Moderne Methoden der Datenanalyse

Multivariate Analyse: Higgs Challenge

Lukas Fritz	1686473
Fabian Leven	1638446
Ali Deniz Özdemir	1724032
Lena Salfenmoser	1723697
Johannes Heizmann	1725035

20. Juli 2016

Inhaltsverzeichnis

1 Einleitung **3**

2 Auswahl einer geeigneten Untermenge an Variablen **3**
2.1 Vorgehen 3
2.2 Auswertung 4

3 Transformation der Eingabevariablen **5**
3.1 Vorgehen 5
3.2 Auswertung 7

4 Optimierung der Parameter des Fisher-Algorithmus **8**

5 Optimierung der Parameter der Likelihood-Methode **8**

6 Optimierung der Parameter des BDT-Algorithmus **8**

7 Optimierung der Parameter des Neuronalen Netzes **11**

8 Fazit und unser Score **13**

1 Einleitung

Die "Higgs-challenge" wurde im Jahr 2014 von einer Gruppe von Wissenschaftlern der ATLAS-Kollaboration ins Leben gerufen. Dabei sollen die Originaldaten des ATLAS-Experiments auf Zerfälle des Higgs-Bosons in zwei Tau-Leptonen untersucht werden. Hierfür wurden Simulations- und Testdaten zur Verfügung gestellt, mithilfe derer Klassifikationsalgorithmen trainiert werden konnten, um schließlich im Original-Datenset zwischen "Hintergrund" und "Tau-Tau-Zerfall eines Higgs" unterscheiden zu können. Die "challenge" besteht dabei darin, die verschiedenen Methoden des maschinellen Lernens optimal auf die gegebene Situation anzupassen, sodass die Klassifizierung möglichst effizient erfolgen kann.

Das Training sowie die Analyse der Daten erfolgt mit dem "Toolkit for Multivariate Data Analysis with ROOT" (kurz: TMVA), das verschiedene Methoden des maschinellen Lernens bereitstellt. Von diesen waren vier (Maximum Likelihood-Methode, Fisher-Diskriminante, Boosted Decision Trees und neuronales Netzwerk) in einem Template vorgegeben. Die Mess- bzw. Trainingsdaten umfassten 30 Variablen. Da nicht alle gemessenen Werte eine Aussagekraft bezüglich der Entscheidung "Hintergrund" oder "Signal" besitzen, galt es zunächst, aus diesen 30 Variablen eine geeignete Untermenge auszuwählen. Anschließend wurde untersucht, inwiefern eine Transformation der (übrigen) Eingabevariablen, sowie eine Anpassung der Parameter der Algorithmen des maschinellen Lernens die Effizienz der Klassifizierung verbessern konnte. Ausschlaggebendes Kriterium war hierbei jeweils der im Training erzielte approximate median significance-Wert (kurz: AMS-Wert). Sobald ein zufriedenstellendes Ergebnis erreicht wurde, konnte der Klassifizierungsalgorithmus auf die Originaldaten angewandt werden.

2 Auswahl einer geeigneten Untermenge an Variablen

2.1 Vorgehen

Um eine geeignete Untermenge an Variablen für die multivariate Analyse zu finden, haben wir die Relevanz der Variablen, die je nach verwendetem Trainingsalgorithmus (Likelihood, Fisher, BDT, MLP) unterschiedlich ausfallen kann, evaluiert. Hierzu haben wir verschiedene Verfahren angewandt, die im Folgenden beschrieben werden.

Im Zuge der Ausführung der Trainingsalgorithmen des vorgegebenen TMVA-Templates wird von diesem für jede Klassifizierungsmethode eine Rangliste bzgl. der Wichtigkeit der einzelnen Variablen erstellt. Um die Tauglichkeit dieser Rangliste zu überprüfen, betrachteten wir die Trainingsalgorithmen separat und trainierten den Likelihood-, Fisher-, sowie den BDT-Algorithmus jeweils insgesamt 30 mal, wobei entsprechend der vom TMVA-Template ausgegebenen Reihenfolge jeweils eine Variable entfernt wurde (beginnend mit der irrelevantesten).Bei jedem dieser insgesamt 90 Trainings wurde der ausgegebene AMS-Wert für das entsprechende Variablen-Subset protokolliert. Die Entwicklung der AMS-Werte mit abnehmender Variablenzahl sind in Abbildung 1 gezeigt. Es ist deutlich zu erkennen, dass der Likelihood-Algorithmus einen ähnlichen Verlauf wie der Fisher-Algorithmus aufweist. Dies lässt sich dadurch erklären, dass es sich hierbei

um ähnliche Algorithmen handelt. Eine sinnvolle Untermenge an Variablen liese sich demnach mithilfe des Maximums des AMS-Verlaufs bestimmen.

