Lehrstuhlversuch im SS2020

Datenanalyse mit IceCube-Monte-Carlo-Simulationsdaten

Fabian Koch
fabian3.koch@tu-dortmund.de
Nils Breer
nils.breer@tu-dortmund.de
Nicole Schulte
nicole.schulte@tu-dortmund.de

Abgabe: xx.xx.2020

TU Dortmund – Fakultät Physik

Inhaltsverzeichnis

1 Theoretische Grundlagen

Kosmische Strahlung besteht hautpsächlich aus hochenergetischen Protonen, schweren Kernen, Myonen und Neutrinos. Die Komposition der geladenen kosmischen Strahlung hängt dabei vom Energiebereich ab. Es können dabei Energien bis zu 10^{20}eV erreicht werden. Die Energieverteilung folgt approximal einem Potenzgesetz der Form

$$\frac{\mathrm{d}\Phi}{\mathrm{d}E} = \Phi_0 E^{\gamma}$$

wobei γ der spektrale Index von etwa -2.7 für geladene Teilchen ist. Astrophysikalische Neutrinos stammen aus Quellen die auch Hadronen beschleunigen. Da Neutrinos einen sehr kleinen Wirkungsquerschnitt besitzen, durchdringen sie selbst dichte Staubwolken, welche für Photen ein Hindernis sind. Außerdem werden Neutrinos nicht durch Magnetfelder abgelenkt und zeigen somit auf die Quelle und könne so Informationen über das innere der Quelle liefern. Aufgrund von galaktischen Magnetfeldern ist es aber bislang nicht gelungen die Quellen der kosmischen Strahlung zu bestimmen. Wenn zur Beschleunigung eine Art Stoßbeschleunigung angenommen wird, also eine Art Fermibeschleunigung für Neutrinos, führt dies auf ein Potenzgesetz für den Neutrinofluss mit spektralen Index von $\gamma \approx -2$. Nun können Neutrinos und Myonen auch aus Wechselwirkungen in der Atmosphäre stammen, wo sie Zerfallsprodukte von Pionen und Kaonen sind. Da diese Mesonen eine vergleichsweise lange Lebensdauer haben verlieren sie vor ihrem Zerfall schon an Energie wodurch sich das Energiespektum einem Potenzgesetz proportional zu $E^{-3.7}$ gleicht. Die so entstandenen Neutrinos und Myonen nennt man konventionelle Neutrinos bzw. Myonen. Andererseits gibt es sogenannte prompte Neutrinos, welche entstehen wenn in hochenergetischen Wechselwirkungen kurzlebige schwere Hadronen erzeugt werden und ohne nennenswerten Energieverlust wieder zerfallen. Aus den (semi-)leptonischen Zerfällen stammen Neutrinos und Myonen welche das Energoespektrum der kosmischen Strahlung erben.

1.1 Messung von Neutrinos mit IceCube

2 Das IceCube-Experiment

Der IceCube Detektor dient der Detektion von hochenergetischen Neutrinos und Myonen und besteht aus drei Komponenten. Dem IceTop Detektor, dem In-Ice Detektor welche die größte Komponente ist und dem DeepCore. Das Experiment befindet aich am geographischen Südpol und die die Hauptdetektionsschicht ist zwischen $1450\,\mathrm{m} - 2450\,\mathrm{m}$ in einer klaren Eisschicht. Mit Hilfe von Cherenkov Licht, welches mit 5160 DOMs[1] an 86 strings detektiert werden kann, lassen sich hochenergetische geladene Teilchen detektieren. Ein schematischer Aufbau ist in Abbildung 1 gegeben.
Das DeepCore besteht aus sieben Kabeln, welche sich im Zentrum des In-Ice Detektors befinden, lässt sich die Energieauflösung dort auf 10 GeV senken gegenüber den 100 GeV des In-Ice Detektors. Das IceTop dient als Luftschauer Experiment welches Cherenkov

