

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

IIIIIOuucii

sensitivi

Baggi

Applicatio

Compariso

Conclusions and outlook

# Application of the rule-growing algorithm RIPPER to particle physics analysis

Markward Britsch<sup>1</sup>, Nikolai Gagunashvili<sup>1,2</sup>, Michael Schmelling<sup>1</sup>

<sup>1</sup>Max-Planck-Institute for Nuclear Physics, <sup>2</sup>University of Akureyri, Iceland







2008-11-5, ACAT 2008, Erice



#### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

IIIIIOuuc

ooot

sensitivi

Baggir

Application

Comparison

- 1 Introduction
- 2 RIPPER
- 3 Cost-sensitive classification
- 4 Bagging
- 5 Applying the MVA algorithm
- 6 Comparison to other methods
- 7 Conclusions and outlook



#### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

. . . . . . .

cost-

sensitiv

Dagg..

. .

Companso

- 1 Introduction
- 2 RIPPER
- 3 Cost-sensitive classification
- 4 Bagging
- 5 Applying the MVA algorithm
- 6 Comparison to other methods
- 7 Conclusions and outlook



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

minoadono

cost-

Baggino

- 55 .

Compariso

Conclusions and outlook

called data mining in computer science community



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introducti

DIDDED

cost-

Baggin

Compariso

- called data mining in computer science community
- classifier (neural network, decision tree ...) learns on a training data set



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

cost-

Baggi

Application

Compariso

- called data mining in computer science community
- classifier (neural network, decision tree . . . ) learns on a training data set
- classifier output: probability (e.g. for a candidate to be signal)



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

costsensitivi

Baggi

Application

Compariso

- called data mining in computer science community
  - classifier (neural network, decision tree . . . ) learns on a training data set
- classifier output: probability (e.g. for a candidate to be signal)
- cut on probability



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introducti

RIPPER

costsensitivi

Baggi

Applicatio

Compariso

- called data mining in computer science community
  - classifier (neural network, decision tree . . . ) learns on a training data set
  - classifier output: probability (e.g. for a candidate to be signal)
  - cut on probability
    - to change signal to background or significance



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

costsensitivi

Baggi

Application

Compariso

- called data mining in computer science community
  - classifier (neural network, decision tree . . . ) learns on a training data set
  - classifier output: probability (e.g. for a candidate to be signal)
- cut on probability
  - to change signal to background or significance
  - to account for larger abundance (in real data) of BG



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

minoduotio

cost-

Poggin

Dayyiii

Comparine

Conclusions

i.e. in HEP often imbalanced problems
 e.g. much more background than signal events



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

costsensitivi

Poggina

Application

Comparison

- i.e. in HEP often imbalanced problems
   e.g. much more background than signal events
- from data mining, possible solution:
  - appropriate classifier
  - Cost-sensitive approach
  - sampling based approach
  - bagging



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

costsensitivi

Baggin

Application

Comparison

- i.e. in HEP often imbalanced problems
   e.g. much more background than signal events
- from data mining, possible solution:
  - appropriate classifier
  - Cost-sensitive approach
  - sampling based approach
  - bagging
- some partly equivalent to cut on probability



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

sensitivi

Bagging

Application

Comparisor

- i.e. in HEP often imbalanced problems
   e.g. much more background than signal events
- from data mining, possible solution:
  - appropriate classifier
  - Cost-sensitive approach
  - sampling based approach
  - bagging
- some partly equivalent to cut on probability
- → investigate the difference between cut on probability and data mining solutions above



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

costsensitivi

Ragging

Applicatio

Comparison

- i.e. in HEP often imbalanced problems
   e.g. much more background than signal events
- from data mining, possible solution:
  - appropriate classifier
  - Cost-sensitive approach
  - sampling based approach
  - bagging
- some partly equivalent to cut on probability
- → investigate the difference between cut on probability and data mining solutions above
- → which is better?



#### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

#### RIPPER

sensitivit

Baggir

Applicatio

Comparison

- 1 Introduction
- 2 RIPPER
- 3 Cost-sensitive classification
  - 4 Bagging
- 5 Applying the MVA algorithm
- 6 Comparison to other methods
- 7 Conclusions and outlook



#### What are rule sets?

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

#### RIPPER

costsensitivit

Ragging

Application

Comparison

Conclusions and outlook

Technique for classifying events using a collection of "if...then..." rules. For example:

```
(IPpi >= 1.039316) and (DoCA <= 0.307358) and (IP <= 0.270767) and (IPp >= 0.800645) => class=Lambda
```

```
(IPpi >= 0.637403) and (DoCA <= 0.159043) and (IP <= 0.12081) and (ptpi >= 149.2332) and (IP >= 0.003371) => class=Lambda
```

=> class=BG



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

....

DIDDED

cost-

Baggin

Dagg...

Commonico

Conclusion

- direct rule based classifier (see Cohen (1995) [1])
  - 1 divide training set into growing and pruning sets



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

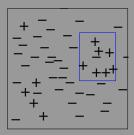
iiiii oddollol

cost-

Baggino

Compariso

- direct rule based classifier (see Cohen (1995) [1])
  - 1 divide training set into growing and pruning sets
  - grow a rule adding conditions greedily



rule 1



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

ii iii oddolioi

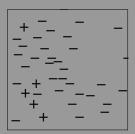
cost-

Bagging

Compariso

Conclusions and outlook

- direct rule based classifier (see Cohen (1995) [1])
  - 1 divide training set into growing and pruning sets
  - grow a rule adding conditions greedily



delete rule 1 instances



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

costsensitivit

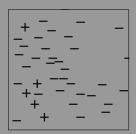
Bagging

Applicatio

Compariso

Conclusions and outlook

- direct rule based classifier (see Cohen (1995) [1])
  - 1 divide training set into growing and pruning sets
  - grow a rule adding conditions greedily
  - g prune rule



delete rule 1 instances



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

cost-

sensitivit

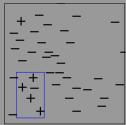
. . . .

Comparison

Conclusions

direct rule based classifier (see Cohen (1995) [1])

- 1 divide training set into growing and pruning sets
- 2 grow a rule adding conditions greedily
- 3 prune rule
- 4 go to 2), stopping criteria: description length, error rate



rule 2



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

cost-

Baggir

Applicatio

Comparison

Conclusion

- direct rule based classifier (see Cohen (1995) [1])
  - 1 divide training set into growing and pruning sets
  - grow a rule adding conditions greedily
  - 3 prune rule
  - 4 go to 2), stopping criteria: description length, error rate
  - optimization of rules



RIPPER

Britsch, Gagunashvili, Schmelling

miroducii

RIPPER

costsensitivi

Baggi

Application

Compariso

Conclusions and outlook

- direct rule based classifier (see Cohen (1995) [1])
  - divide training set into growing and pruning sets
  - grow a rule adding conditions greedily
  - 3 prune rule
  - go to 2), stopping criteria: description length, error rate
  - optimization of rules

#### Advantages:

- rule set: relatively easy to interpret
- good for imbalanced problems



#### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

costsensitivity

Sensilivi

Application

Compariso

- 1 Introduction
- 2 RIPPER
- 3 Cost-sensitive classification
  - 4 Bagging
- 5 Applying the MVA algorithm
- 6 Comparison to other methods
- 7 Conclusions and outlook



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

. . . . . . .

costsensitivity

Baggin

0----

 assign a cost to wrongly (or correctly) classified instances ("events", "candidates")



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

minoduct

cost-

sensitivit

Bagging

. .

Comparison

Conclusions

 assign a cost to wrongly (or correctly) classified instances ("events", "candidates")

→ cost matrix, e.g.:

	predicted BG	predicted signal
true BG	0	100
true signal	1	0



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

costsensitivit

Baggir

Compariso

Conclusions and outlook  assign a cost to wrongly (or correctly) classified instances ("events", "candidates")

→ cost matrix, e.g.:

	predicted BG	predicted signal
true BG	0	100
true signal	1	0

classification algorithm minimizes cost



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

RIPPER

costsensitivit

Raggin

. .