Da das Training des neuronalen Netzwerks mit Abstand die meiste Zeit benötigt, wurde hier ein anderes Verfahren angewandt: Es wurden 30 Trainingsläufe mit je einer entfernten Variablen durchgeführt und die Variablen entsprechend des jeweils erreichten AMS-Werts sortiert. Anschließend führten wir das zuvor beschriebene Verfahren gemäß dieses Rankings durch und erhielten den in Abbildung 1 gezeigten Verlauf der AMS-Werte.

Dieses Verfahren stellte den Startpunkt für eine tiefergehende Analyse der Relevanz der einzelnen Variablen dar: Für jede Methode wurde das Training 30 mal mit jeweils einer aus dem Parameterset entfernten Variable durchgeführt, wobei die irrelevanteste Variable (höchster AMS-Wert) ermittelt wurde. Diese wurde aus dem Variablen-Set entfernt und das Training 29 mal durchgeführt, wobei jeweils eine der übrigen Variablen entfernt wurde. Je nach erzieltem AMS-Wert wurde das Parameter-Subset wiederum um den unwichtigsten Parameter reduziert usw. Da dieses Vorgehen aufgrund der langen Trainingszeit des neuronales Netzwerks hierfür nicht durchzuführen war, beschränkten wir uns hierbei auf die übrigen drei Verfahren. Insgesamt wurde zur Erstellung der entsprechenden Rankings das Training $3 \cdot 465 = 1395$ mal durchgeführt. Der Verlauf des AMS-Werts in Abhängigkeit der Anzahl verwendeter Parameter ist Abbildung 1 zu entnehmen. Da die AMS-Werte höher liegen als bei dem zuvor beschriebenen Verfahren, ist diese Rangliste als aussagekräftiger zu erachten.

2.2 Auswertung

Die Ergebnisse für den Verlauf der AMS-Werte bzgl. der verschiedenen Ranglisten sind in Abbildung 1 gezeigt. Ein senkrechter Strich markiert jeweils das Maximum, anhand dessen die Auswahl der geeigneten Variablenuntermenge festgelegt wurde: Alle Variablen, die zu diesem Zeitpunkt noch Teil des Trainingsprozesses waren, sind Teil der für die weitere Auswertung der Daten gewählten Variablenuntermenge. Welche Variablen dies für die jeweiligen Trainings-Algorithmen sind, ist Tabelle 1 zu entnehmen. An oberster Stelle in der Tabelle stehen dabei die für die Analyse wichtigsten Parameter. Ein waagrechter Strich zeigt die Begrenzung der Variablenuntermenge an.

Das Verkleinerung der Variablenanzahl wirkt sich zweierlei auf die Analyse aus. Erstens kann der erreichte AMS-Wert steigen oder sinken. Zweitens reduziert sich die Trainingszeit. Anhand dieser beiden Kriterien haben wir eine Auswahl für die Untermenge an Variablen getroffen:

Eine deutliche Verbesserung des AMS-Werts durch eine Verkleinerung der Variablenanzahl zeigt sich lediglich bei der Likelihood-Methode (siehe Abbildung 1). Aus diesem Grund arbeiteten wir bei weiteren Untersuchungen dieser Methode mit der verkleinerten Variablenmenge.

Die Fisher-Methode zeigte kleine Verbesserungen des AMS-Wertes bei verkleinerter Variablenanzahl. Die Trainingslaufzeit ist allerdings so kurz, dass es kaum einen Unterschied macht, ob man den vollen Variablensatz verwendet oder einen reduzierten. Welche Auswahl im Einzelnen getroffen wurde, ist im Einzelfall dokumentiert.

Die BDT-Methode wird nur gering von der Anzahl der Variablen beeinflusst. Ob mit der vollen Variablenmenge weiter gearbeitet wurde, oder der verkleinerte Satz benutzt wurde, ist an den entsprechenden Stellen dokumentiert.

Für das neuronale Netz ist die Trainingszeit erheblich. Aus diesem Grund arbeiteten wir mit der verkleinerten Variablenuntermenge weiter.

