Trabalho 6 - Física de Partículas

Nuno Teixeira - 75494 -https://www.kaggle.com/nteixeira

Instituto Superior Técnico -Tópicos Avançados de Física Computacional

11 de Dezembro de 2018

1 Resolução

1.1

Das 30 features em estudo, existem alguma mais interessantes do que outras. Se olharmos para a tabela, vemos que por exemplo a feature número 30 tem pouca relevância para o problema porque contém em quase todas as entradas valores a 999 com pouca variação. As features, foram renormalizadas segundo,

$$z = \frac{(x - \mu)}{\sigma} \tag{1}$$

Em que x é a feature, μ, z a nova feature renormalizada e σ o desvio padrão da distribuição Abaixo estão alguns gráficos que representam as features mais importantes, embora não seja muito explicito, primeiro devido ao facto da quantidade relativa de cada classe estar pesada pelos "weights"e depois pelo facto dos datasets serem grandes. Devido à interface do TMVA, foi possível extrair as features mais interessantes para o problema às quais disponho a combinação das quatro features mais valiosas.

Output do TMVA usando o BDGT com score de 3.53 no treino

```
                          (...)
 Ranking input variables (method specific)...
BDTG                     : Ranking result (top variable is best ranked)
                         : ----------------------------------------------------------
                         : Rank : Variable                     : Variable Importance
                         : ----------------------------------------------------------
                         :    1 : DER_deltar_tau_lep           : 6.799e-02
                         :    2 : DER_mass_transverse_met_lep  : 5.513e-02
                         :    3 : PRI_tau_eta                  : 5.485e-02
                         :    4 : PRI_lep_eta                  : 5.307e-02
                         :    5 : DER_mass_MMC                 : 5.263e-02

                          (...)
```

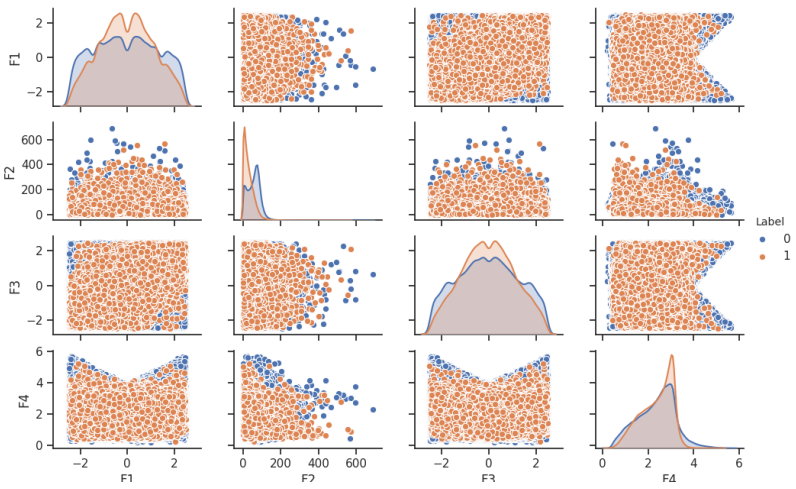

Figura 1: Top 4 das features mais relevantes para o problema(F1, F2 ,F3 ,F4). são as features mais valiosas. Existem algumas regiões onde se consegue distinguir de uma forma mais visível o background e o sinal, mas essa separação não é trivial.

1.2

Este problema é um problema único e invulgar para machine learning. O propósito trata-se de maximizar a função objectiva AMS sendo a expressão:

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

em que neste problema $b_{reg} = 10$ é o background de regularização. Esta função é um pouco mais complicada do que as funções objectivas habituais dificultando a abordagem ao problema. A razão pela qual a função $AMS_2 = s/\sqrt{b}$ não é útil é porque como temos muito mais background do que sinal, esta função não tem grande dinâmica para avaliar o máximo ao passo que a da expressão em Eq. (1) contempla a elevada diferença entre sinal e fundo necessária para uma análise minima de classificação. Nesta competição é importante haver dois sectores, um privado e um público as soluções que são geradas de um sector privado irão ser as com os algoritmos mais diferentes e no qual respostas serão mais diversas. O sector público serve para melhorar a abordagem ao problema para certificar que a melhor optimização do problema é conseguida relativamente aos hiperparametros e ao algoritmo para melhorar o output e minimizar o erro

1.3

Os métodos usados foram o discriminante Fisher e uma Boosted Gradient Decision Tree. O método do discriminante de Fisher.

