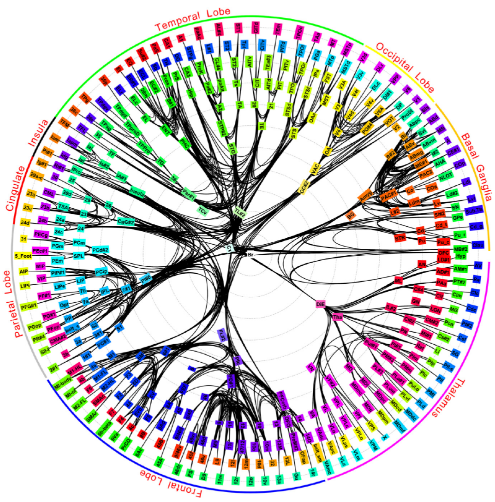
The MASH Project

Validation in Statistics and ML

WIAS, Berlin

Introduction
Complexity in Machine Learning

Artificial learning systems remain extremely simple compared to their biological counterparts.



(Macaque brain long-distance network, Modha and Singh, 2009)

In practice, increasing complexity of ML by simply combining different modalities or prediction methods is very efficient:

- Different modalities tend to catch *complementary information*.
- Different prediction methods tend *to be wrong differently*.

In both case, one ends up combining “independent” variables, and the overall prediction gets better.

The Netflix Prize started in October 2, 2006, and was won in September 2008 by “BellKor’s Pragmatic Chaos”.

> *... we use a set of diverse state-of-the-art collaborative filtering (CF) algorithms, which include: SVD, Neighborhood Based Approaches, Restricted Boltzmann Machine, Asymmetric Factor Model and Global Effects. We show that linearly combining (blending) a set of CF algorithms increases the accuracy and outperforms any single CF algorithm.*

(Jahrer et al., 2010)

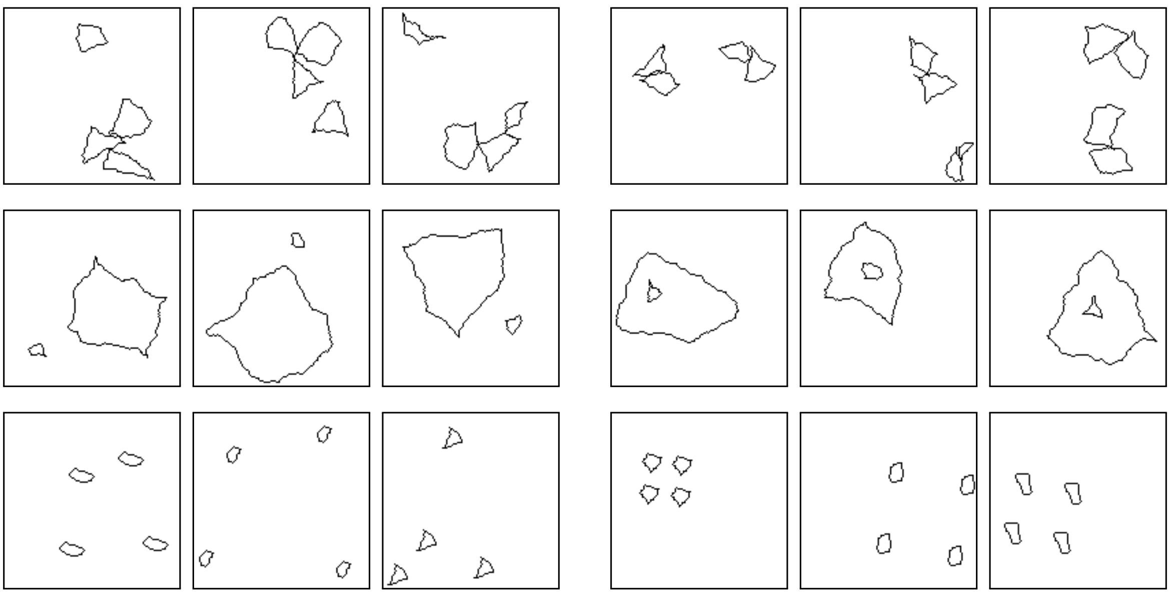
Flower classification

Single features			Combination methods		
Method	**Accuracy**	**Time**	**Method**	**Accuracy**	**Time**
Colour	60.9 ± 2.1	3	Product	85.5 ± 1.2	2
Shape	70.2 ± 1.3	4	Averaging	84.9 ± 1.9	10
Texture	63.7 ± 2.7	3	CG-Boost	84.8 ± 2.2	1225
HOG	58.5 ± 4.5	4	MKL (SILP)	85.2 ± 1.5	97
HSV	61.3 ± 0.7	3	MKL (Simple)	85.2 ± 1.5	152
siftint	70.6 ± 1.6	4	LP-β	85.5 ± 3.0	80
siftbdy	59.4 ± 3.3	5	LP-B	85.4 ± 2.4	98

