Frontiers in particle physics through precision experiments and machine learning

Li.Jin, Andres Ramirez-Morales, Ana Gabriela Sanchez-Reyna, Jorge Salmon-Gamboa

Kyungpook National University

9th July 2020

Introduction: Machine Learning in HEP

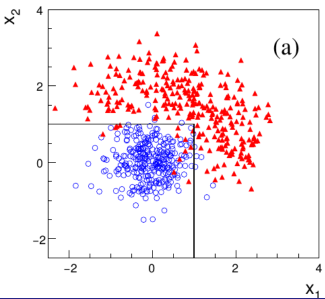
- Machine Learning (ML) $\rightarrow$ Artificial Intelligence
  - Multivariate Analysis MVA used to describe/deduce properties of data
  - ML: Learn(fit) the MVA statistical model using understood(training) data, *i.e.* **knowledge is extracted from experience** $\rightarrow$ Artificial Intelligence
- MVA methods in HEP used to:
  - Identify interesting collisions in a trigger system
  - Reject beam-induced hits in drift chambers during track finding
  - Provide particle identification info
  - Reject background particles/events at analysis stage
  - The training information is usually known from Monte Carlo simulation
- **Classification** is the major task undertaken in HEP
  - It is a statistical model $\hat{f} > C$ (for a given threshold C), that distinguishes between signal and background
- Popular classifiers: Boosted Decision Trees, **Support Vector Machines**, Artificial Neural Networks

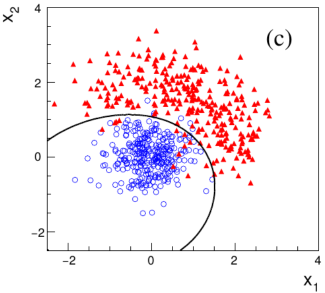
Cut-based Analysis

- 1D cuts set to increase signal/background
- Done in Monte Carlo samples
- Physics knowledge included
- Simple check of data/MC agreement
- Systematics evaluated fairly easy
- Correlation of different features is not fully exploited
- Heavily relies on human expertise

MVA-based Analysis

- MVA creates a data statistical model
- Done in Monte Carlo samples
- Multivariate correlations considered, enhancing the statistical power
- The model is created automatically
- The uncertainties are much more difficult and less clear to evaluate
- Given the complexity → black-box

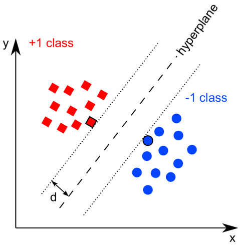


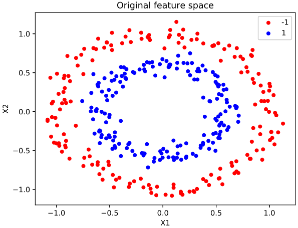


Support Vector Machines in a nutshell

- Decision boundary given by:

$$f(\mathbf{x}) = \langle \mathbf{x}, \phi(\mathbf{x}) \rangle + b$$





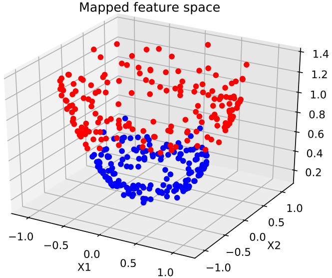
- Minimization problem:

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i$$

$$+ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$$

Subject to:

$$\sum_{i=1}^{N} y_i \alpha_i = 0, \forall i : 0 \leq \alpha_i \leq C$$



- Kernel function k, with ϕ mappings:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

$$= \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2)/2\sigma^2$$

- C and σ are **“the”** SVM hyperparameters

Ensemble methods

- An ensemble, in the ML context, is the combination of several ML classifiers to improve the performance of a single unit classifier, yielding a *Final Classifier* (e.g.BDTs, random forests)
- Boosting (weighting) and bagging (sampling) are popular methods to construct an ensemble
- **Boosting:** Is to combine several classifiers of the same type and weight (boost) each classifier according to a specific rule
- **AdaBoost**(adaptive boosting):
  - The AdaBoost weights are calculated/adapted with the classification errors of a trained classifier, samples are weighted with these, the process is iterated K times
  - AdaBoost has proven to be highly efficient, *i.e.* CERN's ATLAS and CMS, and Belle-II use it as bench mark

AdaBoost algorithm

- Errors ϵ_k in the classifier:

$$\epsilon_k = \sum_{i=i}^{N} w_i^k \quad \text{for} \quad y_i^{(k)} \neq f_k(\mathbf{x}_i)$$

- New sample weights $w_i^{(k+1)}$:

$$w_i^{(k+1)} = w_i^{(k)} \frac{e^{-\alpha_k f_k(\mathbf{x}_i) y_i}}{Z_k}$$

$$\alpha_k = \ln \frac{1-\epsilon_k}{\epsilon_k}, \quad \sum_{i=1}^{N} w_i = 1$$

- *Final classifier*(made of K classifiers):

$$y(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k f_k(\mathbf{x})$$

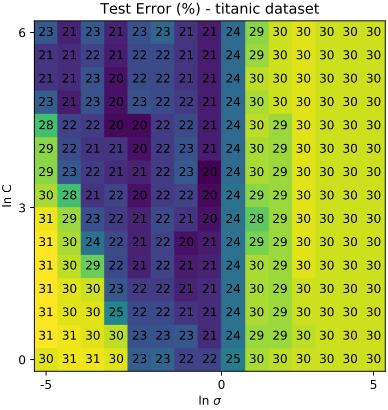
Why combine SVM and Adaboost?

