AI Folk: Sharing Machine Learning Models in a Multi-Agent Community

Andrei Olaru, Alexandru Sorici, Mihai Nan, and David-Traian Iancu

[andrei.olaru@upb.ro]

AI-MAS Group
National University of Science and Technology POLITEHNICA Bucharest

26.06.2024

- Context
- Related Work
- AI Folk
- Experiments
- Conclusions

AI Folk: Sharing Machine Learning Models in a Multi-Agent Community

overview

Problem Context

An autonomous car travelling through a city enters a university campus, where the streets are shared between cars and pedestrians. The car encounters a large number of pedestrians walking in the streets and the current model for controlling the car stops the car abruptly, very often. Other autonomous cars in the campus use specialized models for navigating the campus. The agent contacts other agents in the local community and transfer a campus-specific model which it is able to use so that it will run slower, closer to pedestrians, and more smoothly.

Problem Context

- there is already a large number of ML models and methods, pre-trained on open datasets, which are available to use for a variety of tasks.
- the challenge moves from creating or fine-tuning a model to identifying the appropriate models in terms of tasks, performance, requirements, and situation.

Problem Context

- there is already a large number of ML models and methods, pre-trained on open datasets, which are available to use for a variety of tasks.
- the challenge moves from creating or fine-tuning a model to identifying the appropriate models in terms of tasks, performance, requirements, and situation.
- an agent using an prediction model for a task must be able to identify the current model is not adequate for the current situation

Problem Context

- there is already a large number of ML models and methods, pre-trained on open datasets, which are available to use for a variety of tasks.
- the challenge moves from creating or fine-tuning a model to identifying the appropriate models in terms of tasks, performance, requirements, and situation.
- an agent using an prediction model for a task must be able to identify the current model is not adequate for the current situation → query its community of agents on a more appropriate model

Problem Context

- there is already a large number of ML models and methods, pre-trained on open datasets, which are available to use for a variety of tasks.

- the challenge moves from creating or fine-tuning a model to identifying the appropriate models in terms of tasks, performance, requirements, and situation.

- an agent using an prediction model for a task must be able to identify the current model is not adequate for the current situation → query its community of agents on a more appropriate model → switch to the new model for the task at hand

Problem Context

- there is already a large number of ML models and methods, pre-trained on open datasets, which are available to use for a variety of tasks.

- the challenge moves from creating or fine-tuning a model to identifying the appropriate models in terms of tasks, performance, requirements, and situation.

- an agent using an prediction model for a task must be able to identify the current model is not adequate for the current situation → query its community of agents on a more appropriate model → switch to the new model for the task at hand

[https://aifolk.upb.ro/]

Problem Context

Our goal: Create a methodology and an infrastructure facilitating the process of dynamically selecting prediction models to use when the data distribution in the input shifts away from the original training distribution.

Problem Context

Our goal: Create a methodology and an infrastructure facilitating the process of dynamically selecting prediction models to use when the data distribution in the input shifts away from the original training distribution.

Objectives:

O1. Give agents the ability to use pre-trained machine learning models;

O2. Give agents the ability to detect when the prediction models currently in use become inappropriate for their current situation;

O3. Give agents the ability to create descriptions of their current situation and, conversely, to identify models that match a given description;

O4. Give agents the ability to integrate new prediction models in their model library and to switch between models as needed by the current situation;

Problem Context

Our goal: Create a methodology and an infrastructure facilitating the process of dynamically selecting prediction models to use when the data distribution in the input shifts away from the original training distribution.

Objectives:

O1. Give agents the ability to use pre-trained machine learning models;

O2. Give agents the ability to detect when the prediction models currently in use become inappropriate for their current situation;

O3. Give agents the ability to create descriptions of their current situation and, conversely, to identify models that match a given description;

O4. Give agents the ability to integrate new prediction models in their model library and to switch between models as needed by the current situation;

+ Create a methodology for achieving objectives O1-O4 in any scenario, identifying which aspects are scenario-invariant and which must be developed specifically for each scenario, and how that should be done.

