TMVA

Dr. Öğr. Üyesi Uygar ŞAŞMAZ

Biruni Üniversitesi
Mühendislik ve Doğa Bilimleri Fakültesi
Bilgisayar Mühendisliği (İngilizce)
usasmaz@biruni.edu.tr

TMVA

- Yüksek enerji fiziğinde, giderek daha büyük veri kümelerinde daha küçük sinyaller aranmasıyla, verilerden mümkün olan en fazla bilgiyi çıkarmak zorunlu hale gelmiştir.
- Makine öğrenimi tekniklerine dayalı çok değişkenli sınıflandırma yöntemleri, çoğu analizin temel bir bileşeni haline gelmiştir.
- ROOT'a entegre edilen TMVA, çok çeşitli çok değişkenli sınıflandırma (veya bağlanım) algoritmalarını barındıran bir kütüphanedir.
- Tüm mevcut sınıflandırıcıların eğitimi, testi, performans değerlendirmesi ve uygulaması, kullanıcı dostu arayüzler aracılığıyla eş zamanlı olarak gerçekleştirilir.

TMVA

- Çok Değişkenli Analiz Araç Takımı (TMVA), çok değişkenli sınıflandırma ve bağlanım tekniklerinin işlenmesi, paralel değerlendirilmesi ve uygulanması için ROOT ile entegre bir ortam sağlar.
- İstenilen çıktının bilindiği eğitim olaylarından yararlanarak, bir karar sınırı (sınıflandırma) veya hedef değeri tanımlayan bir yaklaşımı (bağlanım) belirlerler.
- TMVA özellikle yüksek enerji fiziği (HEP) uygulamalarının ihtiyaçları için tasarlanmıştır, ancak bunlarla sınırlı değildir.

Kütüphane İçeriği

- Rectangular cut optimisation (binary splits).
- Projective likelihood estimation.
- Multi-dimensional likelihood estimation (PDE range-search, PDE-Foam k-NN.
- Linear and nonlinear discriminant analysis (H-Matrix, Fisher, LD, FDA).
- Artificial neural networks (three different multilayer perceptron implementations).
- Support vector machine.
- Boosted/bagged decision trees.
- Predictive learning via rule ensembles (RuleFit).
- A generic boost classifier allowing one to boost any of the above classifiers.
- A generic category classifier allowing one to split the training data into disjoint categories with independent MVAs.

Sınıflandırma

- Bir sınıflandırma problemi daha genel anlamda ayrıklaştırılmış bir bağlanım problemine karşılık gelir.
- Bir bağlanım, bir fonksiyonun parametre değerlerini tahmin eden ve bir yanıt değişkeninin (veya vektörün) değerini diğer değişkenlerin (giriş değişkenleri) değerleri açısından öngören bir süreçtir.
- Girdi ve hedef değişkenleri tanımlama, çok değişkenli yöntemlerin kayıt edilmesi, eğitimleri ve testleri için arayüz, sınıflandırma problemlerindeki yazım şekline çok benzerdir.
- Kullanıcı ve TMVA arasındaki iletişim, "Factory" ve "Reader" sınıfları aracılığıyla kolayca ilerler.

MVA Factory

- TMVA, MVA yöntemleri arasında nesnel bir performans karşılaştırması sağlar.
- Tüm yöntemler aynı eğitim ve test verilerini görür ve aynı yürütme işi içinde değerlendirilir.
- Bir Factory sınıfı, kullanıcı ile TMVA analiz adımları arasındaki etkileşimi düzenler.

MVA Reader

- Reader sınıfı kullanılarak, herhangi bir C++, ROOT makro veya python dosyası ağırlık dosyalarını okuyabilir.
- Eğitilmiş MVA yönteminin bağımsız kullanımı için TMVA ayrıca C++ sınıfları içeren kütüphaneler üretir (tüm yöntemler için mevcut değildir), bunlar ağırlık dosyalarından kodlanmış bilgileri içerir.
- Böylece artık ağırlık dosyalarını kullanmak için TMVA'ya, ROOT'a veya başka herhangi bir harici kütüphaneye gerek kalmaz.