(Gehler and Nowozin, 2009)

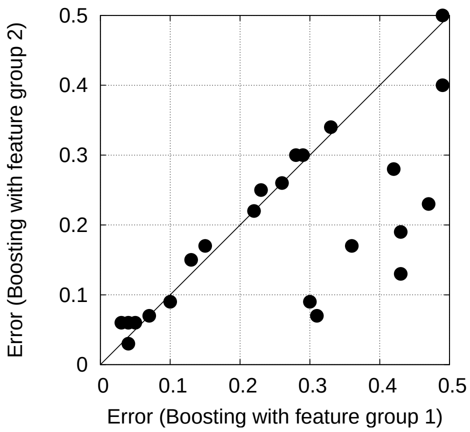
The *learning method* does not matter much.

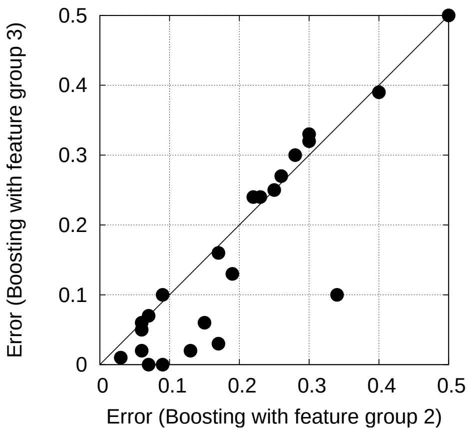
The Synthetic Visual Reasoning Test.



INTRODUCTION
COMPLEXITY IN MACHINE LEARNING

Results on the Synthetic Visual Reasoning Test.





Group 1: Pixel counting
Group 2: Group 1 + Edge-like
Group 3: Group 2 + Fourier-like

INTRODUCTION
THE FOG OF MACHINE LEARNING

So, complex learning systems are worth investigating. However, ML development faces specific difficulties as an engineering task:

- Specifications involve a very complex object (data set, real-world POMDP)
- Limited understanding beyond rough behaviors: Convergence, over-fitting, cost, some invariance.
- Resulting algorithms combine very large numbers of (simple) cues. The emerging behavior is of a different nature.

We often have no idea why it truly works (buggy code sometime works as well ...)

INTRODUCTION
THE FOG OF MACHINE LEARNING

In practice, using ML for applications is a meta-learning algorithm:

Developers go back and forth between identifying mistakes (*High-frequencies patterns generate false alarms!*), fixing them (*Let's add features to detect high frequency blobs!*), repeat (*we do not detect bald people's faces! Let's add features to pick roundish shapes!*), and repeat (*now we are over-fitting! Let's add a L^1 penalty!*)

This is similar to Boosting or SVM: At any moment, the most severe errors drive greedy changes in the constructed predictor.

Human are super-optimizers seeing (a bit) more than the gradient.

THE MASH PROJECT
MOTIVATION

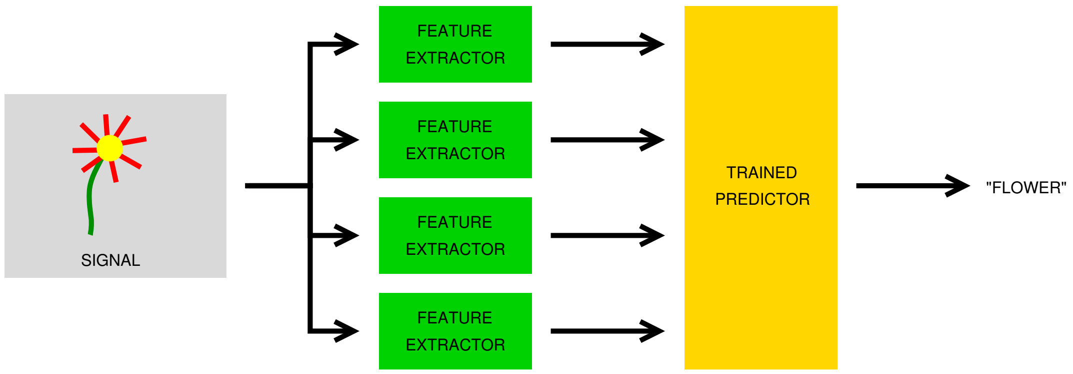
The MASH project is a three-year European research initiative motivated by these observations:

- Machine learning lacks tools to rationalize the design of very complex architectures.
- Combining multiple feature extractors and prediction methods improves performances.
- Internet based collaborative tools allow large teams of individuals to work together.