[1] digital optical modules

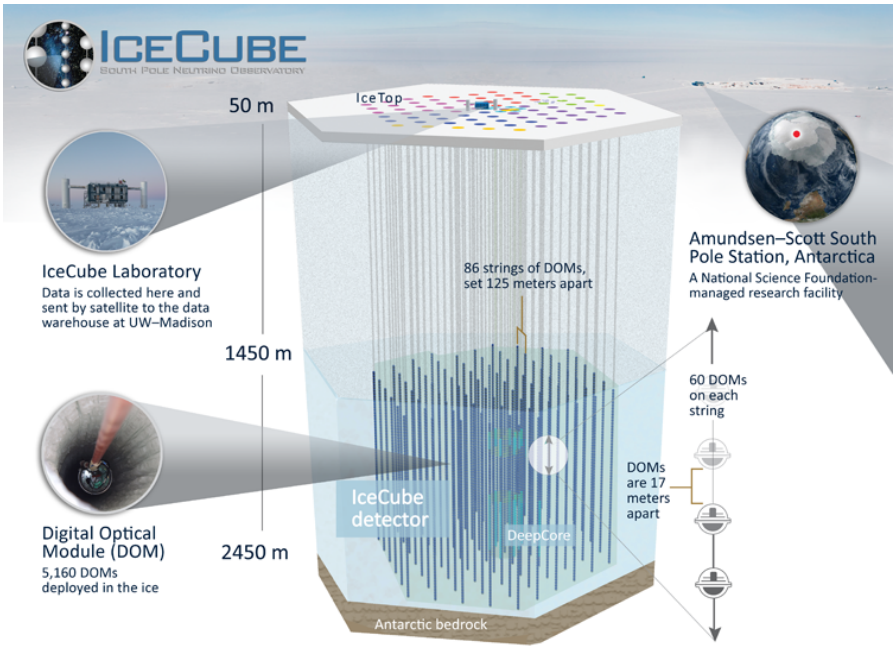

Abbildung 1: Schematischer Aufbau des IceCube Experiments.

Licht in lichtdichten detektiert. Außerdem kann es als Vetoregion für das In-Ice verwendet werden um gewisse Winkelbereiche auszuschließen.
Die Prozesse zur Detektion von Neutrinos geschieht über Sekundärteilchen aus den Wechselwirkungen mit den Kernen im Eis als geladener Strom

$$\nu_l(\bar{\nu}_l) + A \to l^{\mp} + X$$

oder neutraler Strom

$$\nu_l + A \to \nu_l + X\,.$$

Hierbei verursachen Elektronen eine sphärischen Schauer aufgrund des rapiden Energieverlustes. Myonen hingegen haben einen eher langsamen Energieverlust und können größere Distanzen überwinden und haben eine lange "Lichtspur" als Signatur. Tau-Leptonen haben eine ähnliche Signatur wie Elektronen aufgrund ihrer geringen Lebensdauer. Myonen erzeugen zu wenig Cherenkov Licht um detektiert zu werden aber sie generieren Photonen und e^+e^- Paare im Medium, welche selbst wieder schauern und weitere Elektron-Positron Paare erzeugen und von den PMT[2] aufgesammelt wird.

3 Auswertung

Zu Beginn wurden aus den zur Vefügung gestellten Datensätzen alle Features die Monte Carlo Daten, Gewichte und Labels enthalten, entfernt. Außerdem wurden die Spalten

[2]Photomultipliern

entfernt, die ausschließlich den selben Wert, "Inf" oder "not a Number" enthalten. Abschließend wurde Signal und Hintergrund Samples auf Features überprüft, die nur in einem der beiden Samples auftreten und diese ebenfalls entfernt, sodass beide Samples die selben Features enthalten. Um später zwischen Hintergrund und Signal unterscheiden zu können, wurde an das Signal mit "0" gelabelt und der Hintergrund mit "1".
Im folgendenwerden wir drei verschieden Klassifizierer auf eine Auswahl an Features testen. Es werden der `Naive-Bayes` Klassifizierer, der `RandomForestClassifier` und der `KNeighborsClassifier` mit Hilfe von `sklearn` verwendet. Vorab wurden mittels der `SelectKBest` Methode die 20 besten features ermittelt. Die Güte der Features wurde mit dem `f_classif` ermittelt. Diese Features sind der beigefügten pdf des verwendeten jupyter Notebooks zu entnehmen. Aus diesen Features wurden Test- und Trainingsdatensätze extrahiert mit welchen die obigen Klassifizierer getestet nun werden.
Zuerst wurde der `RandomForestClassifier` verwendet. Dieser basiert auf den binären Entscheidungsbäumen. Bei einem Entscheidungsbaum wird an jedem Knoten ein Schnitt in einer Variable durchgeführt und die daraus entstandenen Teilmengen werden in den beiden Ästen des Knotens wiederum durch Schnitte unterteilt, bis entweder eine bestimmte Tiefe des Baumes erreicht ist oder die Blätter nur Ereignisse einer Klasse enthalten. Um die Effekte des Übertrainierens zu minimieren, wird über ein Ensemble unterschiedlicher Entscheidungsbäume gemittelt (Siehe Anleitung). Dafür wurde ein Wald mit 100 Bäumen gewählt. Alle anderen Parameter wurde mit default initialisiert. Mit der Vorhersage des Klassifizierers wurden die Effizienz[3] und die Reinheit("precision") bestimmt. Außerdem wurde der Jaccard Score für unsere obige Attributsauswahl berechnet. Die Entsprechende ROC-Kurve ist Abbildung 2 zu entnehmen. Die Ergebnisse stehen in Tabelle 1.
Für den `KNeighborsClassifier` wurden 20 Nachbarn gewählt. Auch hier wird die Vorhersage und der Jaccard Score in Tabelle 1 dargestellt sowie die ROC-Kurve in Abbildung 2.
Zuletzt wurde der Naive-Bayes Klassifizierer, welcher dem Prinzip der bedingten Wahrscheinlichkeiten genügt, getestet. Die Ergebnisse befinden sich wieder in Tabelle 1 und der Abbildung 2.