Comparison

Conclusions and outlook

 assign a cost to wrongly (or correctly) classified instances ("events", "candidates")

→ cost matrix, e.g.:

	predicted BG	predicted signal
true BG	0	100
true signal	1	0

- classification algorithm minimizes cost
- mainly two ways:
  - threshold adjusting
  - instance weighting



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

RIPPER

costsensitivity

Baggin

Conclusion

Let's start with a cost matrix as before:

	pred. BG	pred. signal
tr. BG	0	C(BG, s)
tr. signal	C(s, BG)	0



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

.....

cost-

sensitivi

Applicati

Compariso

Conclusion

Let's start with a cost matrix as before:

	pred. BG	pred. signal
tr. BG	0	C(BG, s)
tr. signal	C(s, BG)	0

Minimize cost for a rule t, class i = s, BG:

$$C(i|t) = \sum_{j=s,BG} p(j|t)C(j,i).$$



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

costsensitivit

Doggin

Applicati

Compariso

Conclusions and outlook

Let's start with a cost matrix as before:

	pred. BG	pred. signal
tr. BG	0	C(BG, s)
tr. signal	C(s, BG)	0

Minimize cost for a rule t, class i = s, BG:

$$C(i|t) = \sum_{j=s,BG} p(j|t)C(j,i).$$

t is assigned to the signal class if:

$$p(s|t)$$
 >  $p(BG|t)$ 



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

cost-

sensitivi

Applicati

Compariso

Conclusions

Let's start with a cost matrix as before:

	pred. BG	pred. signal
tr. BG	0	C(BG, s)
tr. signal	C(s, BG)	0

Minimize cost for a rule t, class i = s, BG:

$$C(i|t) = \sum_{j=s,BG} p(j|t)C(j,i).$$

t is assigned to the signal class if:



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

DIDDED

costsensitivit

Danasia

Applicati

Compariso

Conclusions and outlook

Let's start with a cost matrix as before:

	pred. BG	pred. signal
tr. BG	0	C(BG, s)
tr. signal	C(s, BG)	0

Minimize cost for a rule t, class i = s, BG:

$$C(i|t) = \sum_{j=s,BG} p(j|t)C(j,i).$$

t is assigned to the signal class if:

$$p(\mathbf{s}|t)C(\mathbf{s}, \mathrm{BG}) > p(\mathrm{BG}|t)C(\mathrm{BG}, \mathbf{s})$$
  
 $\Rightarrow p(\mathbf{s}|t)C(\mathbf{s}, \mathrm{BG}) > 1 - p(\mathbf{s}|t)C(\mathrm{BG}, \mathbf{s})$   
 $\Rightarrow p(\mathbf{s}|t) > \frac{C(\mathrm{BG}, \mathbf{s})}{C(\mathrm{BG}, \mathbf{s}) + C(\mathbf{s}, \mathrm{BG})}$ 



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

cost-

sensitivit

Daggii

Compario

Canalysian

Let's start with a cost matrix as before:

	pred. BG	pred. signal
tr. BG	0	C(BG, s)
tr. signal	C(s, BG)	0

Minimize cost for a rule t, class i = s, BG:

$$C(i|t) = \sum_{j=s,BG} \rho(j|t)C(j,i).$$

t is assigned to the signal class if:

$$\begin{aligned} & p(\mathbf{s}|t)C(\mathbf{s}, \mathbf{BG}) > p(\mathbf{BG}|t)C(\mathbf{BG}, \mathbf{s}) \\ & \Rightarrow p(\mathbf{s}|t)C(\mathbf{s}, \mathbf{BG}) > 1 - p(\mathbf{s}|t)C(\mathbf{BG}, \mathbf{s}) \\ & \Rightarrow p(\mathbf{s}|t) > \frac{C(\mathbf{BG}, \mathbf{s})}{C(\mathbf{BG}, \mathbf{s}) + C(\mathbf{s}, \mathbf{BG})} \end{aligned}$$

→ This is equivalent to a cut on the probability!