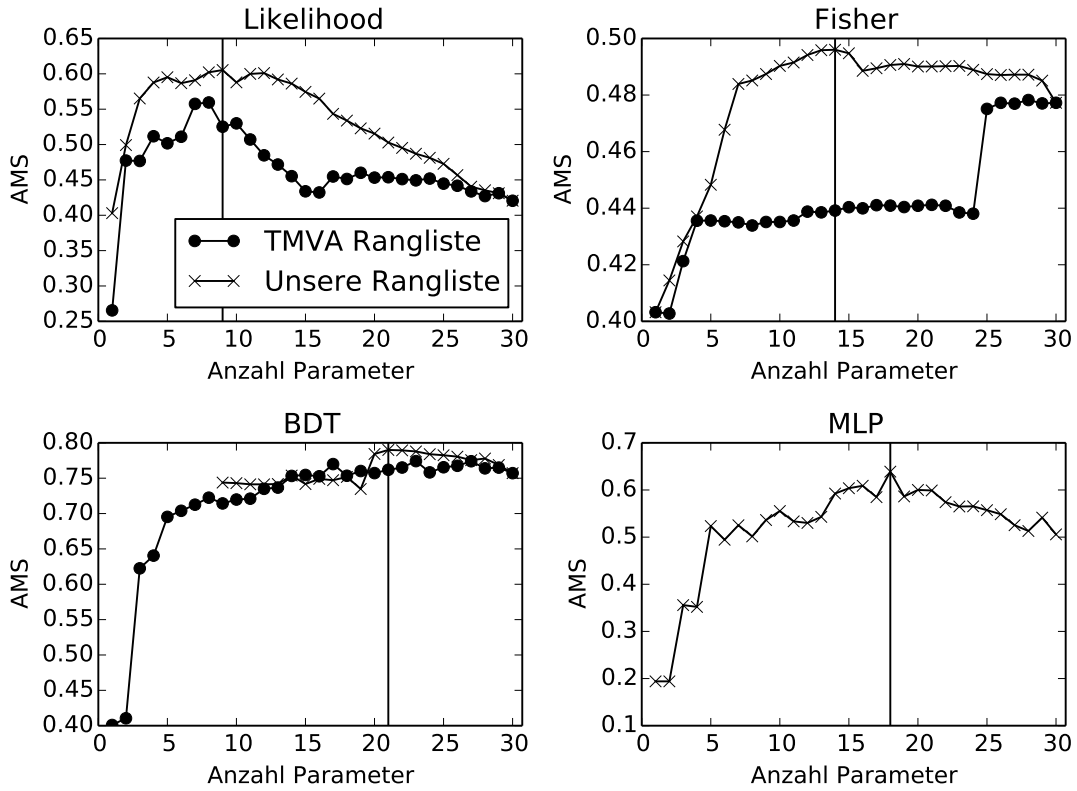


Abbildung 1: AMS über der Anzahl der Variablen.

3 Transformation der Eingabevariablen

3.1 Vorgehen

Gemäß der Aufgabenstellung in der Template-Datei, experimentierten wir mit den Transformationen „Decorrelate“ (D), „Gauß“ (G), und „Normalise“ (N). Um heraus zu finden, ob sie einen Einfluss auf das Ergebnis haben, wendeten wir sie in allen Kombinationen und Reihenfolgen auf den Input an und evaluierten den AMS-Wert. Evaluiet wurden die Fisher- und die BDT-Methode mit der Untermenge an Variablen, die gemäß Abschnitt 2 ermittelt wurden.

Tabelle 1: Rangliste der Variablen bzgl. ihrer Relevanz für die Analyse. Die Wichtigkeit der Parameter nimmt von oben nach unten ab. Für jede Methode ist durch einen waagrechten Strich die durch das Maximum des AMS-Verlaufs gegebene Abgrenzung einer geeigneten Auswahl an Variablen gekennzeichnet. Bei der BDT-Methode wurde unter den besten acht Parametern kein Ranking vorgenommen.