- Both have relatively simple interpretation
- Have been explored relatively less w.r.t. other ML, e.g. NN, BDT
- SVM does not need to have specific parametric inputs, like BDT (how many trees? depth?), or NN (architecture))
- SVM is quite accurate
- SVM is robust against low statistics: important for HEP as some uncertainties are driven by statistical fluctuations

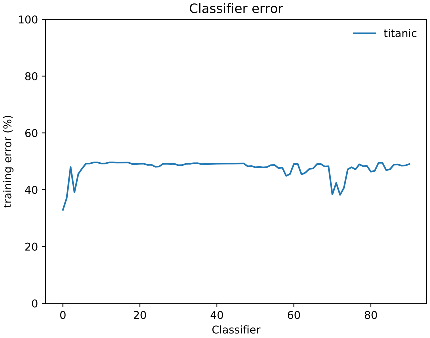
Strategy

- Study standard data. Shown today: Titanic data set ✓
- Boost different learner's hyper-parameters ✓
- Diversity ✓
- Generalization (test errors) ✓
- Play with different kernels
- Different AdaBoost sequences
- At random, select different features to do the training
- Semi supervised learning
- Take care of imbalanced datasets
- Apply it to HEP data

Results: boosted support vector machines



Hyperparameter grid search



Single test error

Results: boosted support vector machines

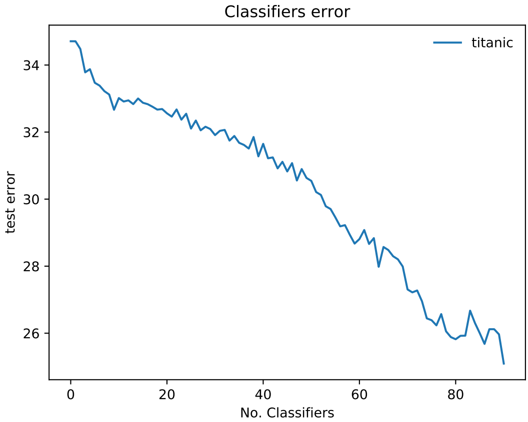


Figure: Bootstrap error calculation as function of no. of ensemble classifiers

Results: boosted support vector machines

Diversity Adaboost VS AdaBoost

Data set	AdaBoostSVM	Classifiers	AdaBoostSVM Diverse	Classifiers
Titanic	21.7	99	21.7	99
Cancer	34.9	93	31.9	25
Car	25.8	88	25.8	88
Contra	39.6	88	40.3	56

Table: Average test errors of different ensemble classifiers in percentage

Summary and outlook

- Boosted support vector machines have been successfully implemented
- Early results are quite promising
- There is still a wide range of strategies to be implemented to further improve this ensemble technique
- Soon will be applied to HEP data

References

- Corinna Cortes and Vladimir Vapnik, Support-Vector Networks, Machine Learning, 20, 273-297 (1995).
- Jerome H. Friedman, Stochastic gradient boosting, Computational Statistics & Data Analysis 38 367-378 (2002).
- Xuchun Li, Lei Wang, Eric Sung, AdaBoost with SVM-based component classifiers, Engineering Applications of Artificial Intelligence 21 785-795 (2008).
- Shifei Ding, Zhibin Zhu and Xiekai Zhang, An overview on semi-supervised support vector machine, Neural Comput & Applic 10.1007.
- A. Bevan, et. al., Support Vector Machines and generalisation in HEP, Journal of Physics: Conference Series, 898, 072021 (2017).
- T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer New York Inc., 2001.
- A. Radovic, et. al., Machine learning at the energy and intensity frontiers of particle physics, Nature 560, 41-48(2018).
- D. Guest, et. al., Deep Learning and Its Application to LHC Physics, Annual Review of Nuclear and Particle Science 68, 161-181(2018).

Abstract

Machine learning (ML) methodologies have gained remarkable popularity in the high energy physics (HEP) community in recent years. This has improved the achieved precision of HEP experimental results. This work presents the study of novel ML techniques which address the problem of classification within experimental data to discriminate physics interesting versus background events. Specifically, it is focused on the combination of a number of individual classifiers through the so-called boosting ensemble algorithms. The novelty of this work resides on the use of a definite strong classifier as a component classifier in an ensemble, that is, support vector machines (SVMs); furthermore, different approaches to build the ensembles are explored. The previous considerations aim to improve the performance of a single SVM as being part of an ensemble and to be competitive with existing ML algorithms used as benchmarks by experimentalists (boosted decision trees and neural networks). The final goal of this effort is to apply the proposed algorithm to HEP measurements and reach precision frontiers. Early results on public available data will be presented.