INSTITUT FÜR INFORMATIK
Computer Vision, Computer Graphics
and Pattern Recognition

Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Classification of data from the ATLAS experiments

Michael Janschek

Bachelor Thesis

Beginn der Arbeit:	10. Dezember 2015
Abgabe der Arbeit:	10. MÃd’rz 2016
Gutachter:	Prof. Dr. Stefan Harmeling
	Prof. Dr. Stefan Conrad

Erklärung

Hiermit versichere ich, dass ich diese Bachelor Thesis selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Düsseldorf, den 10. MÃd'rz 2016

Michael Janschek

Abstract

Hier kommt eine ca. einseitige Zusammenfassung der Arbeit rein.

Contents

1 Introduction

This chapter first presents the Higgs Boson Machine Learning Challenge and explains its motivation and goals. It is concluded by an overview of the thesis structure.

1.1 The Higgs Boson and the ATLAS experiments

2013 the Nobel prize in physics acknowledged the discovery of the Higgs Boson. A particle the physicist Peter Higgs predicted to exist, giving mass to other elementary particles.

1.2 The Higgs Boson Machine Learning Challenge

Kaggle is an internet community of data scientists, it hosts several competitions posed by businesses or organizations. Further services involve open datasets, a "Jobs Board" and "Kaggle Rankings", a scoreboard based on performances of community-members in Kaggles competitions.

1.2.1 Motivation

1.2.2 Goal

> The goal of the Higgs Boson Machine Learning Challenge is to explore the potential of advanced machine learning methods to improve the discovery significance of the experiment. No knowledge of particle physics is required. Using simulated data with features characterizing events detected by ATLAS, your task is to classify events into "tau tau decay of a Higgs boson" versus "background." [Hig15b]

1.3 Overview

In Chap. 2, we will describe the structure of the challenges dataset [Hig15a] and use simple data-analysis methods, to gain first insight about useful features. We will understand the evaluation metric AMS and related formulas we can use as objective functions for optimizing our classifiers. This will be concluded by deriving the formal problem from the challenges task.

Chap. 3 will use first knowledge about the data to choose simple approaches for classification. After these we will describe several more specific and complex methods.

In Chap. 4 we will use the discussed methods and observe their performance on the challenges data. The thesis closes with a discussion about the approaches and their possible influence on other HEP-applications.

2 Understanding the challenge

In Sec. 2.1, the data will be described and first analysis-methods will be used to gain first knowledge about it. As we continue with sections 2.2 and 2.3, we will formulate a problem to solve and learn about the evaluation the challenge used to rank the participants. Doing so, we rely on [ABCG+15], as this paper was created by the challenge-organizers for giving participants a useful documentation.

2.1 The data

All data provided by the challenge- and the "opendata.cern"-dataset [Hig15a] was created by the official ATLAS full detector simulator in a two-part-process. The simulator first reproduces proton-proton collisions, called *events*. Then it tracks these via a virtual model of the ATLAS-detector, the resulting data emulates the statistical properties of the real events. By this procedure it is possible to exactly know if an event is a searched *signal*, or *background*. Signal-events are generated by tau tau decay of the Higgs Boson. Background-events originate from three known processes[1] which produce radiation similar to the signal [ABCG+15]. The true *class* of an event is contained in the feature *Label* as "s" (signal) and "b" (background).

Every event has a feature *Weight*, an artifact by the simulation. Summed, the weights are "an unbiased estimate of the expected number of events falling in the same region during a given fixed time interval. In our case, the weights correspond to the quantity of real data taken during the year 2012" [ABCG+15]. This relation causes the weight-mean of signal-events to be considerably smaller than background-weights. We will see, that the challenges evaluation utilizes this to punish incorrectly classifying background as signal.

The features $Weight$ and $Label$ were originally only provided in the training-dataset. The data used in this thesis is expanded by complete $Weight$-, $Label$-features and the Kaggle-specific features $KaggleSet$ and $KaggleWeight$. Last one is a normalization of $Weight$ for the total number of signal- or background-events with the same $KaggleSet$ feature. This information enables us recreating the challenges original datasets using the opendata.cern-dataset.