Likelihood	Fisher	BDT	MLP
d_mass_transverse_met_lep	d_mass_transverse_met_lep	-	p_jet_num
d_mass_vis	d_pt_ratio_lep_tau	-	d_deltaeta_jet_jet
p_tau_pt	p_lep_pt	-	d_mass_MMC
d_deltar_tau_lep	d_mass_vis	-	p_jet_leading_pt
d_pt_ratio_lep_tau	d_deltar_tau_lep	-	d_mass_transverse_met_l
p_met	d_pt_h	-	p_jet_leading_eta
p_lep_pt	p_tau_pt	-	d_mass_jet_jet
p_lep_eta	d_mass_jet_jet	-	p_jet_leading_phi
d_mass_MMC	p_met	d_mass_MMC	d_pt_h
p_jet_leading_phi	d_prodeta_jet_jet	d_lep_eta_centrality	d_mass_vis
d_met_phi_centrality	p_jet_subleading_pt	p_jet_subleading_phi	p_jet_subleading_phi
p_tau_phi	p_lep_phi	p_tau_pt	d_prodeta_jet_jet
p_met_phi	d_mass_MMC	d_mass_vis	p_jet_subleading_pt
p_tau_eta	d_deltaeta_jet_jet	p_tau_phi	p_tau_pt
p_jet_subleading_pt	d_lep_eta_centrality	p_met_sumet	p_met_phi
d_pt_tot	p_jet_leading_phi	p_lep_pt	p_jet_all_pt
p_jet_subleading_eta	p_jet_leading_eta	p_lep_phi	p_met_sumet
d_prodeta_jet_jet	d_sum_pt	p_jet_subleading_pt	p_tau_eta
p_jet_subleading_phi	p_jet_subleading_eta	p_jet_subleading_eta	p_met
p_met_sumet	p_tau_phi	d_pt_tot	d_met_phi_centrality
d_deltaeta_jet_jet	p_met_phi	p_jet_subleading_eta	d_pt_ratio_lep_tau
d_mass_jet_jet	p_jet_leading_pt	p_jet_subleading_phi	p_tau_phi
d_lep_eta_centrality	p_tau_eta	p_jet_subleading_pt	p_jet_subleading_eta
p_lep_phi	p_met_sumet	d_pt_h	p_lep_phi
p_jet_num	p_jet_num	p_met	d_pt_tot
d_pt_h	d_pt_tot	d_prodeta_jet_jet	p_lep_eta
d_sum_pt	p_lep_eta	p_lep_pt	d_sum_pt
p_jet_leading_pt	p_jet_subleading_phi	d_pt_tot	d_lep_eta_centrality
p_jet_all_pt	p_jet_all_pt	p_met_phi	p_lep_pt
p_jet_leading_eta	d_met_phi_centrality	p_lep_phi	d_deltar_tau_lep

3.2 Auswertung

Die Ergebnisse zeigt Abbildung 2. Man sieht, dass es für die Fisher-Methode Sinn macht, die Daten zu dekorrelieren und zu Normalisieren. Die Reihenfolge hat keine Auswirkungen. Bei der BDT-Methode verschlechtern die Transformationen das Ergebnis. Außerdem gab es bei dieser Methode bei der Dekorrelation einen Programmfehler, den wir nicht weiter untersuchten. In Abbildung 2 sind nur die erfolgreichen Abläufe der BDT-Methode gelistet und deshalb weniger als bei der Fischer-Methode.

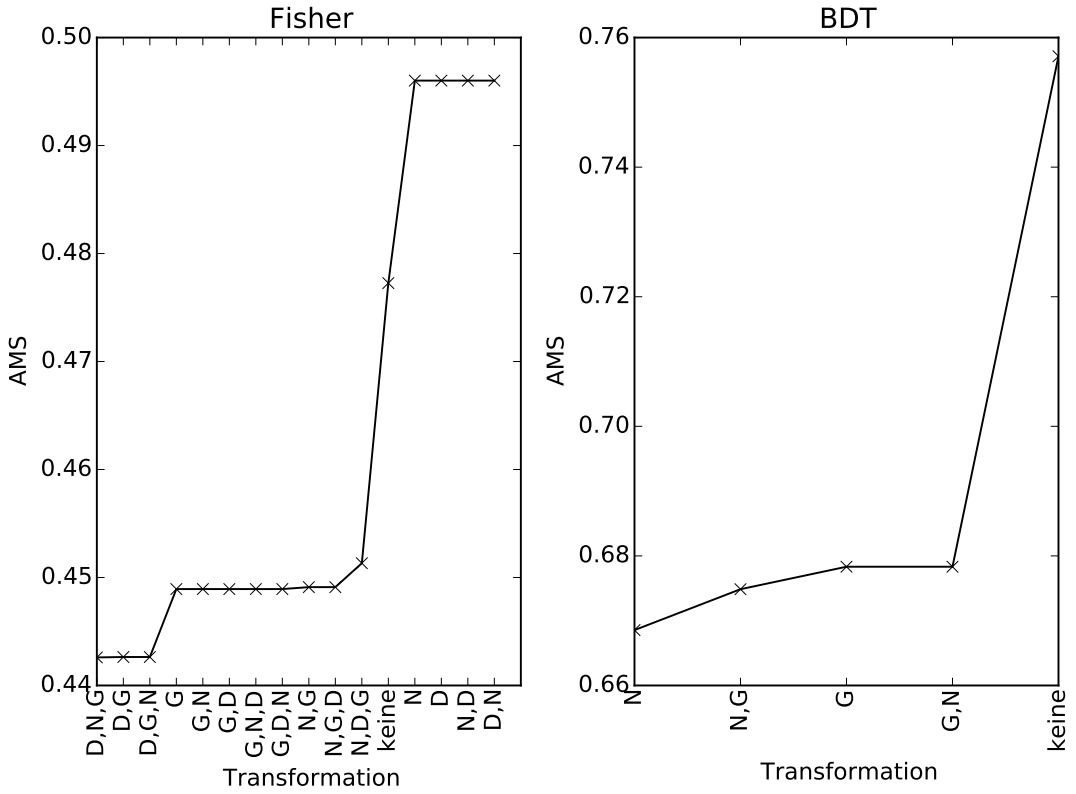


Abbildung 2: AMS über Transformationen der Eingabevariablen. Die Buchstaben stehen für „Decorrelate“ (D), „Gauß“ (G), und „Normalise“ (N). Die Transformationen wurden in der Reihenfolge ausgeführt, wie die Buchstaben von unten nach oben gelistet sind.