# Sampling and instance weighting

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

cost-

sensitivi

Baggin

.

Conclusions

- simplest forms:
  - undersampling by leaving out instances
  - oversampling by replicating instances



# Sampling and instance weighting

Analysis with RIPPER

Gagunashvili, Schmelling

miroduci

cost-

Doggino

Application

Comparisor

Britsch,

#### simplest forms:

- undersampling by leaving out instances
- oversampling by replicating instances
- mainly equivalent to applying a cost:

C(s,BG) (C(BG,s)) – replication factor of signal (BG)



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

miroduc

cont

sensitivit

Applicati

Compariso

Conclusions and outlook

simplest forms:

- undersampling by leaving out instances
- oversampling by replicating instances
- mainly equivalent to applying a cost:

C(s, BG) (C(BG, s)) – replication factor of signal (BG)

 instance weighting: automated sampling/weighting of instances according to cost



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

ooot

sensitivit

Baggii

\_ .

Conducion

simplest forms:

- undersampling by leaving out instances
- oversampling by replicating instances
- mainly equivalent to applying a cost:

C(s, BG) (C(BG, s)) – replication factor of signal (BG)

- instance weighting: automated sampling/weighting of instances according to cost
- for some classifiers (e.g. neural networks) not better than threshold adjusting



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

RIPPER

costsensitivit

Baggir

. .

Companso

Conclusions and outlook simplest forms:

- undersampling by leaving out instances
- oversampling by replicating instances
- mainly equivalent to applying a cost:

C(s, BG) (C(BG, s)) – replication factor of signal (BG)

- instance weighting: automated sampling/weighting of instances according to cost
- for some classifiers (e.g. neural networks) not better than threshold adjusting
- better than threshold adjusting for classifiers that change with the balance of training data



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

KIPPEK

costsensitivit

Baggir

\_ .

Conclusion

simplest forms:

- undersampling by leaving out instances
- oversampling by replicating instances
- mainly equivalent to applying a cost:

C(s, BG) (C(BG, s)) – replication factor of signal (BG)

- instance weighting: automated sampling/weighting of instances according to cost
- for some classifiers (e.g. neural networks) not better than threshold adjusting
- better than threshold adjusting for classifiers that change with the balance of training data
- e.g. decision trees, rules typically using error rate



### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

sensitivit

**Bagging** 

Application

Comparison

- 1 Introduction
- 2 RIPPER
- 3 Cost-sensitive classification
- 4 Bagging
- 5 Applying the MVA algorithm
- 6 Comparison to other methods
- 7 Conclusions and outlook



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

IIIIIOduc

RIPPER

costsensitivi

Baggin

Application

Compariso

Conclusions and outlook

 draw with replacement at random N instances from your sample

orig. sample	1	2	3	4	5
1st iteration	2	5	1	1	4
2 <sup>nd</sup> iteration	5	3	2	2	4



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

cost-

Bagging

Dagg...

Compariso

- draw with replacement at random N instances from your sample
- do this r times



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

.....

cost-

Bagging

Application

Compariso

- draw with replacement at random N instances from your sample
- do this r times
- learn r classifiers (here r rule sets) on these



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

madadada

cost-

sensitivi

Application

.

Conclusion

 draw with replacement at random N instances from your sample

- do this r times
- learn r classifiers (here r rule sets) on these
- let them vote or average their probabilities



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

....

costsensitivi

sensitivi

Application

Comparisor

Conclusions

 draw with replacement at random N instances from your sample

- do this r times
- learn r classifiers (here r rule sets) on these
- let them vote or average their probabilities
- N is typically the number of instances in your sample



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

KIFFEK

sensitivi

Applicati

Compariso

- draw with replacement at random N instances from your sample
- o do this *r* times
- learn r classifiers (here r rule sets) on these
- let them vote or average their probabilities
- N is typically the number of instances in your sample
- this works very well if your classifier is unstable, i.e.
   prone to change with noise (RIPPER, decision trees)