The physical features are separated in two types. Features containing so-called primitive data, properties of events explicitly measured by the simulated ATLAS detector, use the preamble "*PRI_*" in their names. The second type is called derived data, which are features that have been computed from primitive features. Their labels use the preamble "*DER_*".

EventID	DER_mass_MMC	...	Weight	Label	KaggleSet	KaggleWeight
133337	1.337	...	0.4	s	v	0.04

Table 1: Representation of the dataset

[1] decay of Z boson, W boson and a pair of top quarks [ABCG+15]

All features of the dataset are described more detailed in Appendix A.

One might expect decent physics-knowledge as key in succeeding in the challenge, the top-participants did not use a lot domain-knowledge for feature- or method-selection. One goal of the challenges organization was to set a task for data scientists without any physics-background [ABCG+15].

2.1.1 Data visualization

In an first attempt to gain information about our data we use basic methods of data analysis. To evaluate a training sets single feature for classification it is common to plot a histogram of the class-labels, in our case "s" and "b", distribution. A useful way to learn about relations between features is to create *scatter plots* of all two-features-combinations. Using these visualizations, we can identify features with good properties for our task.

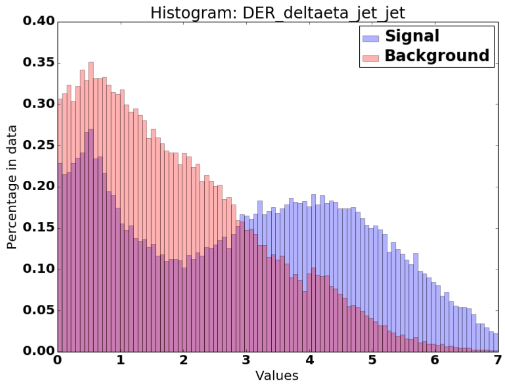


Figure 1:
Histogram of *DER_deltaeta_jet_jet*

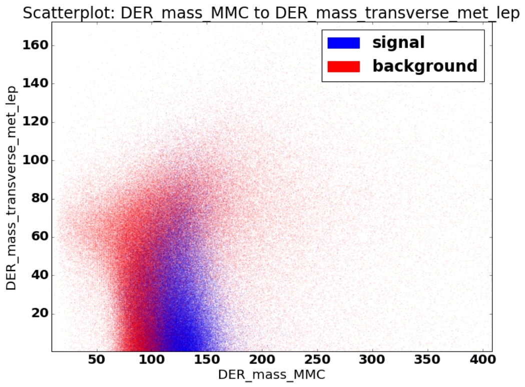


Figure 2: Scatter plot of *DER_mass_MMC* to *DER_mass_transverse_met_lep*

2.2 The formal problem

We use the formal description of the challenge as it is formulated in [ABCG+15].
Let $D = (x_i, y_i, w_i),\ i \,\epsilon\, [1, n]$ be the training sample with n events, where:

- $x_i \,\epsilon\, \mathbb{R}^d$ is a d-dimensional vector
- $y_i \,\epsilon$ {b,s} is the label
- and $w_i \,\epsilon\, \mathbb{R}^+$ is a non-negative weight.

The sums of signal and background-events

$$S = \sum_{y_i = s} w_i$$

and

$$B = \sum_{y_i = b} w_i$$

represent the *expected total number* of signal and background events, during the time of actual data recording.

Let a function $g : \mathbb{R}^d \to \{b, s\}$ be a binary classifier, a set $G_s = \{x : g(x) = s\}$ is called the *selection region*. Our task is to create $g(x_i)$ while maximizing the Approximate Median Significance (AMS). We shall submit an *index set* $\hat{G}_s = \{i : g(x_i) = s\}$ of points that g classifies as signal and an *index set* $\hat{G}_b = \{i : g(x_i) \neq s\}$ while $\hat{G}_s \cap \hat{G}_b = \emptyset$.