We want to create new tools for designing complex learning systems in a collaborative manner.

The MASH project
Feature extraction

The project focuses on standard combinations of feature extractors and ML methods.

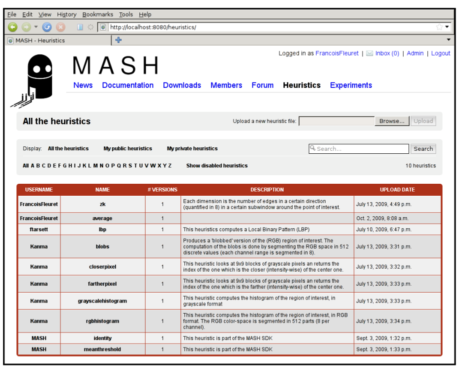


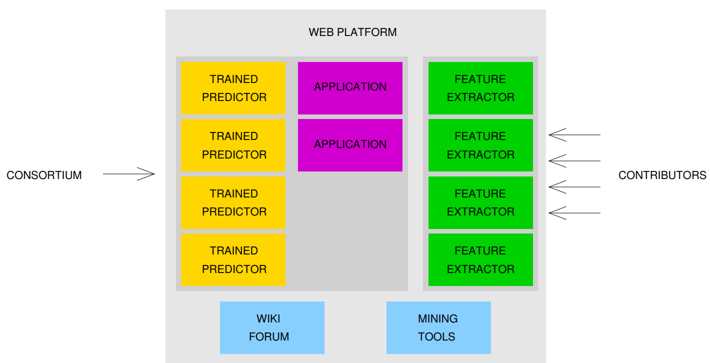
Researchers in the project develop new ML and analysis tools, while *external contributors* design feature extractors.

