

Report from the Heavy Flavor Data Mining Workshop



Marcin Chrzaszcz
mchrzasz@cern.ch



University of
Zurich^{UZH}



Universität Zürich,
Institute of Nuclear Physics, Polish Academy of Science

Zurich meeting, CERN
March 4, 2016

Credits where they belong!



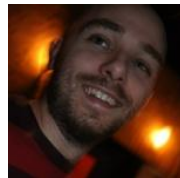
P. Koppenburg



T. Blake



M. Bettler



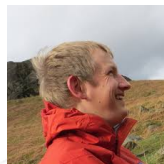
F. Dettori



A. Ustyuzhanin



T. Likhomanenko



T. Head

Workshop sponsors

⇒ Main sponsors:



**University of
Zurich^{UZH}**



⇒ Additional support:



Yandex

Some details

- 50+ participants!
- A good mixture between the ML and Physics community.
- Live discussions!



Useful information

- Indico: <https://indico.cern.ch/event/433556/>
- Full of interesting talks! Please check them out
- Let me try to give you a overview:

⇒ Physics Prize winners:



Alexander Rakhlin, Vicens Gaitan

⇒ We had four ML tutorials in our workshop:

Tutorial "An introduction to Machine Learning with Scikit-Learn".

The screenshot shows the GitHub interface for the repository 'An introduction to Machine Learning with Scikit-Learn' by glouppe. At the top, it displays 44 commits, 1 branch, 0 releases, and 2 contributors. Below this, there are navigation options like 'New file', 'Upload files', 'Find file', and 'HTTPS'. The main content area shows a file list with columns for file name and last update time. The 'README.md' file is selected, showing its content. The README includes a title, a description of the tutorial, contact information, a BSD 3-clause license, and installation instructions. The instructions are numbered: 1) Download and install Anaconda, 2) Install dependencies (conda install numpy scipy scikit-learn jupyter matplotlib), and 3) Clone the repository and start Jupyter. A 'Launch on Binder' button is visible at the bottom of the README content.

File Name	Last Update
img	save 4 months ago
An introduction to Machine Learning with Scikit-Learn.ipynb	Update 16 days ago
README.md	Update README.md 18 days ago
environment.yml	Update environment.yml 18 days ago
tutorial.py	save 4 months ago

An introduction to Machine Learning with Scikit-Learn

Tutorial on machine learning and Scikit-Learn (beginner level).

- Contact: @glouppe
- BSD 3-clause license

Installation instructions

- Download and install the latest Anaconda distribution, coming with Python 3.5 and the full scientific Python stack.
- Install dependencies:

```
conda install numpy scipy scikit-learn jupyter matplotlib
```
- Clone this repository and start Jupyter

```
git clone https://github.com/glouppe/tutorial-scikit-learn.git
cd tutorial-scikit-learn
jupyter notebook
```

Launch on Binder without installing anything!

[Launch Binder](#)

- Gilles Louppe (New York Uni), gave a super tutorial on Scikit-Learn.
- The tutorial was physics oriented → examples of training with weights.
- The material is available on the indico page with Binder.
- Highly recommend to check it out!

Tutorials

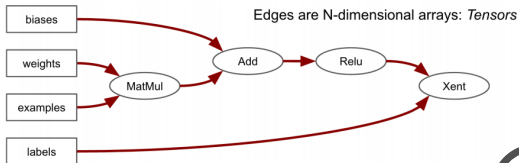
⇒ We had four ML tutorials in our workshop:

TensorFlow Introduction



Presentation by Rafal Jozefowicz, Google Brain

TensorFlow is an open source software library for numerical computation using data flow graphs.



- Rafal Jozefowicz (Google Brain), gave a excellent tutorial on the Googles TensorFlow!
- Since TensorFlow is rather new and most physicists didn't have a chance to hear about it, the tutorial starts from basics!
- The material is available on the indico page.
- Please check it out, this looks like something that we as physicist could really benefit from!