2.3 The evaluation

The evaluation of a single submission to the challenge is related to the common practice in particle physics to rate a discovery by its statistical significance, in this case

$$Z = \sqrt{2\left(n \ln\left(\frac{n}{\mu_b}\right) - n + \mu_b\right)} \tag{1}$$

where n is the total number of observed events and μ_b is the expected number of background-events.
Often in particle physics a significance of at least Z=5 (a five-sigma effect) is regarded as sufficient to claim a discovery [ABCG+15].

By estimating $n = s + b$ and $mu_b = b$ in Eq. (1), we get the *$Approximate\ Median\ Significance$* (AMS)

$$AMS = \sqrt{2\left((s+b)\ln\left(1 + \frac{s}{b}\right) - s\right)} \tag{2}$$

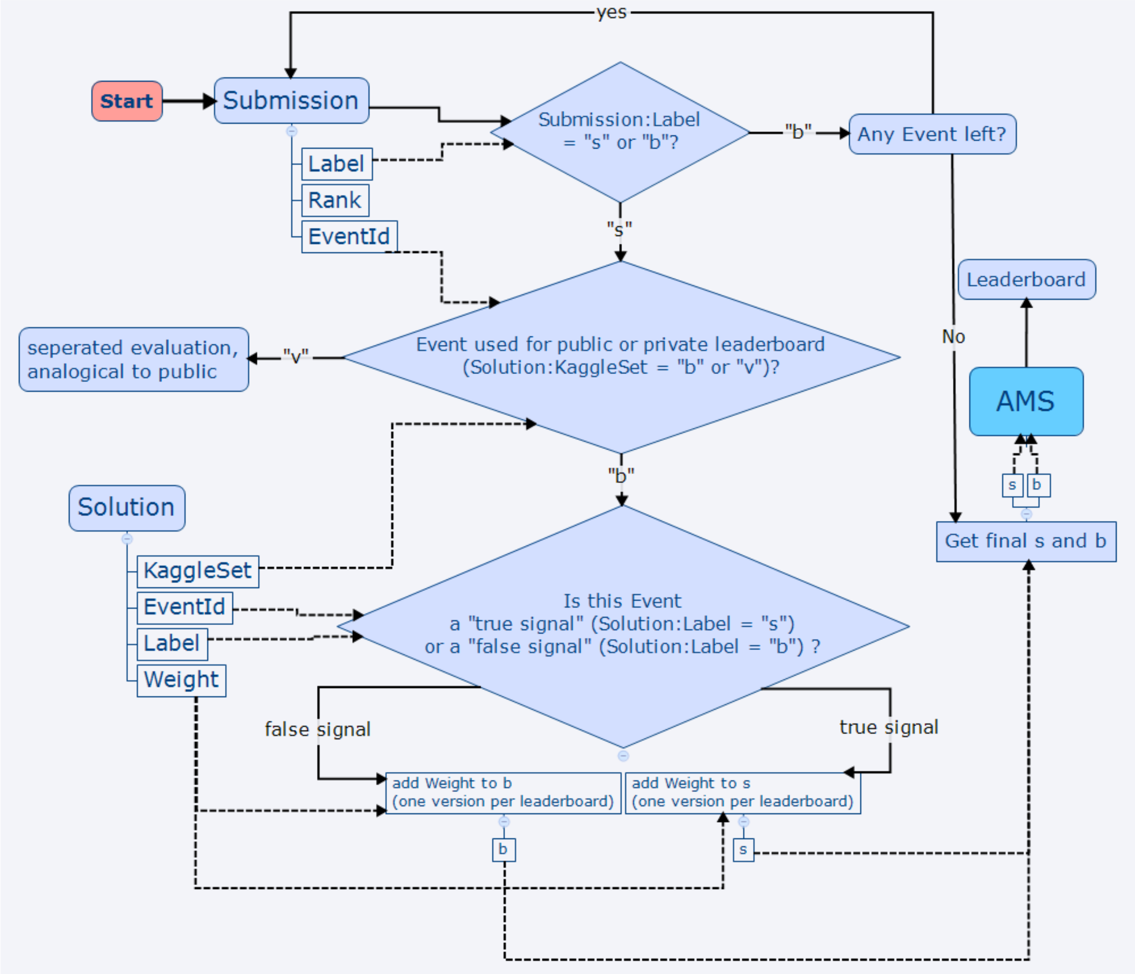
which is used by high-energy physicists for optimizing the selection region for stronger discovery significance [ABCG+15].