4 Optimierung der Parameter des Fisher-Algorithmus

Um einen höheren AMS-Wert zu erreichen, variierten wir folgende zwei Parameter des Fisher-Algorithmus.

$$\begin{aligned} \textit{PDFInterpolMVAPdf} &\in \{\text{ Spline1, Spline2, Spline3}\} \\ \textit{NsmoothMVAPdf} &\in \{5,10,15, \ldots, 195\} \end{aligned} \tag{4.1}$$

Der erste Parameter steht für den Grad der Splines mit denen die Wahrscheinlichkeitsverteilungen der Eingabeparameter interpoliert werden. Der zweite Parameter steht für die Anzahl an iterativen Glättungen dieser Verteilungen. Das Training dieser Methode mit dem gegebenen Datensatz kostet wenig Zeit - meistens weniger als Minute. Das macht es möglich mit einem brute-force-Ansatz alle Parameterkonstellationen durchzuprobieren. Der AMS-Wert war unabhängig von der Wahl der Parameter *PDFInterpolMVAPdf* und *NsmoothMVAPdf* konstant 0.496. Wir verzichteten auf weitere Optimierungen unter Verwendung des Fisher-Algorithmus, da die erzielten AMS-Werte deutlich unter denen der BDT-Methode lagen.

5 Optimierung der Parameter der Likelihood-Methode

Um den AMS-Wert zu erhöhen, wurden drei der Parameter der Likelihood-Methode innerhalb folgender Grenzen variiert:

$$\begin{aligned} \textit{PDFInterpol} &\in \{\text{ Spline0, Spline1, Spline2, Spline3, Spline5}\} \\ \textit{NSmooth} &\in \{0,\ 1,\ \ldots,\ 12\} \\ \textit{NAvEvtPerBin} &\in \{38,\ 39,\ \ldots,\ 52\} \end{aligned} \tag{5.1}$$

Die Verteilung der erzielten AMS-Werte ist Abbildung 3 zu entnehmen. Da auch bei diesem Verfahren die erzielten AMS-Werte deutlich unter $0,5$ blieben, entschieden wir uns dafür, auch diesen Algorithmus in der weiteren Analyse nicht weiter zu verwenden.

6 Optimierung der Parameter des BDT-Algorithmus

Bei dem Boosted Decision Tree wurde der AMS schlechter wenn Variablen entfernt wurden, daher wurde der volle Satz an Variablen verwendet. Um die optimalen Parameter für den BDT zu finden, haben wir für die Parameter

- NTrees
- NEventsMin
- Shrinkage
- nCuts

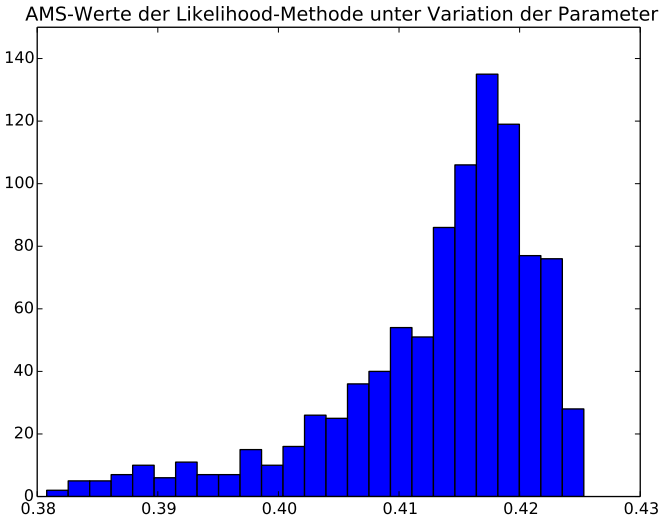


Abbildung 3: Erzielte AMS-Werte.