Performanslar

- Sınıflandırıcıların sinyal verimliliğini ve artalan reddetme (Signal efficiency - background rejection) performansını veya bağlanım hedefi ile tahmin arasındaki ortalama varyansı karşılaştırmak için analizler, bazı kıyaslama değerleri için tablolanmış sonuçları yazdırır.
- Böylece kullanılan yöntemler arasında kıyaslama yapılabilir, veri seti için en iyi yöntemin hangisi olduğu belirlenebilir.
- Ayrıca, eğitim, test ve değerlendirme aşamaları sırasında edinilen çeşitli grafiksel bilgiler bir ROOT çıktı dosyasında saklanır.
- Bu sonuçlar, TMVA arayüzü üzerinden kolayca yürütülen makrolar kullanılarak görüntülenebilir.

Ağırlık dosyaları

- TMVA eğitim işi alternatif olarak bir ROOT betiği, bağımsız bir yürütülebilir dosya veya PyROOT arayüzü üzerinden bir python betiği olarak çalışır.
- Bu uygulamalardan birinde eğitilen her MVA yöntemi kendi yapılandırmasını XML biçimine sahip bir sonuç (weights) dosyasına yazar.

TMVA paketi

- TMVA Factory eğitim ve

  TMVA Reader uygulama örnek dosyalarına

  ROOT klasöründen ve web üzerinden erişilebilir:

  $ROOTSYS/tutorials/tmva/

  https://root.cern/manual/tmva/

- Eğitim örnek dosyaları:

  TMVAClassification.C, TMVARegression.C

- Uygulama örnek dosyaları:

  TMVAClassificationApplication.C, TMVARegressionApplication.C

Eğitim öncesi kümelerin belirlenmesi

- Eğitime başlamadan önce girdi ve çıktıların belirlenmesi gerekir.
- Sınıflandırma kullanırken çıktılar Signal/Background şeklinde olurken, girdiler momentum, değişmez kütle, parçacıklar arasındaki açı vb. şeklinde seçilebilir.
- Signal ve Background ayrı TTree yapılarına aktarılmış bir root dosyası kullanılmalıdır.

Eğitim ayarları

- Tanımlama:
  TMVA::Factory *factory = new
  TMVA::Factory("TMVAClassification", outputFile, "<options>");
- Çıktı dosyaları:
  TMVA::DataLoader *dataloader = new
  TMVA::DataLoader("dataset");
- Girdi değişkenleri:
  dataloader->AddVariable("jp"); //momentum
  dataloader->AddVariable("jm"); //mass
  dataloader->AddVariable("jt");   //angle
- Signal / Background Tree:
  dataloader->AddSignalTree(signalTree);
  dataloader->AddBackgroundTree(backgroundTree);

Eğitim ayarları

- TCut mycuts = ""; // mycuts = "var1<0.5 && var2>1";

  TCut mycutb = "";
- dataloader->PrepareTrainingAndTestTree(mycuts, mycutb,

  "SplitMode=Random:NormMode=NumEvents:!V");
- // mycut: Sinyal ve artalan için ek filtre, SplitMode: Train ve Test verilerinin seçim yöntemi, NormMode: Eşit veya farklı sayıda sinyal ve artalan seçimi.

Yöntem ayarları

- Tüm MVA yöntemleri makul varsayılan yapılandırmalarla çalışır ve ortalama uygulamalar için tatmin edici performans gösterecektir.
- Somut bir sorunu çözmek için, tüm yöntemler, maksimum sınıflandırma ve bağlanım yetenekleri için bazı özel ayarlamalar gerektirebilir.
- Sınıflandırıcıların bireysel optimizasyonu ve özelleştirilmesi, bir yöntem dizesi aracılığıyla elde edilir.