O método de Fisher é um discriminador linear aplicado em transformações de variáveis sem correlação linear ao distinguir os valores médios das distribuições do sinal e do fundo. A classificação em sinal e fundo depende das seguintes características: a média $\bar{x_k}$ da variável de input, a média $x_{\bar{s(b)},k}$ da classe e a matriz de covariância total C. Esta matriz pode ser decomposta em outras duas, a matriz "within"W e a matriz "between"B.

$$W_{k,l} = \frac{1}{2}\sum_{U=s,b} =< x_{U,k} - x_{\bar{U},k} >< x_{U,l} - x_{\bar{U},l} > \tag{3}$$

$$B_{k,l} = \frac{1}{2}\sum_{U=s,b} =< x_{\bar{U},k} - x_{\bar{U},k} >< x_{\bar{U},l} - x_{\bar{U},l} > \tag{4}$$

Os coeficientes de Fisher são dados por,

$$F_k = \frac{\sqrt{N_s N_b}}{N_s + N_b}\sum_{k=1}^{n_{var}} W_{kl}^{-1}(x_{s,l}^{-} - x_{b,l}^{-}) \tag{5}$$

O discriminante de Fisher é dado por:

$$y_{Fi} = F_0 + \sum_{k=1}^{n_{var}} F_k x_k(i) \tag{6}$$

Em que F_0 é um offset que deixa a média y_{Fi}^{-} a 0.

Uma árvore de decisão (decision tree) é uma estrutura de classificação modo árvore. Repetidos critérios booleanos são aplicados até se chegar a uma condição de paragem. O espaço de fases é continuamente separado em sinal e background, dependendo dos casos com que acabamos o treino de acordo com os nodos das folhas. Na fase de treino, a árvore de decisão é desenvolvida de tal

maneira que começando na raiz, começamos a dividir a árvore a após satisfazer um critério de divisão. Este processo é repetido até a árvore estar totalmente desenvolvida. Em cada nodo, a divisão é obtida encontrando a variável e corte respectivo que proporciona a melhor separação entre sinal e fundo. De modo a evitar overfitting deve-se cortar a árvore para torna-la o mais simples possível.

Ao aplicar este método com momento, neste caso, Gradient Boost, estamos a certificarnos que a árvore é robusta face a fluctuações e desta forma a assegurar a estabilidade. Tendo o nosso output, y e $F(x)$ o nosso modelo, podemos definir a função de perda (loss function).

$$L(F, y) = (F(x) - y)^2 \tag{7}$$

Para o caso de Gradient Boost a função de perda é dada por

$$L(F, y) = \log(1 + e^{-2F(x)y}), \tag{8}$$

em que para encontrar o mínimo desta função recorremos ao gradiente. O treino foi feito de modo ao programa procurar o threshold no treino ao qual a função AMS, Eq. (1.2) é maximizada em que o maximizante é o score do algoritmo e considerar esse como nosso modelo para aplicar no test set.

1.4

Para o discriminante de Fisher, foram aplicados os seguintes parametros ver [1]

```
method=factory.BookMethod( dataloader, TMVA.Types.kFisher, "Fisher",

"H:!V:Fisher:VarTransform=None:CreateMVAPdfs:
PDFInterpolMVAPdf=Spline2:NbinsMVAPdf=50:NsmoothMVAPdf=10" );
```

Obteve-se um score de 2.09055, abaixo está disposta a curva AMS obtida relativa ao score de fisher

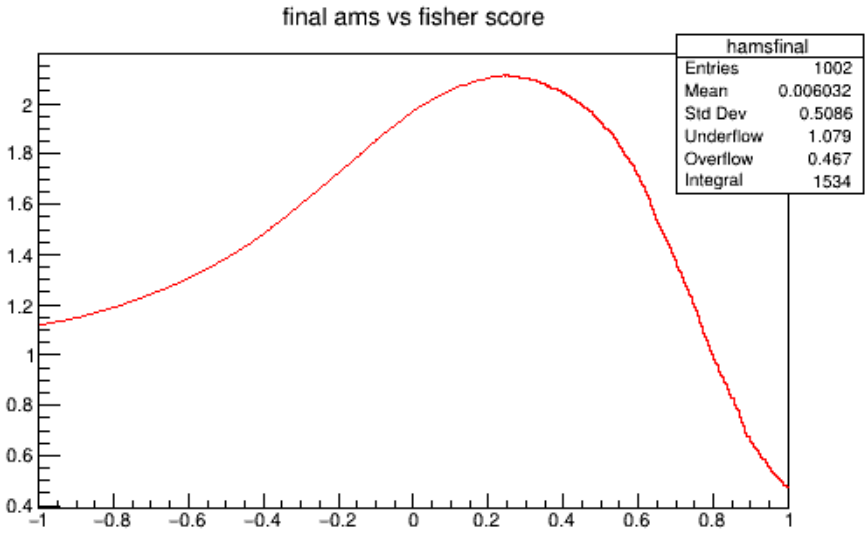

Figura 2: Curva AMS para para o algoritmo Fisher. Obteve-se um score de treino de 2.13 e um threshold para o score de 0.3