Related Work

- transferring models between entities in a system is a current practice in federated learning, but for the same task, and contributing to the same, or to very similar models
- describing ML models is similar to the initiatives of FAIRnets [Nguyen and Weller, 2019] and ANNETT-O [Klampanos et al., 2019]
- describing driving scenes and other prediction input semantically is done in previous work [Ma et al., 2022, Qian et al., 2024]

The AI Folk Methodology

- General principles
  - formation of experience-sharing communities
  - identification of salient prediction models
  - transfer, loading and running of prediction models
- Practical principles
  - separation of scenario-independent and scenario-specific concerns
  - management of loading a variety of models for various tasks
  - support for describing a variety of models and tasks

The AI Folk Methodology

get input data
run current prediction model M
use returned prediction

The AI Folk Methodology

```
get input data
run current prediction model $M$
evaluate input data (detect situation)
evaluate if situation fits model $M$ description
  use the semantic description of model $M$
if fit is inadequate
  construct query with current situation parameters
  send query to other agents in the community
  when response received with model $M'$
    integrate description of model $M'$
    replace model $M$ with $M'$ in the pipeline
use returned prediction
```

The AI Folk Methodology

scenario-independent

```
get input data
run current prediction model $M$
  (use general ontological concepts)
  if fit is inadequate
    send query to other agents in the community
    when response received with model $M'$
      integrate description of model $M'$
      replace model $M$ with $M'$ in the pipeline
  use returned prediction
```

scenario-specific

```
evaluate input data (detect situation)
evaluate if situation fits model $M$ description
  use the semantic description of model $M$
  construct query with current situation params
```

The AI Folk Methodology

- AI Folk core ontology
- AI Folk interaction protocol
- deployment and communication infrastructure
- AI Folk model switching behavior

The AI Folk Methodology

- AI Folk core ontology
- AI Folk interaction protocol
- deployment and communication infrastructure – the FLASH-MAS framework
- AI Folk model switching behavior

The AI Folk Methodology

FLASH-MAS – a flexible, modular multi-agent framework · relies on *entities* as first-class abstractions

- *nodes* – containers for entities
- *pylons* – represent communication infrastructures
- *agents* – the actual agents
- *shards* – sub-agent entities encapsulating behaviors
- *drivers* – node-local entities interfacing with other tools

The AI Folk Methodology

FLASH-MAS – a flexible, modular multi-agent framework · relies on *entities* as first-class abstractions

- *nodes* – containers for entities
- *pylons* – represent communication infrastructures
- *agents* – the actual agents
- *shards* – sub-agent entities encapsulating behaviors
- *drivers* – node-local entities interfacing with other tools ← AI Folk-first custom entity

The AI Folk Methodology

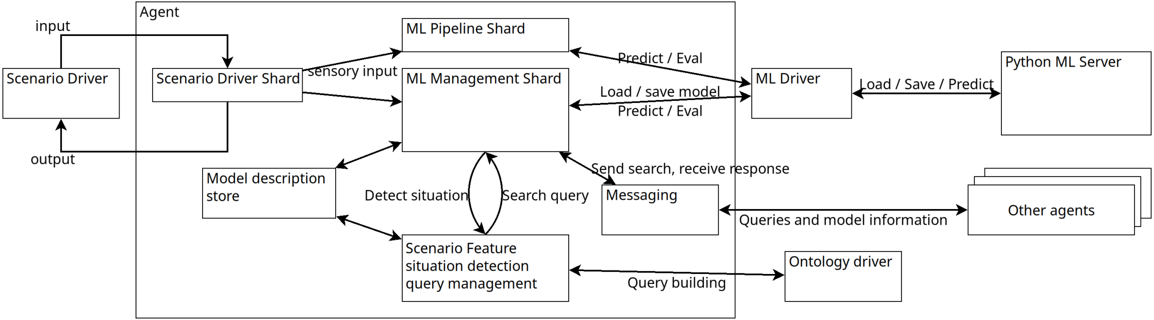
```
            scenario-independent                                        scenario-specific
 [scenario] Get input from the scenario
get input data
 [ML] run current prediction model $M$

                                                         [ML] evaluate input data (detect situation)
                                                         [onto] evaluate if situation fits description of $M$
   [onto] (use general ontological concepts)               use the semantic description of model $M$
  if fit is inadequate

                                                           construct query with current situation params
    send query to other agents in the community
    when response received with model $M'$
      integrate description of model $M'$
      replace model $M$ with $M'$ in the pipeline
  use returned prediction
   [scenario] report output to the scenario
```

The AI Folk Methodology

Given an application domain and a specific prediction task:

- create a domain-specific ontology using the concepts in the AI Folk core ontology and create semantic descriptions of the prediction models in use
- implement a policy which selects the important features in the input data and decides if the description of the current model matches the current situation
- implement the creation of queries that describe the requirements of a given situation

The AI Folk Methodology



Experiments

- initial experiments in the application domain of autonomous driving:
- scenario: an agent receives images and analyzes the number of pedestrian in each image → if the number of pedestrians surpasses the number of pedestrians usually in the data on which the current model has been trained → query agents in the community for other model(s) → switch to a new, more appropriate model