Yöntem ayarları

- factory->BookMethod(dataloader,
  TMVA::Types::kTMlpANN, "TMlpANN",
  "!H:!V:NCycles=200:HiddenLayers=N+1,N:LearningMethod=
  BFGS:ValidationFraction=0.3");
- // H: Yardım mesajları, V: Ayrıntılı çıktı, NCycles: Eğitim döngü sayısı, HiddenLayers: Gizli katman sayısı, LearningMethod: Öğrenme yöntemi, ValidationFraction: Eğitim setindeki doğrulama parçası oranı.

Train, Test, Evaluate

- Kullanılmak istenen yöntemler hazırlandıktan sonra seçilen yöntemlerin tümü için eğitim, test ve değerlendirme fonksiyonları çağırılır.

- factory->TrainAllMethods();
- factory->TestAllMethods();
- factory->EvaluateAllMethods();

Örnek dosyayı çalıştırma

- Eğitim dosyası:

  root TMVAClassification.C

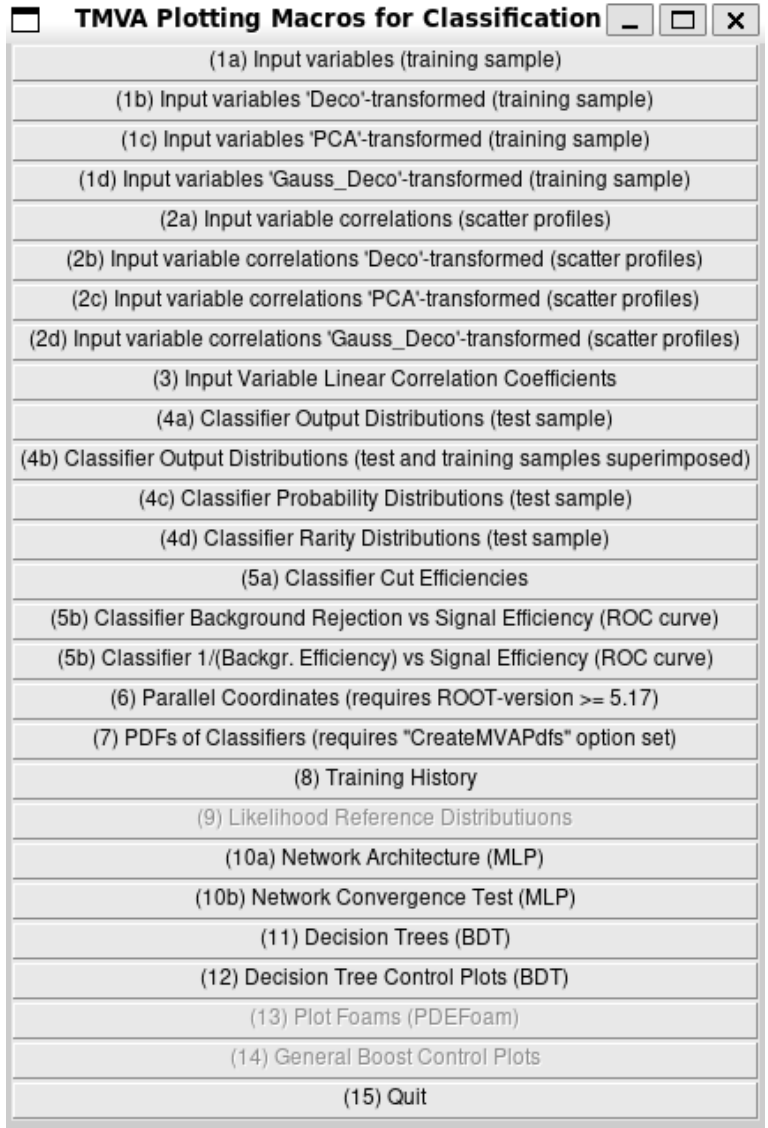
- Eğitim sırasında kullanılacak yöntem(ler) girdi argümanı olarak da gönderilebilir.