- MaxDepth

sinnvolle Bereiche festgelegt und wollten in diesem Parameterraum den besten AMS finden. Da die Laufzeit des Trainings für jeden Punkt im Parameterraum in der Größenordnung von Minuten liegt und wir 5 Parameter haben, lassen sich herkömmliche Optimierungsalgorithmen nicht sinnvoll anwenden.
Um einen Überblick zu erhalten ob es gewisse Bereiche gibt in denen der AMS generell besser ist, haben wir sehr oft (1389 mal) zufällige Parameter gewählt und den AMS dazu gespeichert. Wenn ein neuer Bestwert erreicht wurde, wurden die kontinuierlichen Parameter NEventsMin und Shrinkage Gaußverteilt um die zugehörigen Werte erzeugt. Wie in den Abbildungen 5 (a)-(e) zu erkennen ist, gibt es für einzelne Parameter kaum einen optimalen Wert. Aus diesem Grund haben wir uns für die Parameter entschieden mit denen wir den besten AMS-Wert erreicht hatten. Dies war der Fall bei

- NTrees = 167
- NEventsMin = 7.28244760204
- Shrinkage = 0.0530833026321
- nCuts = 275
- MaxDepth = 9 ,

was auf den Trainingsdaten zu einem AMS-Wert von 0,773 führte.

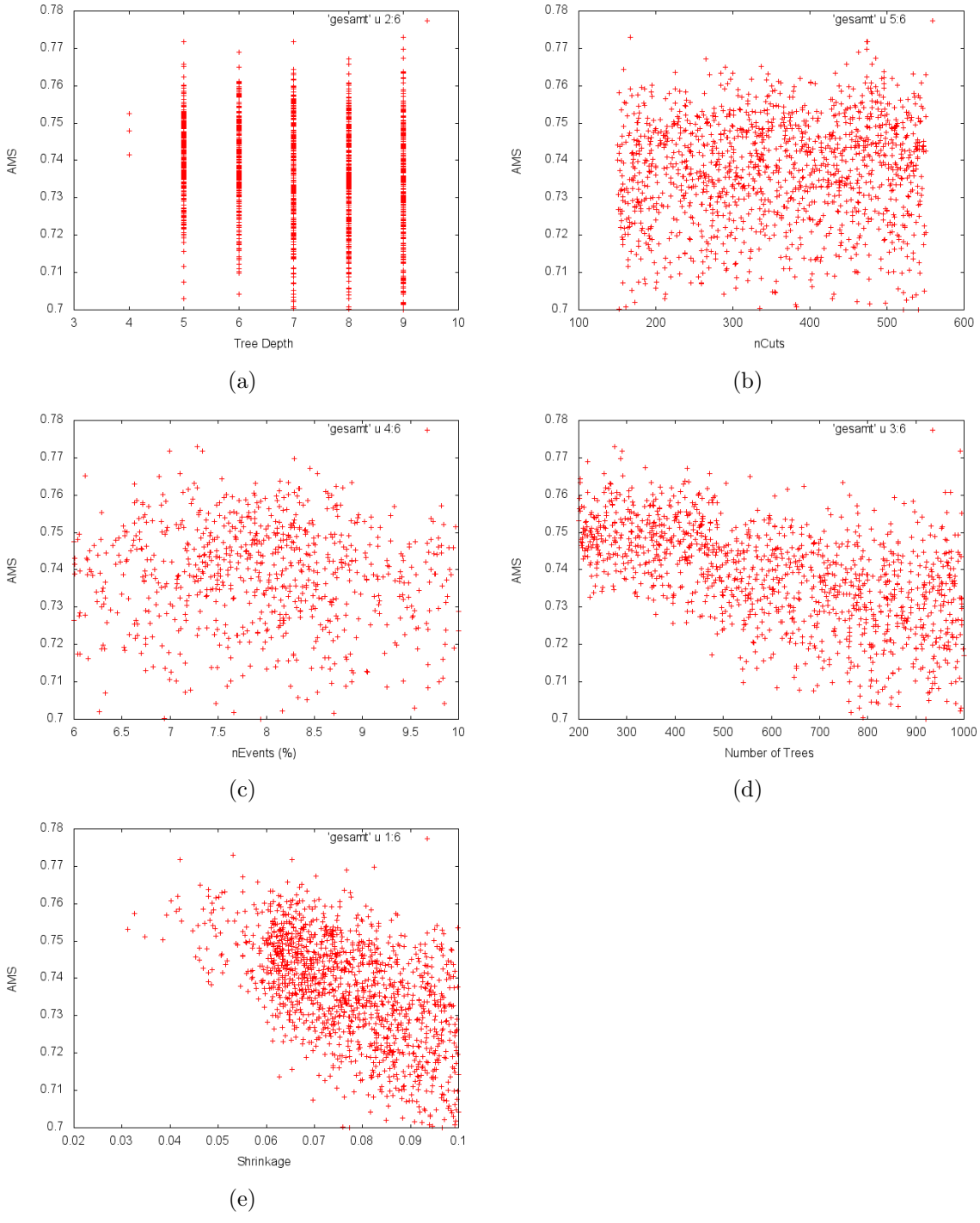
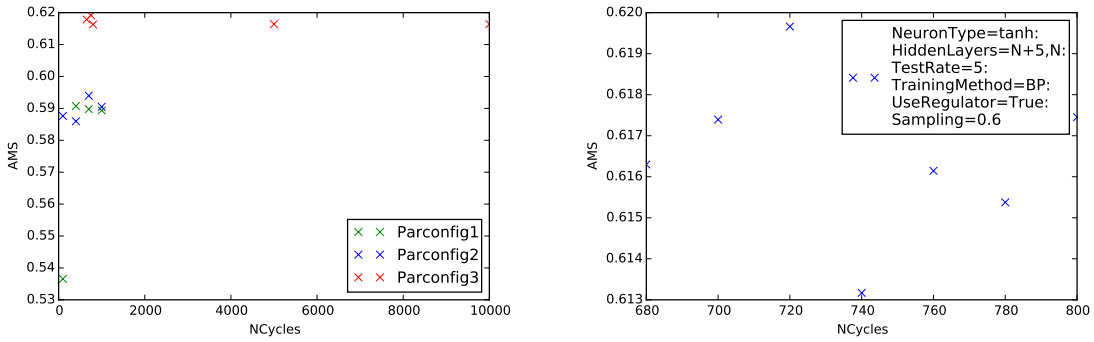


