National University of Singapore
School of Computing
CS3243 Introduction to AI

Project 1: A Search Problem in Automated Machine Learning

Issued:
February 6, 2021

Due: Plan – February 13, 2021
Report – March 3, 2021

Overview

Many people have the misconception that Machine Learning (ML) and Artificial Intelligence (AI) are synonymous. However, AI actual encompasses ML, and additionally, includes many other topics.

In this project, you will explore the most common AI problem: Search. However, the application of Search will be set in ML, and more specifically, within the scope of Automated Machine Learning (AutoML).

There is no coding involved in this project. However, the work done here will directly relate to Project 2, where you will apply the work done in this project, extend upon it, and then pit your constructed agent against those developed by your peers.

You have two deliverables as noted above. The first corresponds to a Project Plan, while the second corresponds to a Report, where you will formalise the given search problem and begin to design an agent that will solve it.

The remainder of this Project 1 description includes:

- A very brief introduction to AutoML and the problem of Algorithm Selection, as well as the specification of the search problem you are to tackle.
- The project objectives, which includes a specification of the tasks you will have to complete.
- The rubrics that will be used to assess your submissions.
- The submission details.

This project is worth 10% of your final grade.

Introduction: AutoML and the Algorithm Selection Problem for Classification

AutoML (Feurer et al., 2015) focuses primarily on the algorithm selection problem (Smith-Miles, 2009), which is part of the larger domain of meta-learning (Lemke, Budka & Gabrys, 2015; Vilalta & Drissi, 2002).

The algorithm selection problem, succinctly described, corresponds to the construction of a model, $\boldsymbol{F}$, that maps various dataset characteristics (i.e., meta-features), $\boldsymbol{X}$, to algorithm choices (i.e., meta-labels), $\boldsymbol{Y}$ – i.e., in solving the algorithm selection problem, one seeks to determine some $\boldsymbol{F}$, such that, with high accuracy, $\boldsymbol{F}(\boldsymbol{X}) = \boldsymbol{Y}$.

One issue with the above is that a standard corpus of datasets that is tailored to the algorithm selection problem is not readily available. More specifically, repositories such as the UCI repository (Dua & Graff, 2019) and the ImageNet repository (Deng et al., 2009) cannot be taken as representative samples of the entire population of all datasets. In fact, it is unlikely that such a sample can be easily produced, and without such a sample, we may not be able to properly construct a classifier (i.e., agent) that will be able to adequately solve the algorithm selection problem.

The thesis central to the work you are to undertake is thus as follows:

Given that we may never be able to know the overarching distribution that governs all datasets that may be encountered, the next best option is assume a uniform distribution. More specifically, given that we wish to

*perform algorithm selection over a set of classification algorithms, **A**, then we may define a representative sample, **S***, as one that has good and uniform coverage over the expertise space of **A**, **E**.*

At this point, it should be noted that solving the algorithm selection problem is not the focus of this project. Instead, the search problem that you will be tackling, is to generate the representative sample of datasets, **S***. From this point onwards, we will denote this as ***the search problem***.

To comprehend the above, let us review some of the concepts mentioned.

- A dataset in this context corresponds to a classification dataset – refer to Ler (2009) for a more detailed description. For the purposes of this project, consider all such datasets to be binary datasets (i.e., datasets with exactly two class values).

- A classification algorithm is a specific kind of supervised learning algorithm that, when given a dataset, is able to output a classification model (i.e., a classifier) – refer to Ler (2009) for a more detailed description.

- The representative sample, ***S***, refers to a set of classification datasets that may eventually be used to generate the algorithm selection model (i.e., the algorithm selection classifier) – refer to Giraud-Carrier (2008) for a more detailed description.

- The space in question, ***E***, i.e., the expertise space of ***A***, corresponds to an *m*-dimensional space ($m = |\boldsymbol{A}|$), in which, each dimension corresponds to the performance of an algorithm, $\boldsymbol{a}_i \in \boldsymbol{A}$, and as such, the space is to be populated by the *n* datasets in ***S***. Note that performance here refers to generalisation performance – refer to Ler (2009) for a more details on the evaluation of classification algorithms and generalisation performance.

- For the purposes of this project, you may assume that ***A*** comprises 3 basic machine learning algorithms:
  - *k*-Nearest Neighbours
  - Decision Trees
  - Naïve Bayes

  The specified implementations of the above (looking ahead to Project 2), will be based on the implementations within the scikit-learn Python package (Pedregosa et al., 2011).