The MASH project
Overall architecture

Research is organized through a web collaborative platform at

http://mash-project.eu

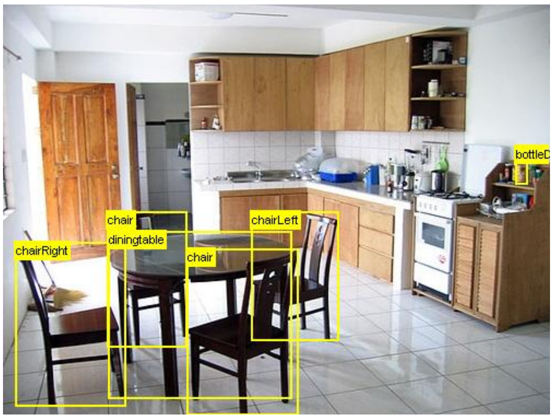




THE MASH PROJECT
APPLICATION

Vision tasks

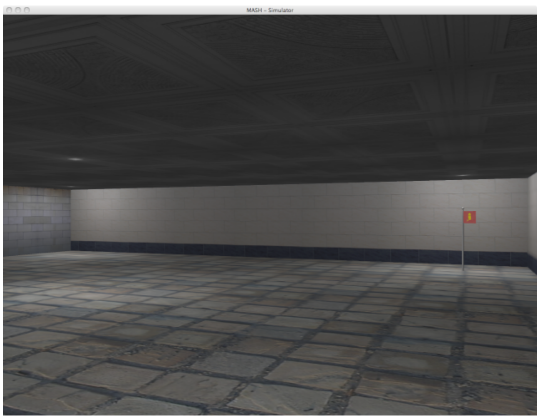
Image classification



Object detection

THE MASH PROJECT
APPLICATION

Goal-planning

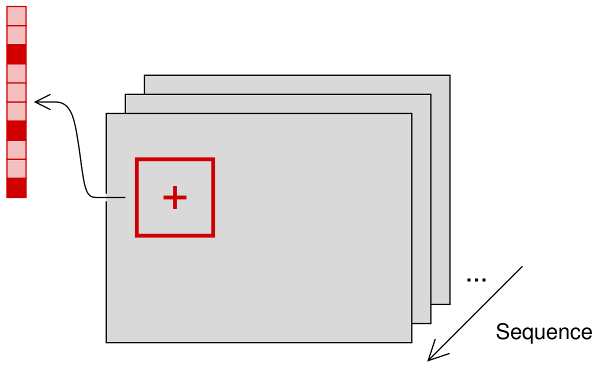


Simulated environment

Robotic arm

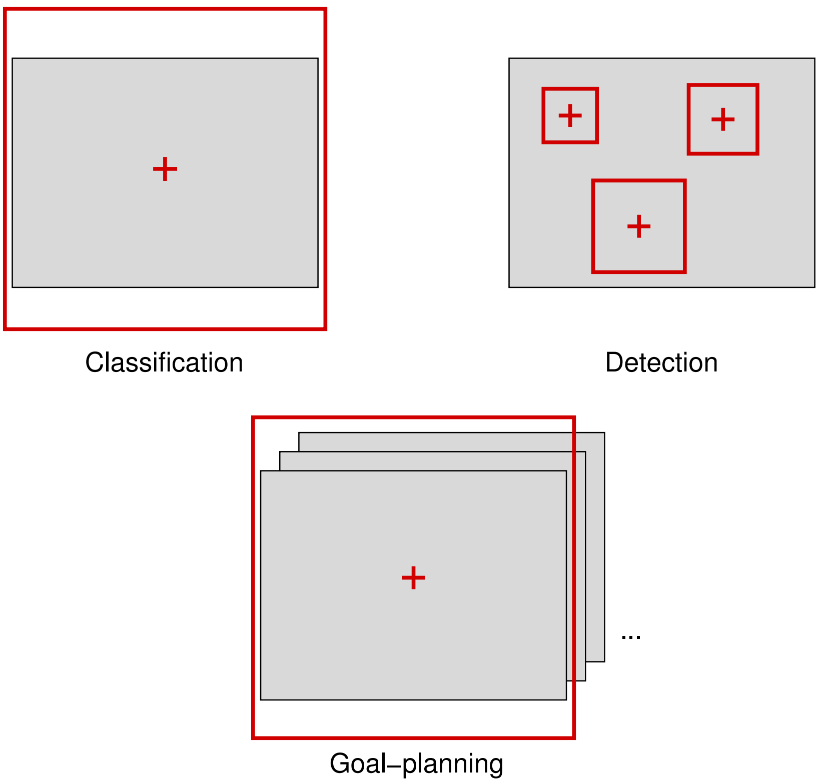
THE MASH PROJECT
NOTION OF HEURISTIC

We define the concept of *Heuristic*, a feature extractor with a persistent state.



Contributions are C++ sources implementing such a heuristic.

THE MASH PROJECT
HEURISTICS FOR APPLICATIONS



The MASH project
Notion of heuristic

If we ignore the persistent state, a heuristic is a mapping

$$H : [0, 1]^{3WH} \times \{1, \ldots, W\} \times \{1, \ldots, H\} \to \mathbb{R}^D$$

```
void init();
unsigned int dim();
void prepareForImage();
void finishForImage();
void prepareForCoordinates();
void finishForCoordinates();
scalar_t computeFeature(unsigned int feature_index);
```


Collaborative platform
Contributor tools

The tools to help contributors include three main components:

- Documentation and social tools: Wiki, screencasts, forum, and private messaging.
- A multi-platform Software Development Kit.
- On-line development tools:
  - **Heuristic repository:** Hosts multiple versions of each heuristics, private or public. Keeps track of the phylogeny.
  - **Private experiments:** Combine a limited number of heuristics, and reduced settings for the database and predictors.
  - **Public experiments:** Defined by the consortium, combine many heuristics and full-scale data-sets.