  root TMVAClassification.C("Fisher","BDT")

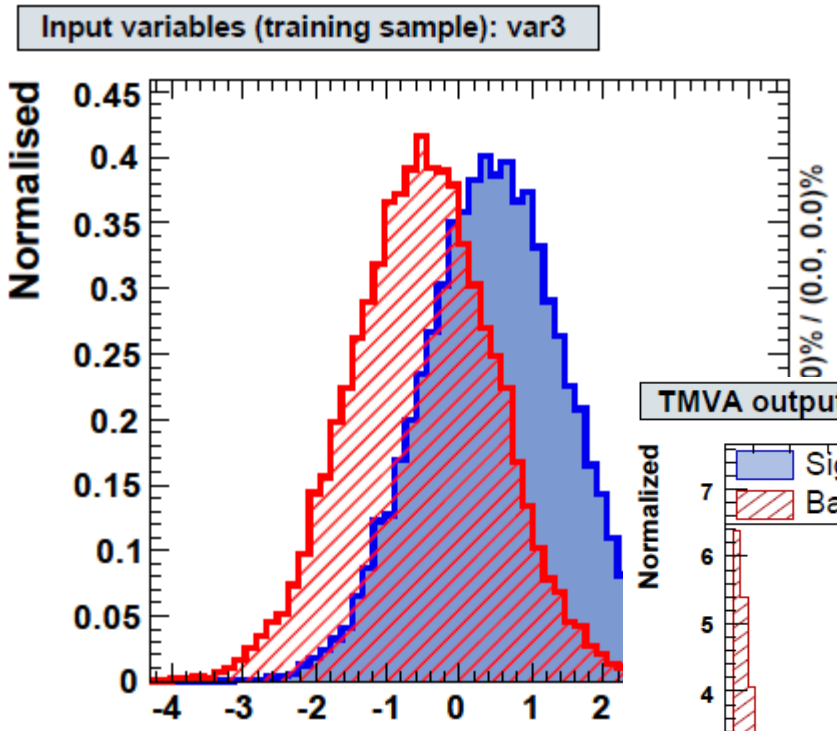
- Eğitim tamamlandıktan sonra sonuçlar ve gerekli parametreler bir root dosyasına yazılır ve arayüz üzerinden grafiklerin oluşturulacağı bir menü açılır.

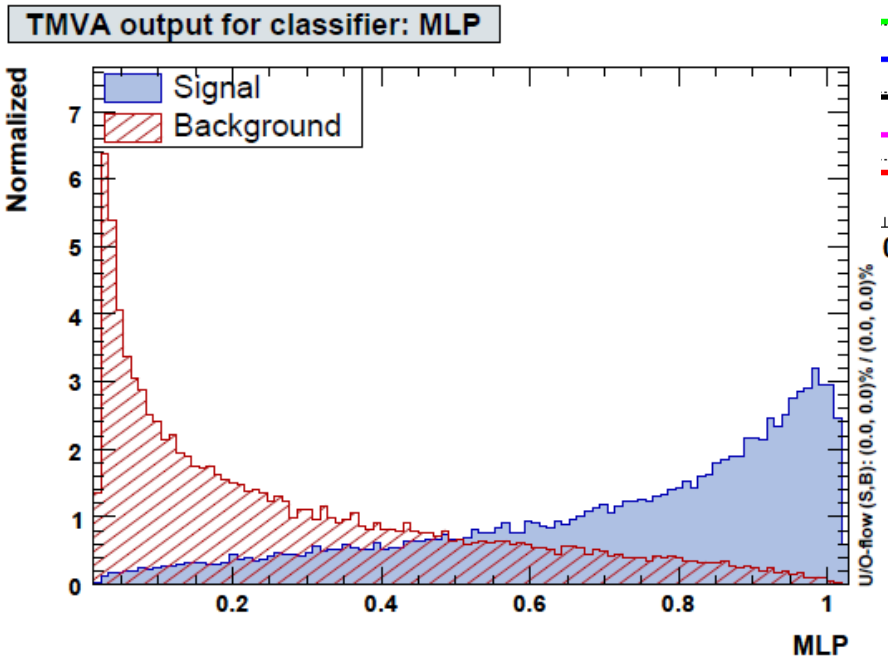
- Batch modunda çalıştırılmış ise, aynı menü çıktı dosyası ile eğitimden sonra açılabilir.