For the challenge, a regularization-term b_{reg} was introduced as an artificial shift to b to decrease variance of the AMS, as this makes it easier to compare the participants if the optimal signal region was small. "The value $b_{reg} = 10$ was determined using preliminary experiments." [ABCG+15]

This addition to Eq. (2) makes the final evaluation-formula complete:

$$AMS_2 = \sqrt{2\left((s+b+b_{reg})\ln\left(1 + \frac{s}{b+b_{reg}}\right) - s\right)} \tag{3}$$

For simplicity, we will call it just AMS, as Eq. (2) will not have further appearances in this thesis. Fig. 3 gives a visual representation of the calculation.



yes
Start
Submission
Label
Rank
EventId
Submission:Label = "s" or "b"?
"b"
Any Event left?
"s"
Leaderboard
No
seperated evaluation, analogical to public
"v"
Event used for public or private leaderboard (Solution:KaggleSet = "b" or "v")?
AMS
s
b
"b"
Solution
Get final s and b
KaggleSet
EventId
Label
Weight
Is this Event a "true signal" (Solution:Label = "s") or a "false signal" (Solution:Label = "b") ?
false signal
true signal
add Weight to b (one version per leaderboard)
add Weight to s (one version per leaderboard)
b
s

Figure 3: Flowchart describing AMS calculation for leaderboards

2.3.1 The leaderboard

In Kaggle-Challenges, competitors are ranked in a leaderboard, rank 1 is the participant who submitted a solution which achieved the best score on the evaluation used in this challenge, in our case the AMS (3). To prevent participants from simply training algorithms on optimizing the leaderboard-score, Kaggle uses a so-called *public* and *private leaderboard*, which use different data from the test set to generate the AMS. The weights were normalized with respect to the number of events used for computing the public and private score, in our dataset [Hig15a] this information is given by the features *KaggleSet* and *KaggleWeight* so it is possible to compute these scores.

During the challenge, the public leaderboard is visible to any visitor of Kaggle, so participants are able to get an evaluation of their submitted solution and work on better classification for a higher rank. After reaching the *final submission deadline*, the private leaderboard is accessible, which shows the final ranking of the challenge and the differences to the public leaderboard. Only the private rank is relevant for winning the challenge. Fig.

4 visualizes the final rankings of public and private leaderboards[2]. We notice several big differences in public and private rank of the same submission, a possible sign of overfitting the classifier. In Chap. 5 the variance of the leaderboards will be studied further, for now we conclude to use the public AMS to evaluate our own classification-approaches. This enables better insight to what might cause the differences.