  Details on these algorithms maybe me found within the scikit-learn user guide at:
  https://scikit-learn.org/stable/user_guide.html

- ***S**** should have a good and uniform coverage over ***E***, which implies that the *n* datasets (where $n = |\boldsymbol{S}^*|$) in ***S**** should be as uniformly distributed as possible across the entirety of ***E***.

- In order to generate ***S***, you may assume that you will be given a set of datasets ***S*** (containing only binary datasets), which you will mutate in order to derive a ***S***. This means that for any dataset $\boldsymbol{s}_j \in \boldsymbol{S}$, you may change its composition by adding and/or removing instances from $\boldsymbol{s}_j$. Essentially, this means that you may change the values/class of each instance in $\boldsymbol{s}_j$ as you see fit. However, do note that when mutate $\boldsymbol{s}_j$ so, you should ensure that the following constraints are satisfied:
  - The size of the dataset should remain the same – i.e., if $\boldsymbol{s}_j$ contains 100 instances, then a new mutation of $\boldsymbol{s}_j$ should also contain 100 instances.
  - You must not make the dataset any more imbalanced than it already is – i.e., if the dataset had 30 instances of class 1, and 70 instances of class 2, the mutated dataset must have a class 1 to class 2 ratio that falls between 3:7 and 1:1 – i.e., it cannot fall to 1:99 or even 29:71, etc.
  - The number of unique instances in the dataset should not change – i.e., you cannot simply replace some instances with duplicated copies of other instances.
  - Any newly created instances should have continuous values that fall within the ranges of the original dataset – i.e., you cannot pick a new variable value for an instance that is less than the minimum value for that variable or greater than the maximum value for that variable. Likewise, newly created instances should have discrete values that match values for that variable that already exist – i.e., you cannot pick new discrete values for any discrete variable.
  - You must ensure that no two instances in the newly created dataset have the same variable values, but have different classes (i.e., assume no noise).

If the description above leaves you with more questions, then you would do well to review the following additional readings:

- ***Ler, D. (2008). Fundamentals of Algorithm Selection for Classification. Technical Report.***
  This paper provides an in-depth introduction to machine learning, inductive bias, and the evaluation of classification algorithms as they are applied to classification problems (i.e., to classification datasets). As a part of AI, these ML fundamentals give you further insight into the contrast between Search problems and Supervised Learning problems. It also serves as a primer that will allow you better appreciate the tasks in this project.

- ***Giraud-Carrier, C. (2008). Metalearning – A Tutorial. In Tutorial at the 7th international conference on machine learning and applications.***
  This paper provides an introduction to meta-learning, which includes the algorithm selection problem. It does well to convey the main problems in the area of meta-learning, and also reviews some of the earlier work that was done, though, in terms of algorithm selection, the focus is primarily on meta-features (i.e., the representations that can be used in models that solve algorithm selection).

- ***Chen, H., Liu, Y., Ahuja, J. K., & Ler, D. (2020). A Distance-Weighted Class-Homogeneous Neighbourhood Ratio for Algorithm Selection. In Asian Conference on Machine Learning, 1-16.***
  A recent paper that begins to broach the problem that is the focus of this project. It also succinctly reviews the algorithm selection problem and specifies the concepts used in this project description.

The following are the references cited in this section:

- ***Chen, H., Liu, Y., Ahuja, J. K., & Ler, D. (2020). A Distance-Weighted Class-Homogeneous Neighbourhood Ratio for Algorithm Selection. In Asian Conference on Machine Learning, 1-16.***

- ***Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, 248-255.***

- ***Dua, D. & Graff, C. (2019). UCI Machine Learning Repository.***

- ***Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In Proceedings of Advances in Neural Information Processing Systems 28, 2962-2970.***

- ***Giraud-Carrier, C. (2008). Metalearning – A Tutorial. In Tutorial at the 7th international conference on machine learning and applications.***

- ***Lemke, C., Budka, M., & Gabrys, B. (2015). Metalearning: a survey of trends and technologies. Artificial Intelligence Review, 44(1), 117-130.***

- ***Ler, D. (2008). Fundamentals of Algorithm Selection for Classification. Technical Report.***

