Flavor Tagging Using Machine Learning Algorithms

Fan Yang[1], Bingyang Zhang[2], Gang Li[2] and Manqi Ruan[2]

[1]Depart of Automation, Xiamen University
[2]The Institute of High Energy Physics of the Chinese Academy of Sciences

November 7, 2017

Outline

1. Background
2. Related work
3. Machine Learning algorithms
4. Experiments
5. Conclusions

Outline

1. Background
2. Related work
3. Machine Learning algorithms
4. Experiments
5. Conclusions

Background: Flavor Tagging at CEPC

- Quarks & Gluons are Fragmented into jets: essential to determine the original flavor of the parton
- Physics motivation:
  - At Higgs program, Essential to distinguish H–>bb, H–>cc and H–>gg events: the measurement of g(Hbb), g(Hcc) and g(Hgg)
  - Enhance the Signal/background separation for multiple analysis
  - Searching for FCNC & exotic decay of Bosons
- Technically difficult, especially to identify the c-quark jets:
  - TMVA method has been heavily used in Flavor tagging studies;
  - Try state of the art machine learning algorithms

Outline

1. Background

2. Related work

3. Machine Learning algorithms

4. Experiments

5. Conclusions

Related work

- Flavor tagging
  Distinguishing different classes of jets, e.g. *b* quark, *c* quark and *uds* can be regarded as a binary or multi-class classification task.
- Classification or Prediction
  input: high-dimensional variables
  output: labels
  train a classifier on the training data, then predict the label of an unseen point
- Variable selection or feature selection
  Enhance the interpretability of the models
  Investigating the effect of variable selection with Recursive Feature Elimination (RFE) in Flavor tagging

Outline

1 Background

2 Related work

3 Machine Learning algorithms

4 Experiments

5 Conclusions

Classification algorithms

An investigation on the prediction performances of State-of-The-Art approaches.

- Deep learning
  DNN: deep neural networks
- Tree ensemble methods: a collective of decision trees
  GCForest: multi-Grained Cascade forests
  GBDT: gradient boost decision trees
  Xgboost: eXtreme Gradient Boosting

Decision Tree

A decision tree recursively partitions the events in the feature space, which consists of many nodes.

At each node, the model select a 'best' variable to split.

Impurity measures the homogeneity of a node.

Gini impurity measures the degree of impurity.

$$Gini(t) = 1 - \sum_{i=0}^{c-1} [p(i|t)]^2$$

Node splitting–> impurity decreases –> events are classified to different leaf-nodes

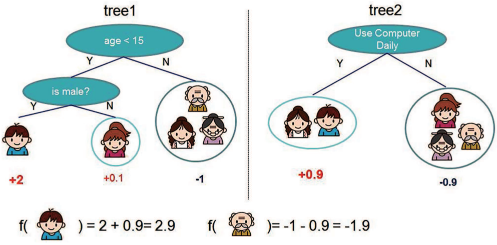


Figure: Decision tree model[1]

[1] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C],Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016: 785-794.

Deep learning

- Deep neural network(DNN)
  The neural network contains an input layer, many hidden layers and an output layer.
  A Black Box.

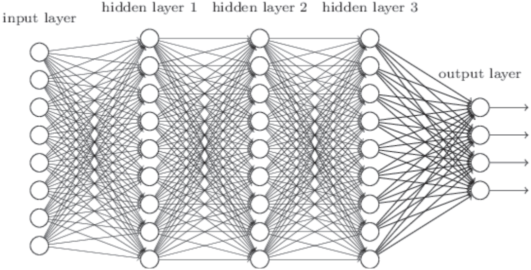


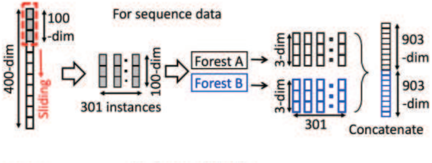
Figure: structure of DNN [1]

[1] http://www.cnblogs.com/pinard/p/6418668.html

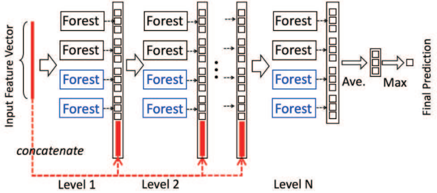
GCForest

GCForest (multi-Grained Cascade Forest)[1] consists of many random forest models.