As variações aos parametros foram feitas no algoritmo de BDTG. Foi usada a biblioteca TMVA do ROOT para aplicar a árvore de decisão. Em ambos os casos foi aplicado o método de gradiente estocástico. Para o primeiro caso obteve-se um score de 3.13 para os seguintes parametros

```
method =  factory.BookMethod( dataloader, TMVA.Types.kBDT, "BDTG",
"!H:!V:NTrees=500:MinNodeSize=5%:BoostType=Grad:Shrinkage=1:
UseBaggedBoost:BaggedSampleFraction=0.5:nCuts=30:
MaxDepth=2:SeparationType=MisClassificationError")
```

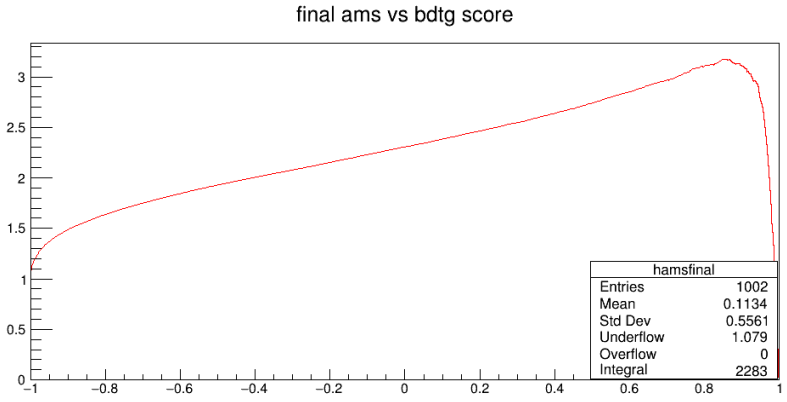

Figura 3: Curva AMS para o primeiro caso do algoritmo de gradiente estocástico. Obtevese um score de threshold de 0.83 para a BDGT. No treino obteve-se uma pontuação ligeiramente superior ao score do teste, o que era de esperar, mas os dados estão de acordo com o modelo.

No segundo caso, experimentou-se valores mais sensatos no que toca ao shrinkage, aumentou-se o número de árvores e possibilitou-se o corte das árvores para evitar o overfitting e seguiu-se uma abordagem de variação mais de acordo com [1] e obteve-se um score de 3.43.

```
method =  factory.BookMethod( dataloader, TMVA.Types.kBDT,
"BDTG", "!H:!V:NTrees=800:MinNodeSize=5%:BoostType=Grad:Shrinkage=0.15:
UseBaggedBoost:BaggedSampleFraction=0.7:nCuts=-1:MaxDepth=2:
SeparationType=MisClassificationError
:PruneMethod=CostComplexity:UseBaggedGrad=True")
```

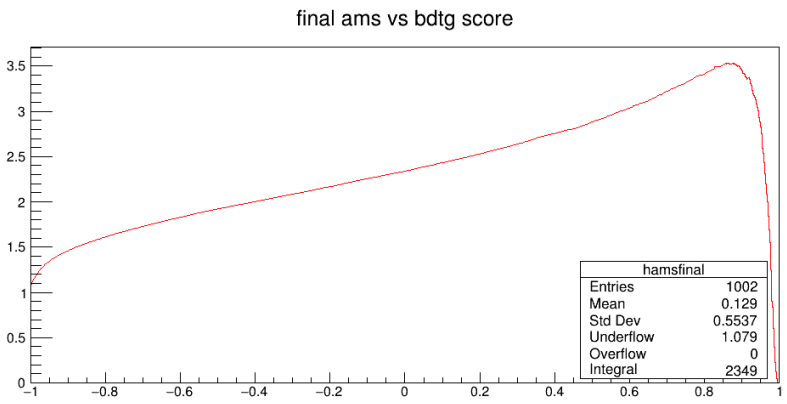

Figura 4: Curva de AMS para a segunda calibração de de BDTG. Obteve-se um máximo no treino, superior a 3.5 mas no teste, este score foi ligeiramente inferior, 3.43

Referências

[1] https://root.cern.ch/download/doc/tmva/TMVAUsersGuide.pdf