- ***Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.***

- ***Smith-Miles, K. A. (2009). Cross-disciplinary perspectives on meta-learning for algorithm selection. ACM Computing Surveys, 41(1), article 6.***

- ***Vilalta, R., & Drissi, Y. (2002). A perspective view and survey of meta-learning. Artificial Intelligence Review, 18(2), 77-95.***

Some of the above papers (notably, the ones you should read) are readily available at:
LumiNUS > CS3243 > Files > Projects > Project 1 > References

Project Objectives

There are several objectives that you are to achieve within this project. They are as follows:

1. Define a clear project plan, which includes:
    a. List of team members
    b. List of Project sub-tasks
    c. The project timeline specified using a Gantt Chart
    d. A description of the communication protocols to be used by the team members of this project group

    Do note that the task of defining the above should not be included in this project plan.

2. Define the performance measure for this Search problem. Justify your specification.

3. Define the generation of ***S**** (as defined in the previous section) as a search problem. You must clearly specify the following:
    a. State Representation (and the Initial State)
    b. Actions
    c. Cost Function
    d. Transition Model
    e. Goal Function

    Describe the issues, if any, with your definition, and discussion what steps may be taken to resolve them.

4. Discussion which search algorithm you will adopt to solve the problem, justifying your choice.

The project plan (Objective 1) must not exceed 1 page. ***This is due by February 13, 2021, 2359 hrs.***

Your report (Objectives 2-4) must not exceed 4 pages. ***This is due by March 3, 2021, 2359 hrs.***

Project Rubrics

Objective 1 [1 mark]	0	0.5	1
	Very poorly defined. The plan cannot be executed simply by looking at it.	The plan can largely be followed, but there are some items missing (e,g., standard communication channels, some tasks missing, etc.)	Well defined and easy to follow.

Objective 2 [3 marks]	0	1	2	3
	No correlation to the problem objective.	While somewhat correlated to the problem objective, there are critical flaws	Well correlated to the problem objective. However, some minor issues are present.	Perfectly correlated to the problem objective as defined in the project description.

Objective 3a [2 marks]	0	1	2
	State representation is inappropriate.	State representation captures most of the necessary information.	State representation captures all necessary information.

Objective 3b [6 marks]	0	1-2	3-4	5-6
	Actions are inappropriate.	Actions only randomly mutate datasets and do not satisfy all the criteria for mutation.	Actions defined have some link to how the dataset may be positioned in the expertise space. However, still largely random. The actions satisfy all criteria.	Actions defined have a clear link to how the dataset may be positioned in the expertise space. The branching factor is reasonable.

Objective 3c [1 mark]	0	0.5	1
	Cost function is ill-defined and/or inappropriate.	Cost function is an adequate approximation of the costs associated with all actions. Or else, cost function accurately approximate action costs, but computing this is inefficient.	Cost function accurately approximates the costs associated with all actions at acceptable computational cost.

Objective 3d [1 mark]	0	0.5	1
	States ill-defined (i.e., 0 awarded for Objective 1), ***or*** no clear transition model on specifying the new state given an action that is taken.	States are, at least, reasonably defined (i.e., greater than 0 awarded for Objective 1), ***and*** there is a transition model defined, but some elements of the action taken are not fully captured.	States are, at least, reasonably defined (i.e., greater than 0 awarded for Objective 1), ***and*** the transition model defined fully captures the action taken

Objective 3e [2 mark]	0	1	2
	Goal function is inappropriate.	Goal function only captures some of the problem objectives; it is incomplete.	Goal function captures all key elements of the problem objective.

Objective 4 [4 marks]	0	1-2	3	4
	No algorithm specified, or arbitrary choice without justification.	An algorithm is chosen. However, the justification is flawed and/or weak.	The chosen algorithms is reasonably justified. However, there are minor flaws in the justification.	The chosen algorithm is well justified.

This project is marked out of 20. Recall that it is worth 10% of the overall weight for this module.

Submission Instructions

- Please make sure that your submissions are TYPE-WRITTEN.
- Each submission must indicate your group number at the top left corner of the first page.
- For the Project Plan, submit a single PDF to:
  LumiNUS > Files > Projects > Project 1 > Project Plan
- For the Report, submit a single PDF to:
  LumiNUS > Files > Projects > Project 1 > Report
- ***Late submissions will not be accepted without a very good reason.***