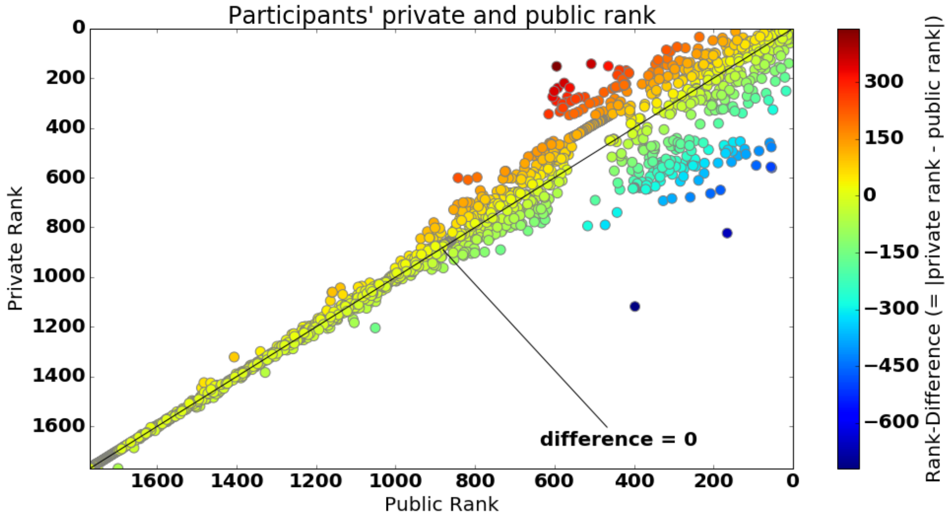


Figure 4: Compared rankings of the public and private leaderboard

[2] data was accessed from [Hig15b] with a simple python-script

3 Methods of classification

- setup
- why these own methods, why are they *simple*?
- why these winning methods
  - XGBoost is awesome, why?
  - resources

3.1 Basic data science methodology

3.1.1 Feature selection and -engineering

An intuitive approach to optimize classifiers is to vary the used features of the data sets. The selection can be performed manually by considering e.g. histograms[Fig.1] and scatter plots[Fig.2]

3.1.2 Cross-validation

Overfitting a model is a known problem in machine learning. In Fig. 4 we observed single submissions to the challenge, that had big differences in ranks on the public and private leaderboard. This could be explained by overfitted classifiers, especially in one case a submission fell 720 ranks from its public to private rank.

A common technique to prevent overfitting is to use only one part of the training set for fitting the classifier while using the other one for testing, calculating a evaluation score, like *mean squared error*(MSE). This method is called *cross-validation*. This procedure is repeated several times and returns a mean score which gives information about the reliability of the classifier and its used parameters.

3.2 Choosing the classifiers

3.2.1 Setup

In order to choose an effective approach to a solution for the challenges task, we need to consider the system which will run the classifiers.

Operating System:	Windows 8.1 64bit
CPU:	Intel i7 K875 @ 2.93GHz (4 Cores)
Memory:	16GB DDR3
GPU:	NVIDIA GeForce GTX470 @ 1.22GHz
VRAM:	1280 MB GDDR5

Table 2: Specifications of used System

While this system enables us to use various methods that utilize minor parallelization, it still limits the performance of classifiers dependent on high parallelization, like neural networks.

3.2.2 scikit-learn

For all own approaches to the challenge we will use *scikit-learn*, an open source machine learning library for Python. Originated from a "Google Summer of Code"-project started 2007, it grew into a popular toolkit. Using scikit-learn has advantages as disadvantages, we consider the most important ones.

Advantages:	
Usability	scikit-learn is easily installed via Python-package manager systems like *pip* or *Anaconda*, its required dependencies are NumPy, SciPy, and a working CC++ compiler[sci16b].
Scale and documentation	A reason for its popularity is the amount of "well-established algorithms"[sci16a] this toolkit contains. It also offers good code-documentation and mathematical explanation of all models included.
Multi-core CPU support	With every version [sci16c] scikit-learn improves its multi-core support for algorithms that profit of parallelization. This will speed up our classification considerably.
Disdvantages:	
No GPU support:	While the library utilizes multi-core CPUs, GPU-parallelization is not supported by
(Almost) no custom evaluation functions	scikit-learn offers rarely an easy way to use custom evaluation functions for e.g. fitting a classifier. Some algorithms, like *sklearn.cross_validation*, offer choice of pre-built functions via a *scoring*-parameter.
Speed	One focus of the development of the toolkit is optimizing and speed-up of the algorithms, but projects devoted to single learning techniques are probably faster.

Table 3:
Comparison of properties of scikit-learn

3.2.3 Shape of data

The choice of classifiers is dependent on the data. The training set contains 250000 samples with 32 features (EventId not counted), the test set 550000 samples with 30 features. Considering this some classification methods would be ineffective, e.g. *Support Vector Classification* due to its complexity $> O(n^2)$[sci16a].