Abbildung 4: AMS-Werte für die Parameter des Boosted Decision Tree als Scatter-Plots. Jeder Punkt entspricht einem Training



(a) (b)

Abbildung 5: AMS-Werte für die Parameter des Neuronalen Netzes. Jeder Punkt entspricht einem Training

7 Optimierung der Parameter des Neuronalen Netzes

Als (künstliches) Neuronales Netz wird im Allgemeinen jede Annsammlung von verbundenen Neuronen bezeichnet, die ein individuelles Ansprechverhalten zu einem gegebenen Input Signal aufweisen. Das Neuronale Netz besteht meist aus mehreren Schichten (HiddenLayers), wobei die erste Schicht Input Neuronen und die letzte die Output Neuronen darstellen. In unserem Fall besteht das Output-layer aus lediglich einem Neuron, da nur zwischen Signal und Background unterschieden werden muss.

Um die Zeit zu verringern, die das NN benötigt um die Daten zu trainieren, wurde teilweise mit der Option Sampling=0.X gearbeitet, welche nur den X-ten Anteil der Trainings- und Testdatensätze verwendet. Zwar muss man dadurch einen etwas schlechteren AMS-Wert in Kauf nehmen, die Laufzeit verringert sich jedoch enorm. Das Sampling wurde benutzt um den Parameter NCycles und die Anzahgl der Hiddenlayers zu optimieren. Eine Variation dieser Paramter hatte allerdings eine recht marginale Auswirkung auf den AMS, wie sich im Folgenden zeigen wird. Außerdem konnte aufgrund der langen Laufzeit nicht wahllos beliebig viele Parameterkonfigurationen ausprobiert werden.

Zunächst wurde bei den festen Parametern NCycles = 100,400,700,1000 der Parameter HiddenLayer=N+5 zu HiddenLayer=N+5,N variiert was dem Einfügen einer weiteren Schicht mit N Neuronen in das Netz entspricht (siehe 5a, Parconfig wird in der Fußnote beschrieben[1]).
Es stellte sich heraus, dass die N+5,N layer Architektur minimal besser abschnitt und diese nun in den folgenden trainings verwendet wird.

[1]
Parconfig1=NeuronType=tanh:HiddenLayers=N+5:TestRate=5:TrainingMethod=BP:UseRegulator=True:Sampling=0.3:
Parconfig2=NeuronType=tanh:HiddenLayers=N+5,N:TestRate=5:TrainingMethod=BP:UseRegulator=True:Sampling=0.3
Parconfig3=NeuronType=tanh:HiddenLayers=N+5,N:TestRate=5:TrainingMethod=BP:UseRegulator=True:Sampling=0.6