Collaborative platform

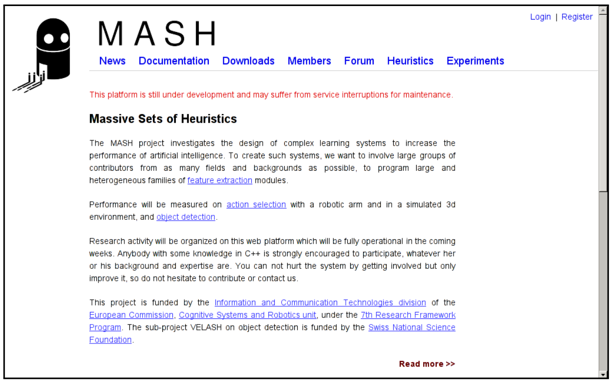
Contributor tools

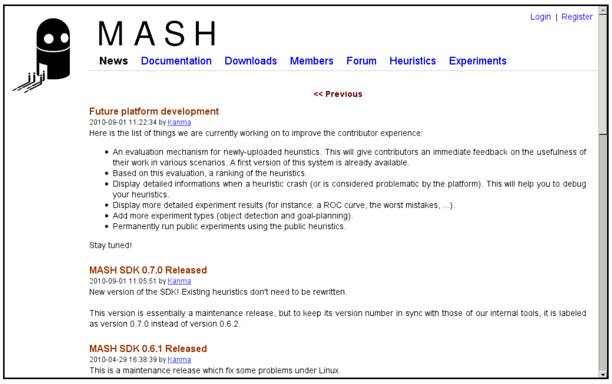
The interaction between a contributor and the MASH system should follow roughly the following steps:

1. Look at the existing heuristics, and the current performance of the system on the public experiments.
2. Download and improve one heuristic, or write one from scratch
3. Upload it, the platform checks it works, runs a series of tests (two settings, ten runs on each at the moment), and provide a ranking.
4. Run additional private experiments to assess more specific strengths and weaknesses of the new contribution, and its complementarity with others.