- multi-grained scanning
- cascade forest



multi-grained scanning[1]



cascade forest[1]

[1] Zhou Z H, Feng J. Deep forest: Towards an alternative to deep neural networks[J]. arXiv preprint arXiv:1702.08835, 2017.

Boosted Decision Trees - The baseline

- Boosting

  Boosting ensemble improves when new basis functions $f_t(\mathbf{x})$ are added.

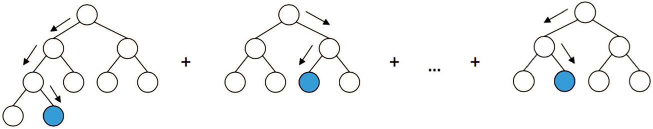
  $$F_t(\mathbf{x}) = F_{t-1}(\mathbf{x}) + f_t(\mathbf{x}),$$

  $f_t(\mathbf{x})$ is obtained by minimize loss function $L(\cdot)$.

- GBDT[1]

  Negative gradient "$-g(\mathbf{x})$" gives the best step direction.

  $$f_t = \arg\min_f L\big(-g_t(\mathbf{x}), f(\mathbf{x})\big),$$



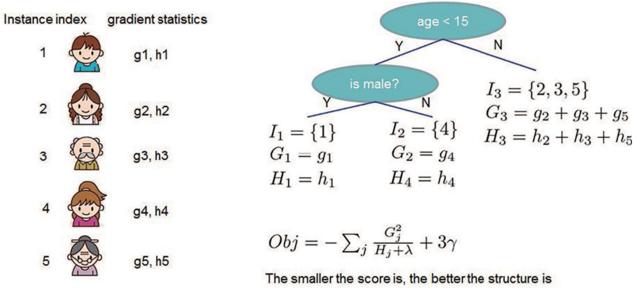
[1] Friedman J H. Greedy function approximation: a gradient boosting machine[J]. ,Annals of statistics, 2001: 1189-1232.

XGboost

In XGboost[1] algorithm, second-order approximation of loss function is used to optimize the prediction learner.

$$L^{(t)} = \sum_{i=1}^{n}[l(y_i, F_{t-i}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t),$$

where g_i and h_i are first and second order gradient.



Instance index	gradient statistics
1	g1, h1
2	g2, h2
3	g3, h3
4	g4, h4
5	g5, h5

$I_1 = \{1\}$, $G_1 = g_1$, $H_1 = h_1$

$I_2 = \{4\}$, $G_2 = g_4$, $H_4 = h_4$

$I_3 = \{2, 3, 5\}$, $G_3 = g_2 + g_3 + g_5$, $H_3 = h_2 + h_3 + h_5$

$$Obj = -\sum_j \frac{G_j^2}{H_j+\lambda} + 3\gamma$$

The smaller the score is, the better the structure is

[1] Chen T, Guestrin C. Xgboost: A scalable tree boosting system[C],Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016: 785-794.

Summary of the algorithms

algorithm	basic unit	integration form	output form
DNN	neuron	former layer's outputs are passed to the next layer as inputs	neurons in the output layer determine the value of output vector
gcforest	forest	former layer's outputs combined with initial data are passed to the next layer	averaging the outputs of forests in the last layer
GBDT	CART tree	training residuals are passed to the next	results of basis learners are summed up as final output
xgboost	tree	second-order approximation of loss function is used to optimize child nodes	weighted sum of all leaf nodes' output

Variable selection

To enhance the interpretability.