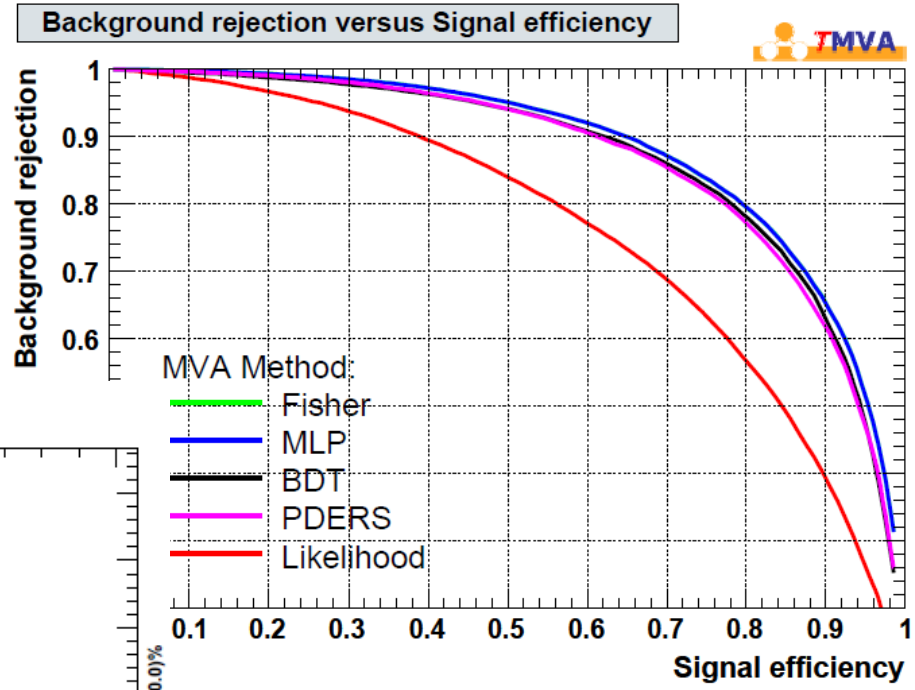
  root -e 'TMVA::TMVAGui("TMVA.root")'



Girdi dağılımları ve eğitim performansları







Ağırlık dosyalarının okunması

- Uygulama dosyası TMVAClassificationApplication.C içeriğinde:
  TMVA::Reader *reader = new TMVA::Reader("!Color:!Silent");
- Float_t jp,jm,jt;
  reader->AddVariable("jp",&jp); //momentum
  reader->AddVariable("jm",&jm); //mass
  reader->AddVariable("jt",&jt); //angle
- TString method = "TMlpANN";
- TString methodName = method + TString(" method");
  TString weightfile = TString ("dataset/weights/TMVAClassification_")
  + method + TString (".weights.xml");
- reader->BookMVA(methodName, weightfile);
- Float_t tmvaout = reader->EvaluateMVA(methodName);

Referanslar

1. A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, and H. Voss, TMVA - Toolkit for Multivariate Data Analysis, PoS ACAT 040 (2007), arXiv:physics/0703039
2. Brun and F. Rademakers, \ROOT - An Object Oriented Data Analysis Framework", Nucl. Inst. Meth. in Phys. Res. A 389, 81 (1997).
3. https://root.cern.ch/download/doc/tmva/TMVAUsersGuide.pdf