Experiments

Even the deployment of a simple scenario brings technical challenges:

- interoperation between the components implemented in Java and ML models implemented in Python
  - → a local web service is used to interface requests between Java and Python to add, remove, configure, and use ML models

Experiments

Even the deployment of a simple scenario brings technical challenges:

- interoperation between the components implemented in Java and ML models implemented in Python
  → a local web service is used to interface requests between Java and Python to add, remove, configure, and use ML models

- isolation of scenario-specific functionality
  → a new sub-agent abstraction is used – the *Feature* – encapsulating all things related to a specific application domain, e.g. how to decide if a model is appropriate for the current situation or not

Experiments

Even the deployment of a simple scenario brings technical challenges:

- interoperation between the components implemented in Java and ML models implemented in Python
  - → a local web service is used to interface requests between Java and Python to add, remove, configure, and use ML models
- isolation of scenario-specific functionality
  - → a new sub-agent abstraction is used – the *Feature* – encapsulating all things related to a specific application domain, e.g. how to decide if a model is appropriate for the current situation or not
- the usage of ML models is very different, even for the same task, in terms of necessary libraries, processing of input and interpretation of output
  - → ML models are accompanied by a description, as well as a python file containing the code for imports and input and output processing

Experiments

- select an application domain and a task (e.g. road segmentation for autonomous driving)
- select subsets of one or more datasets for the task, which differ significantly in terms of *context*, e.g. weather, day/night conditions, number of pedestrian, type of area (rural, urban, mixed traffic)

Experiments

- select an application domain and a task (e.g. road segmentation for autonomous driving)
- select subsets of one or more datasets for the task, which differ significantly in terms of *context*, e.g. weather, day/night conditions, number of pedestrian, type of area (rural, urban, mixed traffic)
- train / fin-tune one or more ML models on the selected subsets
- define *change conditions* when a prediction model becomes inadequate for the current conditions
- define conditions on which available models are *selected*, based on their description and a set of context requirements

Experiments

- select an application domain and a task (e.g. road segmentation for autonomous driving)
- select subsets of one or more datasets for the task, which differ significantly in terms of *context*, e.g. weather, day/night conditions, number of pedestrian, type of area (rural, urban, mixed traffic)
- train / fin-tune one or more ML models on the selected subsets
- define *change conditions* when a prediction model becomes inadequate for the current conditions
- define conditions on which available models are *selected*, based on their description and a set of context requirements
- evaluate how performance – in terms of both prediction quality and of computational effort – is improved when context-specific fine-tuned models are used in various situations, as compared to using the same model in all contexts

Conclusions

We have developed an infrastructure, tools, and a methodology facilitating the management of pre-trained prediction models in a community of software agents, allowing agents to dynamically decide which models to use, and switch among them.

Future Work

- future: create comprehensive experiments in several fields, validating the AI Folk approach according to the specified evaluation protocol

- future: extend the AI Folk approach to include a feedback on the experience of using a given prediction model

- future: extend the AI Folk approach to allow improving of models via fine-tuning, reaching the interesection with the field of federated learning

Thank You!

Questions are welcome!

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2021-1422, within PNCDI III

[andrei.olaru@upb.ro]

Klampanos, I. A., Davvetas, A., Koukourikos, A., and Karkaletsis, V. (2019).
ANNETT-O: an ontology for describing artificial neural network evaluation, topology and training.
Intl Journ of Metadata, Semantics and Ontologies, 13(3):179–190.

Ma, Y., Wang, Y., Wu, Y., Lyu, Z., Chen, S., Li, X., and Qiao, Y. (2022).
Visual knowledge graph for human action reasoning in videos.
In Proceedings of the 30th ACM International Conference on Multimedia, pages 4132–4141.

Nguyen, A. and Weller, T. (2019).
Fairnets search-a prototype search service to find neural networks.
In SEMANTiCS (Posters & Demos).

Qian, T., Chen, J., Zhuo, L., Jiao, Y., and Jiang, Y.-G. (2024).
nuScenes-QA: A multi-modal visual question answering benchmark for autonomous driving scenario.
In AAAI Conference on Artificial Intelligence, volume 38, pages 4542–4550.

Thank You!

Questions are welcome!

This work was supported by a grant of the Ministry of Research, Innovation and Digitization, CNCS - UEFISCDI, project number PN-III-P1-1.1-TE-2021-1422, within PNCDI III

[andrei.olaru@upb.ro]