3.3 Logistic regression

- why logreg?
- what is logreg?
- optimizing objective function (via cv => ROC-AUC)
- performance
- AMS

3.4 k-nn classification

- why kNN?
- what is kNN?
- optimization
- performance
- AMS
- [PPSS14] are noobs :-P

3.5 The winning methods

resources

3.5.1 Neural networks

1st place https://github.com/melisgl/higgsml https://github.com/melisgl/higgsml/blob/master/doc/model.md 24gb+ ram minimum: titan gpu 3rd place https://www.kaggle.com/c/higgs-boson/forums/t/10481/third-place-model-documentation/55390#post55390

3.5.2 Regularized greedy forest

https://github.com/TimSalimans/HiggsML 2nd place Requires 64gb+ RAM

3.5.3 XGBoost

special prize 1st AND 2nd place in second CERN-Kaggle challenge!

impact of xgboost?

[CH14]

4 Results on the Kaggle data

4.1 k-nn classification

4.2 Comparision of all methods

5 Discussion

5.1 Impact of the Challenge

Use [CGG+14]!

- Ersetzen von AMS durch weighted AUC (papers) [DMNV14]
  - Leaderboard-Shakeup
  - video bei flavour of physics
- XGBoost and Kaggle
- weitere CERN-Challenges

[DMNV14]

References

[ABCG$^+$15] ADAM-BOURDARIOS, Claire ; COWAN, Glen ; GERMAIN, Cécile ; GUYON, Isabelle ; KÉGL, Balázs ; ROUSSEAU, David: *Learning to discover: the Higgs boson machine learning challenge*. http://www.http://opendata.cern.ch/record/329, January 2015. – Version 2.3

[CGG$^+$14] COWAN, Glen (Hrsg.) ; GERMAIN, Cécile (Hrsg.) ; GUYON, Isabelle (Hrsg.) ; KÉGL, Balàzs (Hrsg.) ; ROUSSEAU, David (Hrsg.) ; CERN, CRNS (Veranst.): *NIPS 2014 Workshop on High-energy Physics and Machine Learning*. 2014

[CH14] CHEN, Tianqi ; HE, Tong: Higgs Boson Discovery with Boosted Trees. In: *NIPS 2014 Workshop on High-energy Physics and Machine Learning* Bd. 42, 2014 (JMLR: Workshop and Conference Proceedings), S. 69–80. – http://jmlr.csail.mit.edu/proceedings/papers/v42/diaz14.pdf , Accessed: 2016-01-18

[DMNV14] DÍAZ-MORALES, Roberto ; NAVIA-VÁZQUEZ Ángel: Optimization of AMS using Weighted AUC optimized models. In: *NIPS 2014 Workshop on High-energy Physics and Machine Learning* Bd. 42, 2014 (JMLR: Workshop and Conference Proceedings), S. 109–127. – http://jmlr.csail.mit.edu/proceedings/papers/v42/diaz14.pdf , Accessed: 2016-01-18

[Hig15a] *Dataset from the ATLAS Higgs Boson Machine Learning Challenge 2014*. http://www.http://opendata.cern.ch/record/328, 2015. – Accessed: 2015-12-10

[Hig15b] *Higgs Boson Machine Learning Challenge*. https://www.kaggle.com/c/higgs-boson, 2015. – Accessed: 2016-01-03

[PPSS14] PEREZ, Jocelyn ; PONMALAI, Ravi ; SILVER, Alex ; STRACK, Dacoda: *ML2014: Higgs Boson Machine Learning Challenge*. http://www.http://opendata.cern.ch/record/331, 2014. – Accessed: 2015-12-10

[sci16a] *scikit-learn Homepage*. http://scikit-learn.org/stable/index.html, 2016. – Accessed: 2016-02-03

[sci16b] *scikit-learn on Github*. https://github.com/scikit-learn/scikit-learn, 2016. – Accessed: 2016-02-03

[sci16c] *scikit-learn Version-History*. http://scikit-learn.org/stable/whats_new.html, 2016. – Accessed: 2016-02-03

List of Figures

List of Tables

A The detailed description of the features