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

**RIPPER** 

costsensitivi

Bagging

Application

Compariso

- draw with replacement at random N instances from your sample
- o do this *r* times
- learn r classifiers (here r rule sets) on these
- let them vote or average their probabilities
- N is typically the number of instances in your sample
- this works very well if your classifier is unstable, *i.e.* prone to change with noise (RIPPER, decision trees)
- reduces overfitting (for oversampling)



### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

. . . . . .

sensitivi

Baggir

#### Application

Compariso

- 1 Introduction
- 2 RIPPER
- (3) Cost-sensitive classification
- 4 Bagging
- 5 Applying the MVA algorithm
  - (6) Comparison to other methods
  - 7 Conclusions and outlook



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

....

cost-

Baggin

Application

Compariso

$$\circ$$
  $\Lambda \rightarrow p^+ + \pi^-$ 



Analysis with

Britsch, Gagunashvili. Schmelling

Application

$$\circ \Lambda \rightarrow p^+ + \pi^-$$

LHCb Monte Carlo minimum bias



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

IIIIIoductio

cost-

\_ .

Bagging

Application

$$\circ \Lambda \rightarrow p^+ + \pi^-$$

- LHCb Monte Carlo minimum bias
- candidates: pairs of differently charged long tracks



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

IIIIIoductio

cost-

Doggir

Application

Compariso

$$\circ \Lambda \rightarrow p^+ + \pi^-$$

- LHCb Monte Carlo minimum bias
  - candidates: pairs of differently charged long tracks
  - training set: 5 × 1000 Λ, 13000 BG



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

cost-

sensitivi

Application

, ipplication

Conclusion

 $\circ \Lambda \rightarrow p^+ + \pi^-$ 

LHCb Monte Carlo minimum bias

candidates: pairs of differently charged long tracks

 $\circ$  training set: 5  $\times$  1000  $\Lambda$ , 13000 BG

testing set: 1000 Λ, 180000 BG



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

cost-

sensitivi

. . . .

Application

Compariso

$$\wedge \wedge \rightarrow p^+ + \pi^-$$

- LHCb Monte Carlo minimum bias
  - candidates: pairs of differently charged long tracks
  - training set:  $5 \times 1000 \text{ A}$ , 13000 BG
- testing set: 1000 Λ, 180000 BG
- use 10 geometric and kinematic variables



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

. . . . . . .

cost-

Dogodio

Bagging

Application

Compansor

- classification step using WEKA [2] package:
  - bagging
  - 2 set cost (instance weighting)
  - 3 apply RIPPER



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

cost-

Bagging

Dayyırıç

Application

- classification step using WEKA [2] package:
  - bagging
  - (2) set cost (instance weighting)
  - 3 apply RIPPER
- make two classification steps:



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

milioduc

KIFFLIX

sensitivi

Baggi

Application

Compariso

- classification step using WEKA [2] package:
  - bagging
  - 2 set cost (instance weighting)
  - 3 apply RIPPER
- make two classification steps:
  - preclassification using bagging (10 bags) (high cost for loosing Λ → keep almost all Λs, reduce BG)

	pr. BG	pr. Λ	
tr. BG	0	1	
tr. ∧	0		
preselection cost matri			



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

IXII I LIX

sensitiv

Baggii

Application

Comparisor

- classification step using WEKA [2] package:
  - bagging
  - 2 set cost (instance weighting)
  - 3 apply RIPPER
- make two classification steps:
  - preclassification using bagging (10 bags) (high cost for loosing Λ → keep almost all Λs, reduce BG)
  - classify using bagging (25 bags) with high cost for wrongly accepted BG

	pr. BG	pr. Λ	
tr. BG	0	1	
tr. $\Lambda$	100	0	
preselection cost matri			

	pr. BG	pr. Λ	
tr. BG	0	X	
tr. $\Lambda$	1	0	
main cost matrix			



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

muoduc

KIFFLIX

sensitivi

...