Klassifizierer	Effizienz	Reinheit	Jaccard-Score
RandomForest	0.93425	0.92247	0.87762
KNeighborsClassifier	0.87913	0.87376	0.78463
Naive-Bayes	0.79763	0.75398	0.68410

Tabelle 1: Effizienz und Reinheit des RandomForestClassifiers.

[3]wir haben den "accuracy score" von sklearn verwendet da es keinen anderen gab

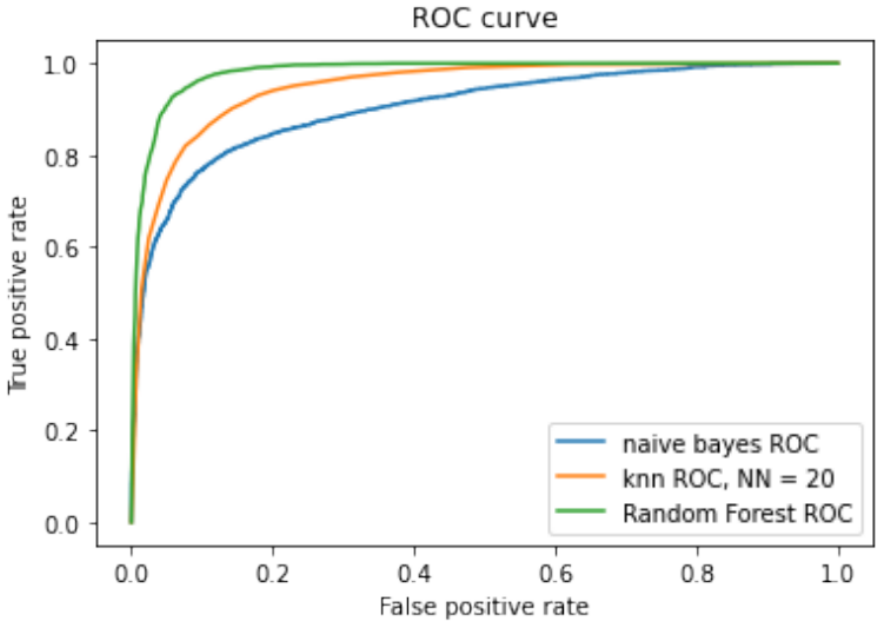

Abbildung 2: ROC-Kurven aller getesteten lassifizierer.

4 Diskussion

Im allgemeinen ist festzuhalten, dass der Naive-Bayes Lerner die schlechtes Effizienz der drei Klassifizierer hat. Einerseits ist der Naive-Bayes Lerner relativ schnell und trifft gute Vorhersagen, wenn die Attribute unabhängig von einander sind. Außerdem funktioniert er am besten mit kategorischen Attributen und weniger gut mit den hier verwendeten numerischen Eingaben. Außerdem wird der Naive-bayes Lerner auch "schlechter Schätzer" genannt, weswegen der Output von Methoden wie `predict_proba` mit Bedacht verwendet werden sollten.
Der kNN-Klassifizierer, welcher auch als "Lazy lerner" bekannt ist gehört zu der Familie des überwachten maschinellen Lernens und wird oft als Benchmark(übersetzen bitte) für komplexere Lerner wie SVMs verwendet. Der kNN ist relativ schnell für wenige Attribute doch leidet sehr unter dem Fluch der Dimensionalität. Wir haben hier 20 Attribute verwendet und nehmen auch die 20 nächsten Nachbarn, as den Algorithmus sehr langsam macht und die Vorhersage mit zunehmender Attributszahl immer schlechter wird. Außerdem sollten die Attribute die selbe Größenordnung besitzen da unsere Abstandsbestimmung mit der euklidischen Norm berechnet wurde. Um den Algorithmus noch effizienter zu machen sollte man hier die Daten vor dem Training normieren. Dennoch ist der kNN immer noch besser als der Naive-Bayes Lerner.
Der `RandomForestClassifier` ist auf unserem Sample der beste Klassifizierer. Er kann Bäume dekorrelieren um eben Korrelationen in den Attributen zu kontrollieren. Außerdem sind die Fehler vergleichsweise klein, da der Random Forest den Output jedes Baums verwendet und so Abweichungen minimiert. Hier könnte man die Effizienz noch

verbessern indem eine andere Feature Selection verwendet wird. Zum Beispile könnte eine Hauptkomponentenanalyse verwendet werden um die besten Attribute zu finden. Man könnte auch mehr oder weniger Attribute benutzen und die Effizienzen neu berechnen um eventuell Overfitting Effekte zu finden falls bestimmte Attribute die anfangs als wichtig gelabelt wurden trotzdem die Effizienz verringern.

5 Anhang

Literatur

[1] IceCube Collaboration, *The IceCube Neutrino Observatory: instrumentation and online systems*, Journal of Instrumentation, JINST 12 P03012 (2017)

[2] IceCube Collaboration, Detektor, https://icecube.wisc.edu/science/icecube/detector