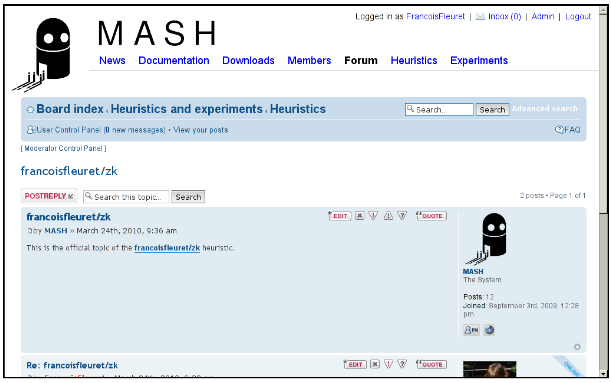
Collaborative platform

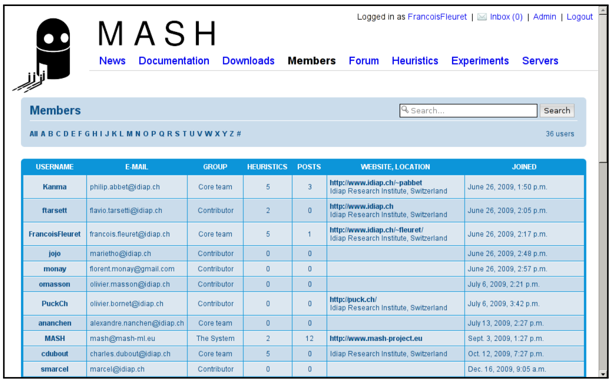
Screenshots



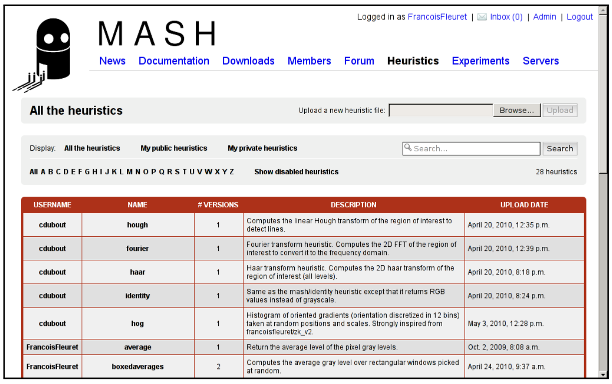


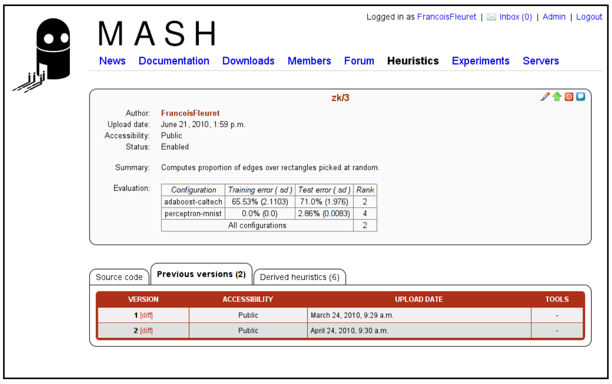
Collaborative platform
Screenshots

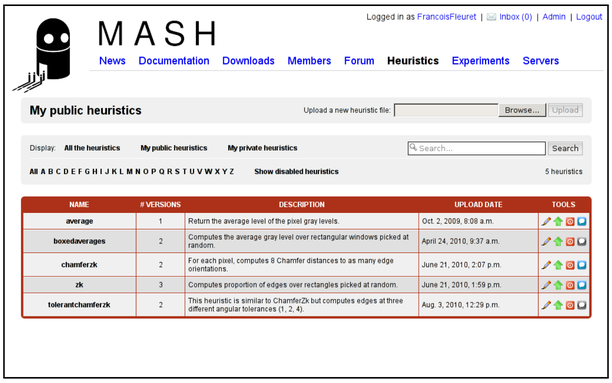




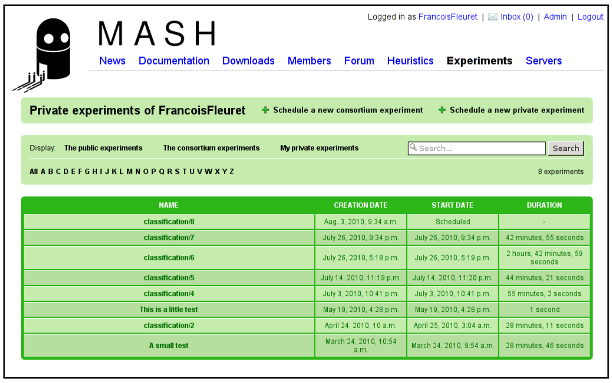
Collaborative platform
Screenshots

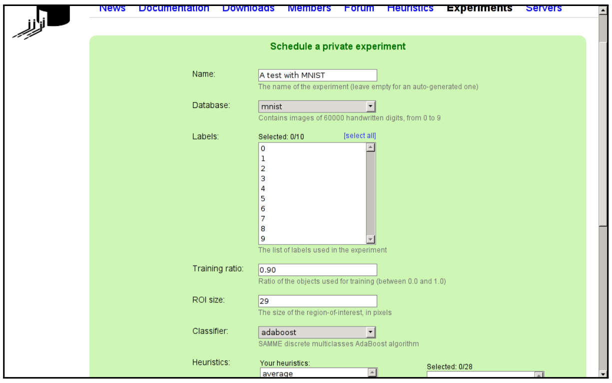






Collaborative platform
Screenshots





Validation
Specific issues

Performance evaluation is crucial to the MASH project, and involves the following difficulties:

- Over-fitting: The feature space is huge.
- “Meta” over-fitting: Contributors will design features specific to the problems at hand.
- Fairness: Contributions must be judged on an equal foot.
- Computational cost: (Pre-)processing can not be factorized as much as usually.
- Security: We have to compile and execute alien code on our machines.
- Intellectual property: The resulting system integrates code from multiple origins.

VALIDATION
OVER-FITTING

We can handle over-fitting with standard recipes:

- Over-fitting: regularization / feature selection methods, together with validation / cross-validation.
- "meta" over-fitting: our own data sets. Some of the samples / goal-planning tasks kept away from the contributors.

The goal-planning problems are randomized at every round (environment geometry, textures, target and avatar placement, etc.)

VALIDATION
FAIRNESS AND COMPUTATIONAL COST

Evaluation has to be fair to motivate contributors properly.

- We test heuristics with identical sample sets (e.g. common random seed in the basic experiments).
- We should estimate how likely a heuristic *could* have been selected during learning.
- Since the source code is public, *slight variations* should not be rewarded too much.
- Application-specific heuristics must not be penalized too much.
- Computation time has to be allocated fairly between contributors. We have setup a time-budget policy.

VALIDATION
FUTURE

The platform has to provide a better analysis of the heuristics and trained predictor performance.

In particular, we are looking at methods to provide users with:

- Mistakes, sorted by severity.
- Contributions, sorted by efficiency.
- Clustering of mistakes, according to their similarities on the contributions.
- Clustering of contributions, according to their similarities on the tasks.

CONCLUSION

The MASH project is a first attempt at allowing large teams to create large-scale complex learning systems.

We want to make tools to do more efficiently what we have been doing for years in ML, and to tap into a larger and more diverse population of experts.

The source code of all the feature extractors will be available over the course of the project under GPL2.

Your contribution is welcome !

THANK YOU

http://mash-project.eu