Application

Compariso

- classification step using WEKA [2] package:
  - bagging
  - 2 set cost (instance weighting)
  - 3 apply RIPPER
- make two classification steps:
  - preclassification using bagging (10 bags) (high cost for loosing Λ → keep almost all Λs, reduce BG)
  - classify using bagging (25 bags) with high cost for wrongly accepted BG
  - 3 to produce ROC curve: scan cost x

	pr. BG	pr. Λ	
tr. BG	0	1	
tr. A 100 0			
preselection cost matrix			

		pr. BG	pr. Λ
	tr. BG	0	X
	tr. ∧	1	0
main cost matrix			



#### **ROC** curve

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

KIFFLIX

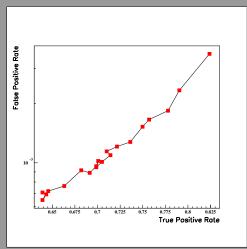
sensitiv

Baggii

Application

Compariso

Conclusions and outlook



Cost x = 10, 20, ..., 200



## Mass plots

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

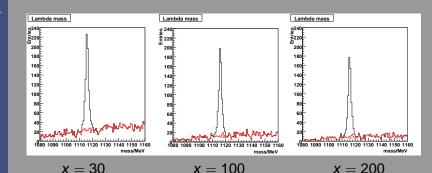
minoductic

cost-

Baggi

Application

Compariso





## Without bagging and instance weighting

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

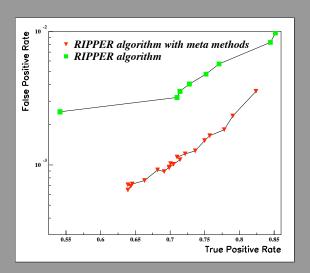
.....

cost-

Baggir

Application

Compariso





## Threshold adjusting vs. instance weighting

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

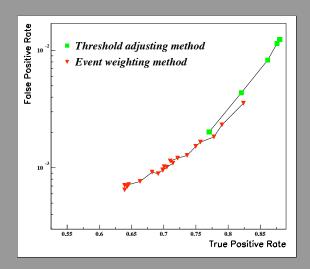
Introduction

cost-

sensiliv

Application

Compariso





## Different bagging parameters

Analysis with RIPPER

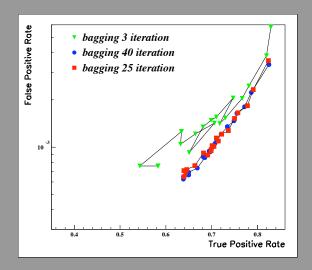
Britsch, Gagunashvili, Schmelling

Introduction

cost-

00110141

Application





### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

cost-

sensitivi

Dayyiii

. .

Comparisor

- 1 Introduction
- 2 RIPPER
- 3 Cost-sensitive classification
- 4 Bagging
- 5 Applying the MVA algorithm
- 6 Comparison to other methods
- 7 Conclusions and outlook



### Comparison with TMVA decision tree

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

RIPPER

cost-

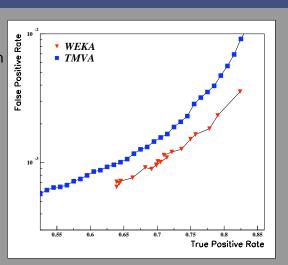
301131111

Daggii

Conclusions

TMVA [3] decision tree (by Helge Voss):

- boosting
- pruning



false positive rate vs. true positive rate: TMVA tree, RIPPER



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

RIPPER

costsensitivi

Baggin

Dagg...

Compariso

Conclusions and outlook

Was RIPPER the right algorithm to choose?



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

**RIPPER** 

costsensitivi

Baggin

Conclusion

Was RIPPER the right algorithm to choose? Compare with neural network (NN) and decision tree (DT); bagging and cost-sensitivity for *all* of the algorithms (no preclassification for *any*)



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

....

sensitivi

Baggin

...

Compariso

Conclusion and outlook

Was RIPPER the right algorithm to choose?

Compare with neural network (NN) and decision tree (DT); bagging and cost-sensitivity for all of the algorithms (no preclassification for any) neural network:

- multi layer perceptron
- 3 layers, 6 internal nodes
- binary output



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduct

sensitivi

Baggin

\_\_\_\_\_

Compansor

Conclusion and outlook

Was RIPPER the right algorithm to choose?

