.
[88] W.J. Scheirer, A. de Rezende Rocha, A. Sapkota, T.E. Boult, Toward open set recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (7) (2013) 1757–1772. http://dx.doi.org/10.1109/TPAMI.2012.256.
[89] P. Jaccard, Étude comparative de la distribution florale dans une portion des Alpes et des Jura, Bull. Soc. Vaud. Sci. Nat. 37 (1901) 547–579.
[90] T. Sørensen, A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Ist Application to Analyses of the Vegetation on Danish Commons, Det Kongelige Danske Videnskabernes Selskab, Munksgaard 1948.
[91] M.A. Jaro, Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida, J. Amer. Statist. Assoc. 84 (406) (1989) 414–420.
[92] M.A. Jaro, Probabilistic linkage of large public health data files, Stat. Med. 14 (5–7) (1995) 491–498.
[93] W.E. Winkler, String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage, in: Proceedings of the Section on Survey Research, 1990, pp. 354–359.
[94] F.A. Faria, J.A. dos Santos, A. Rocha, R. da Silva Torres, Automatic classifier fusion for produce recognition, in: 25th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2012, Ouro Preto, Brazil, August 22–25 2012: pp. 252–259, 2012. URL http://dx.doi.org/10.1109/SIBGRAPI.2012.42.
[95] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2002, pp. 133–142.
[96] J. Xu, H. Li, Adarank: a boosting algorithm for information retrieval in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 391–398.
[97] Y. Freund, R. Iyer, R.E. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences, J. Mach. Learn. Res. 4 (2003) 933–969.
[98] D.C.G. Pedronette, R.da S. Torres, Image re-ranking and rank aggregation based on similarity of ranked lists, Pattern Recognit. 46 (8) (2013) 2350–2360. http://dx.doi.org/10.1016/j.patcog.2013.01.004. URL http://www.sciencedirect.com/science/article/pii/S003132031300023X.
[99] W.J. Scheirer, A. Rocha, R.J. Michaels, T.E. Boult, Meta-recognition: The theory and practice of recognition score analysis, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1689–1695.

Rafael de Oliveira Werneck received the B.Sc. degree in computer science from the Universidade Federal de Juiz de Fora (UFJF), Juiz de Fora, Brazil, in 2011, and the M.Sc. degree in computer science from the University of Campinas (Unicamp), Campinas, Brazil, in 2014. Currently, he is a Ph.D. student at University of Campinas. His research interests include remote sensing, machine learning, and pattern recognition.

Waldir Rodrigues de Almeida received his bachelor's degree in Computer Science from the University of Campinas in 2015. As an undergraduate student, he gained experience in Machine Learning and Computer Vision through two internships. He was also a visiting student at the Technical University of Munich for one year. He is currently working on his master's thesis on automatic detection of presentation attacks to mobile face recognition systems.

Bernardo Vecchia Stein has a B.Sc. in Computer Science from State University of Campinas, and holds an interest and appreciation for Machine Learning applications.

Daniel Vatanabe Pazinato graduated from UNICAMP in July of 2017. During 2012 and 2013 worked in Pattern Recognition project with Samsung. The main goal was to classifying ultrasound medical images. Daniel helped to publish 3 papers in this project.



Pedro Ribeiro Mendes Júnior is a Ph.D. candidate at University of Campinas (UNICAMP), working mainly with recognition in open-set scenarios, developing methods that can be applied to general open-set problems. He obtained the master's degree through UNICAMP in open-set recognition. Through Federal University of Ouro Preto (UFOP) he obtained a bachelor's degree in Computer Science. During the undergraduate, he worked mainly with digital image processing. Currently, he is working at University of Colorado Colorado Springs (UCCS) as a research assistant investigating open-set recognition.

Otávio Augusto Bizetto Penatti is a researcher at Samsung Research Institute in Brazil. He received his Ph.D. and M.Sc. degrees from University of Campinas in 2012 and 2009, respectively. His research interests include computer vision, pattern recognition, machine learning, and multimedia geocoding.

Anderson Rocha He is an associate professor at the Institute of Computing, University of Campinas. His main interests include Reasoning for Complex Data, Digital Forensics and Machine Intelligence. He is an IEEE Senior Member, an elected affiliate member of the Brazilian Academy of Sciences (ABC) and of the IEEE Information Forensics and Security Technical Committee. He is a Microsoft Research Faculty Fellow, a Google Research Faculty Fellow and a Tan Chin Tuan Fellow. Finally, he is currently the principal investigator of a number of research projects in partnership with public funding agencies and multinational companies having already licensed several patents.

Ricardo da Silva Torres is Full Professor of computer science at the University of Campinas (UNICAMP). Dr. Torres was director of the Institute of Computing, University of Campinas from 2013 to 2017. Dr. Torres received a B.Sc. in Computer Engineering from University of Campinas, Brazil, in 2000 and his Ph.D. degree in Computer Science at the same university in 2004. Dr. Torres is co-founder and member of the RECOD lab, where he has been developing multidisciplinary e-Science research involving Multimedia Analysis, Multimedia Image Retrieval, Databases, Digital Libraries, and Geographic Information Systems. Dr. Torres is author/co-author of more than 100 articles in refereed journal and conferences and serves as PC member for several international and national conferences.