Tutorials

⇒ We had four ML tutorials in our workshop:

- Alison B Lowndes(nVidia), gave a super tutorial on nVidia solutions towards neural networks trainings.
- Very impressive stuff is done by nVidia.
- So impressive that some guys from UZH want to try it out in LHCb!
- Personal view: More effort should be put inside the experiments towards using those cards.



⇒ We had four ML tutorials in our workshop:

- Andrey Ustyuzhanin, Aleskei Rogozhnikov (Yandex), gave a great tutorial on using rep.
- All experiments suffer from reproducibility!
- A mature solution is proposed.

Reproducible Experiment Platform

- | Python-based (numpy, pandas, ...), Jupyter-friendly
- | Unified scikit-learn-like API to many ML packages (Sklearn, XGBoost, uBoost, TMVA, Theanets, ...)
- | Meta-algorithms pipelines («REP lego»)
- | Configurable interactive reporting & visualization to ensure model quality (e.g. check for overfitting)
- | Pluggable quality metrics
- | Parallelized training of classifiers & grid search (IPython parallel)
- | Demo server: <https://lhcb-rep.cern.ch>, password: 'rep'
- | Github: <https://github.com/yandex/rep>

Kaggle Winning solutions

⇒ We have rewarded Kaggle Physics prize on the workshop:

- Vicens Gaitan (Grupo AIA), presented his winning solution.
- He used so called data-doping technique to reduce data-MC agreement.
- Vincens did his PhD with LEP experiments so he understands the two worlds.
- Please check it out as it might be useful for you!
- [Talk](#)

BREAKING THE RULES: DATA DOPING

- The idea is to "dope" (in the semiconductor meaning) the training set with a **small number of Monte Carlo events from the control channel, but labeled as background.**
This disallow the classifier to pick features discriminating data and Monte Carlo.

Control Channel: Known Physics (C), Known Physics (D)
Analysis Channel: New Physics (A), Background (B)
MC: Known Physics (C), New Physics (A)
Real Data: Known Physics (D), Background (B)

There are two parameters that regularize the learning:

- The number of "doping" events
- the complexity of the classifier (for instance number of trees)

Kaggle Winning solutions

⇒ We have rewarded Kaggle Physics prize on the workshop:

- Alexander Rakhlin, presented his winning solution.
- He used so called Transfer learning
- Transfer learning is method that can be used in the training if some of the underlying distributions are not well known.
- I think I don't need to convince anyone that might be useful in physics ;)
- **Talk**

Proposed solution: Transfer Learning



We relate the problem to known paradigm in Machine Learning – **Transfer Learning** between different underlying distributions.

We propose a solution that brings the problem to transductive transfer learning (TTL) and **simple covariate shift**, a primary assumption in domain adaptation framework.

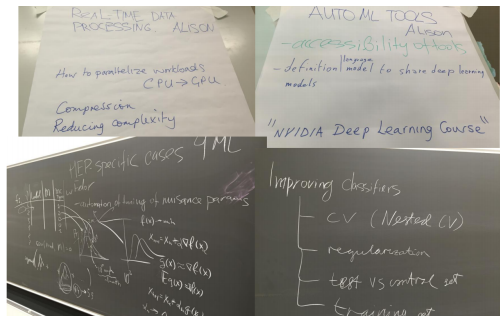
Finally, we present transfer learning model (one of a few) that finished the competition on the 5 place.

Abstract

3

Open-Space discussion

- As an experiment we had a Open-Space discussion.
- It turns out that one can have meetings without conveners ;)
- **Summary Talk**



Other interesting talks

- Automatic Tuning of Hyperparameters
- Classifier output calibration to probability
- Classifiers for centrality determination in proton-nucleus and nucleus-nucleus collisions
- Data Fusion Surogate Modeling on Incomplete Factorial Design of Experiments
- Mathematics of Big Data
- OpenML: Collaborative machine learning
- Boosting applications for HEP
- Efficient Elastic Net Regularization for Sparse Linear Models in the Multilabel Setting
- Deep Learning for event reconstruction

Summary

- ⇒ I hope I interest you enough that you check out the workshop!
- ⇒ The workshop was a success and future events like this should happen!