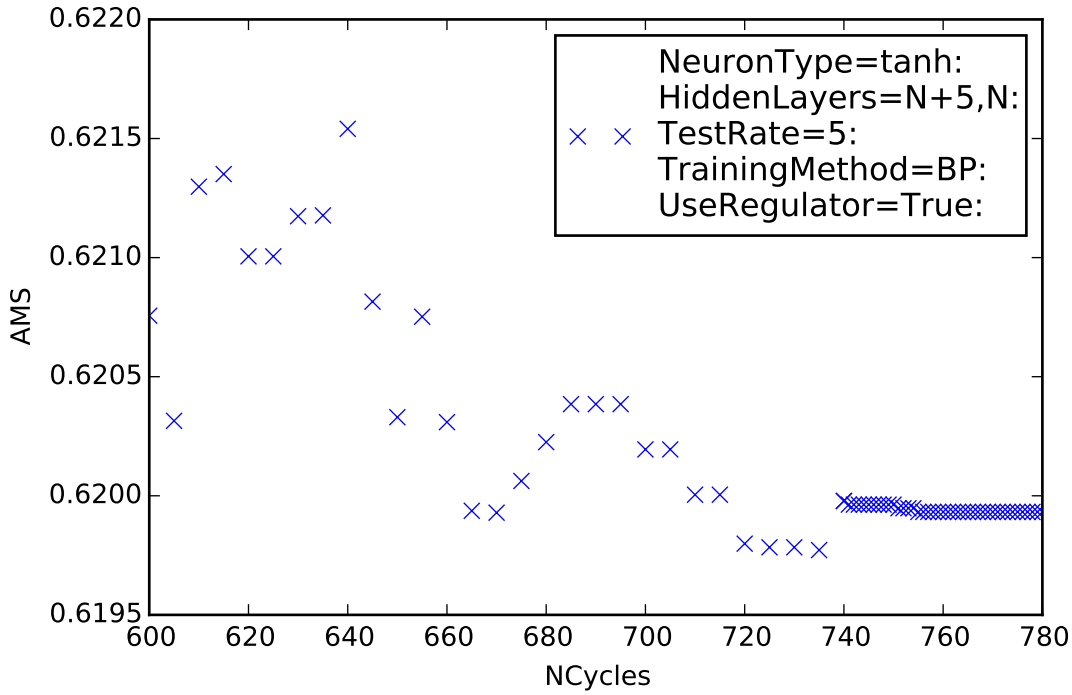


Abbildung 6: AMS-Werte über NCycles

Um das Verhalten bei großen Anzahlen von Zyklendurchläufen zu untersuchen wurde zusätzlich bei einem Sampling=0.6 die Cycles=650,750,800,5000,10000 ausgewertet (siehe 5a, rote Punkte). Eine signifikante Erhöhung der Cycle Anzahl hatte zwar eine rießige Laufzeitverlängerung zur Auswirkung, aber keine Verbesserung des AMS; im Gegenteil dieser hatte sich sogar minimal verschlechtert.

Im Folgenden wurde der Bereich um NCycles = 700 näher untersucht, da vermutet wurde, dass dort der AMS am besten ist. Hierzu wurden die NCycles=800,780,760,740,720,700,680 mit einem Sampling=0.6 simuliert (siehe 5b). Da diese Abbildung noch keinen Schlussfolgerungen zulässt wurde in dem darauffolgenden Schritt die Schrittgröße der NCycles verringert und erneut ausgewertet (dieses Mal unter Verwendung des vollen Trainings- und Testdatensatzes, Sampling=1) (siehe 6). Es stellte sich heraus, dass der AMS bei NCycles≈640 den größten Wert von 0.6215 annimmt. Da dieser Wert jedoch bei Weitem nicht mit dem des BDT mithalten kann, wurde das NN als Methode verworfen.

8 Fazit und unser Score

Für die Analyse des kompletten Datensatzes verwendeten wir die BDT-Methode mit dem Parametersatz, der die besten Trainingswerte erreichte. (siehe Abschnitt **??**) Die Eingabevariablen wurden nicht transformiert und der volle Variablensatz wurde verwendet. Der erreichte AMS-Wert liegt bei

$$2.54 \tag{8.1}$$

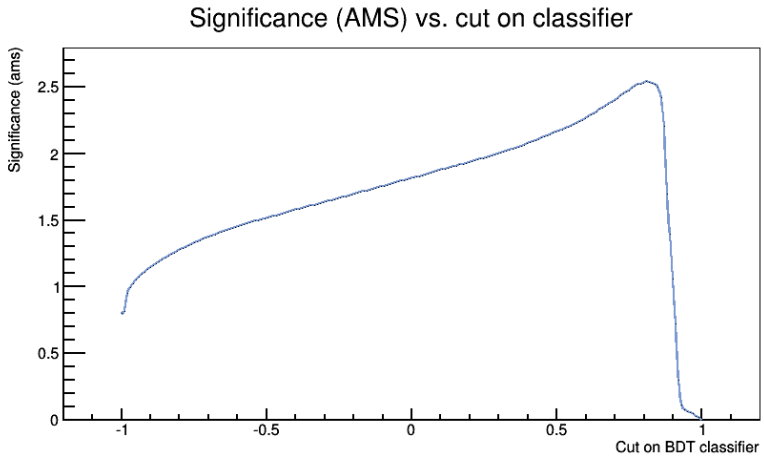


Abbildung 7