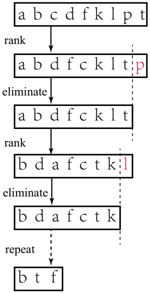
Make the model simpler.

Can be implemented easily in trees ensemble methods which can output importance scores for each variable.

- RFE

  The main idea of Recursive feature elimination (RFE) is removing the least important feature from the current feature set recursively.

- training an estimator with the current feature set(10-fold cross validation);
- obtaining feature importance scores by averaging;
- eliminating the least important feature from the feature set;
- repeating the procedure.



a b c d f k l p t

rank

a b d f c k l t p

eliminate

a b d f c k l t

rank

b d a f c t k l

eliminate

b d a f c t k

repeat

b t f

Outline

1 Background

2 Related work

3 Machine Learning algorithms

4 Experiments

5 Conclusions

Experiment setting

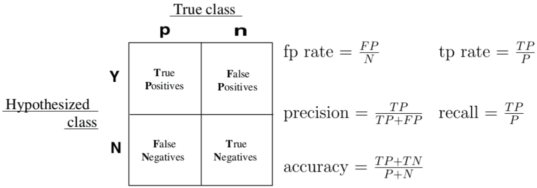
- Data
  - Experimental data were generated from simulation tools.The number of variables is 63.
  - To avoid overfitting, 630000 events are split randomly into a training set (400000), three validation sets (50000 events per set) and a test set (80000).
- Evaluation metrics
  - Tagging accuracy.
  - Misidentification vs. Tagging efficiency.
  - Area under the ROC (AUC)
- Classification methods
  - Hyperparameters are fine-tuned by maximizing the average accuracies on validation sets.
- Variable selection
  - Scores of models with different features are compared in order to achieve a "performance-complexity" balance.

ROC and AUC

Confusion matrix, table of confusion

ROC, Receiver Operating Characteristics Curve, illustrates the diagnostic ability of a binary classifier.

AUC, Area under the curve.



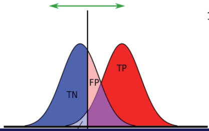
		True class	
		p	n
Hypothesized class	Y	True Positives	False Positives
	N	False Negatives	True Negatives

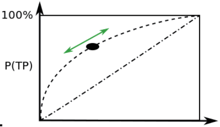
$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

$$\text{accuracy} = \frac{TP+TN}{P+N}$$

Figure: Confusion matrix





Cross validation(CV)

Optimum parameters for every algorithm can be fine-tuned on the training data in 10-fold CV, which is widely used as model evaluation technique.