Compare with neural network (NN) and decision tree (DT); bagging and cost-sensitivity for all of the algorithms (no preclassification for any) neural network:

- multi layer perceptron
- 3 layers, 6 internal nodes
- binary output

#### decision tree

- OC 4.5
- includes pruning



## Comp. with neural network and decision tree

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

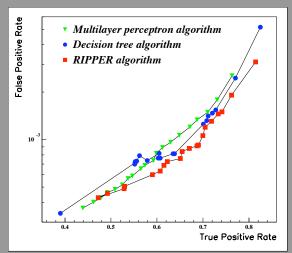
cost-

sensitivi

Applicati

Compariso

Conclusions



false positive rate vs. true positive rate: NN, tree, RIPPER



### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

.....

sensitivi

Baggir

Comparisor

- 1 Introduction
- 2 RIPPER
- 3 Cost-sensitive classification
  - 4 Bagging
- 5 Applying the MVA algorithm
- 6 Comparison to other methods
- 7 Conclusions and outlook



#### Conclusions

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduction

. . . . . . .

costsensitivit

Bagging

. . . . .

Comparison

- Cost-sensitive, sampling and cutting on probability are very similar
- instance weighting better for some classifiers
- bagging helps unstable classifiers, reduces overfitting



#### Conclusions

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introduc

sensitivi

. ..

Comparison

Conclusions and outlook

- Cost-sensitive, sampling and cutting on probability are very similar
- instance weighting better for some classifiers
- bagging helps unstable classifiers, reduces overfitting

#### Analysis w/ RIPPER, bagging, instance weighting:

- RIPPER fast and efficient to use
- bagging and instance weighting very important
- better than TMVA decision tree
- RIPPER better than NN or DT



#### Outlook

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Introductio

cost-

Poggin

Application

Comparisor

- use it for other analyzes (e.g.  $D^0$ )
- implement instance weighting in TMVA (and RIPPER?)
- dependence on MC errors
- increase weight on background in a more efficient way



#### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Bibliography

Backup slides

8 Bibliography

9 Backup slides



Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Bibliography

Backup slide

- W. W. Cohen, Fast Effective Rule Induction, In Machine Learning: Proc. of the 12th International Conference on Machine Learning, Lake Tahoe, California: Morgan Kaufman, 1995
- http://www.cs.waikato.ac.nz/ml/weka/
- TMVA Toolkit for multivariate data analysis with root, A. Höker et al., arXiv physiscs/0703039, http://tmva.sf.net.



#### Outline

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Bibliography

Backup slides

8 Bibliography

9 Backup slides



#### Set of variables

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Bibliograph

Backup slide

- DoCA distance of closest approach
- FL signed flight-length
- c · t flight-length in ∧ frame
- IPp IP proton
- IPpi IP pion
- $v_2 = \frac{IPpi^2 + IPp^2}{IP^2}$
- ptp pt proton
- ptpi pt pion
- $\tan \vartheta = \frac{pt}{pz}$
- $\circ$  cos  $\xi$ ,  $\xi$  angle between impact vectors



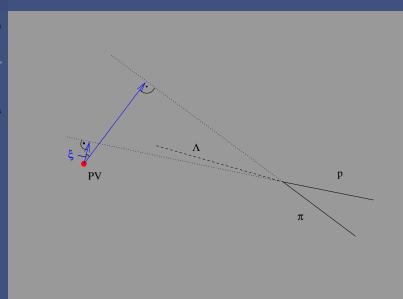
### A new variable: $\xi$

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Bibliography

Backup slides





## Imbalanced data sets in HEP and data mining

Analysis with RIPPER

Britsch, Gagunashvili, Schmelling

Bibliography

Backup elide

- typical data mining application
  - credit card fraud, AIDS test
  - rare class is "fraud transaction" or "AIDS infection"
  - → high cost not detecting rare instance
- HEP mostly (particle selection)
  - high background
  - high cost for (non-rare) background classified as signal
- which translates to:
  - data mining: imbalance has to be balanced
  - HEP particle selection: imbalance has to be enhanced