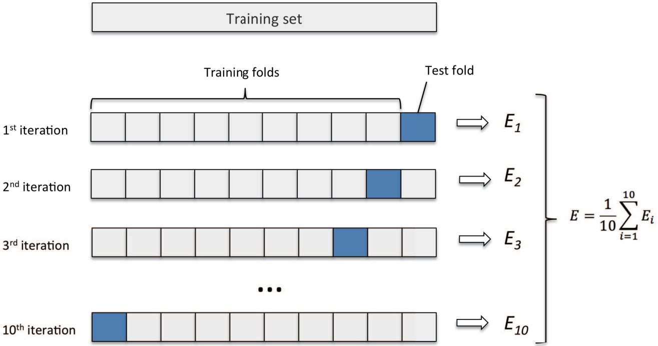


Figure: K-fold cross-validation

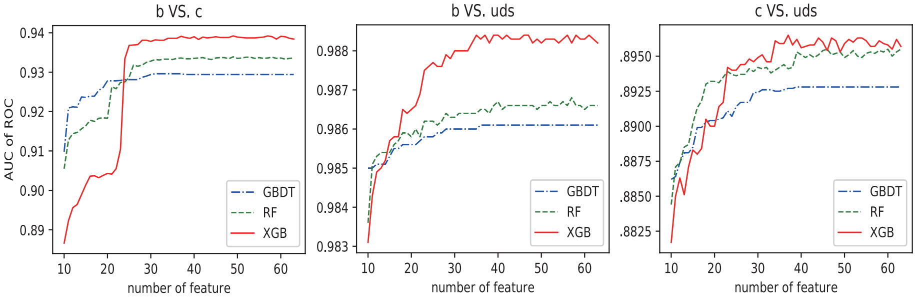
Classification results

- Accuracy
  For events in the test set, each algorithm has three outputs which represent the probabilities of three categories. The label with the maximum probability is the predicted class of the events.
  The average accuracies are shown in the table.

Algorithm	DNN	BDT	GBDT	gcforest	xgboost
Accuracy	0.788	0.776	0.794	0.785	0.801

Experiment

Variable selection with Recursive Feature Elimination



Experiment

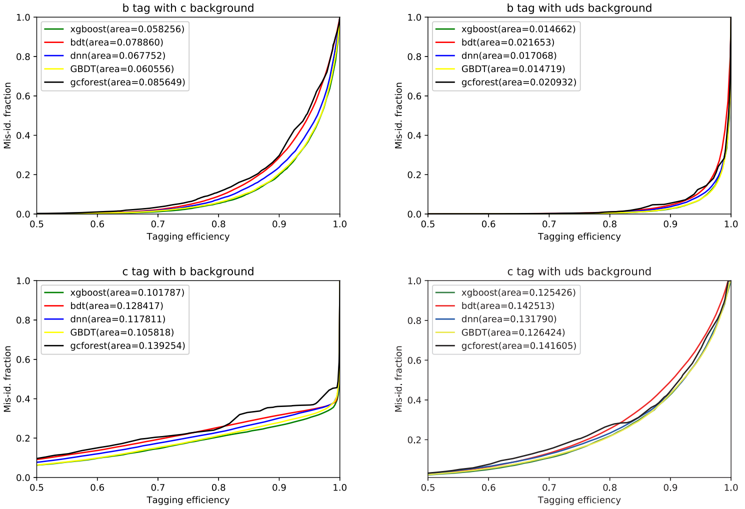
The highest AUC and corresponding number of features

Algorithm	[b-c]	[b-uds]	[c-uds]
GBDT	0.929 / 31	0.986 / 36	0.893 / 39
RF	0.933 / 49	0.987 / 57	0.895 / 46
XGB	0.939 / 49	0.988 / 35	0.896 / 37

Data in the table is displayed in the form of "a/n" where "a" denotes the highest AUC and "n" denotes the corresponding number of features.

Experiment

Tagging efficiency vs. Mis-Id fraction:



Experiment

Tagging efficiency vs. Mis-Id fraction:

tag-background	efficiency (%)	Mis-id fraction (%)				
		xgboost	DNN	GBDT	BDT	gcforest
b-c	80	5.4	7.5	5.8	9.3	10.8
	90	20.1	23.7	20.6	29.2	26.3
	95	39.0	43.5	39.6	50.2	56.3
b-uds	80	0.5	0.7	0.5	1.0	1.1
	90	2.7	3.7	2.8	4.7	4.9
	95	7.8	9.7	7.8	11.3	13.6
c-b	80	20.8	23.1	21.5	25.6	25.1
	90	26.5	30.2	28.1	32.1	36.1
	95	30.6	33.9	31.8	34.4	36.8
c-uds	80	22.3	23.3	22.3	26.0	27.4
	90	43.4	43.5	43.8	51.9	43.5
	95	63.6	61.7	62.1	68.8	66.1

Outline

1. Background
2. Related work
3. Machine Learning algorithms
4. Experiments
5. Conclusions

Conclusions

- State-of-the-art machine learning algorithms achieve high accuracies.
- XGboost is a promising tool for its interpretability and accuracy.
- Future work.
  Imbalanced data
  Real-world data