Bayesian non parametric modelling of Higgs pair production

Bruno Scarpa

Department of Statistical Sciences - University of Padua

September 1, 2016

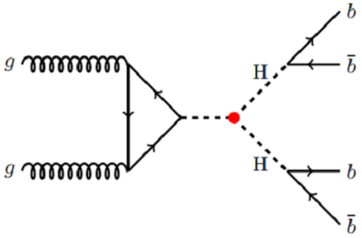
Università degli Studi di Padova

joint work with Annalisa Balata (University of Padua) and Tommaso Dorigo (INFN)

Università degli Studi di Padova

Goal

Isolate the signal of the Higgs boson pairs decays in the final state characterised by 4 jets of *b*-quark: $hh \to 4b$



CMS data

Goal

Isolate the signal of the Higgs boson pairs decays in the final state characterised by 4 jets of *b*-quark: $hh \to 4b$

Data

- *background*: 1 259 973 observations collected by CMS during the LHC "Run 1" in 2012

  (only if HLT-DiPFJet80-DiPFJet30-BTagCSVd07d05 trigger path is present)
- *signal*: 300 000 $hh \to b\bar{b}b\bar{b}$ events.
  Monte Carlo simulated events (Alwall et al., 2011; Gao et al. 2014)

Preselection of data: choice of the 4 jets

Events where 4 jets correspond to hadronisation of the *b*-quark

1. *b*-tagging algorithm CMVA (Das et al., 2013)
2. Selection of the first 3 jets in *b*-tag ranking, provided their CMVA is above the *medium cut*, 0.679
3. The fourth jet is chosed by requiring the least invariant mass difference between pairs of mached dijets

Events where 4 jets correspond to hadronisation of the *b*-quark

1. *b*-tagging algorithm CMVA (Das et al., 2013)
2. Selection of the first 3 jets in *b*-tag ranking, provided their CMVA is above the *medium cut*, 0.679
3. The fourth jet is chosed by requiring the least invariant mass difference between pairs of mached dijets

Final dataset

At the end of preselection, we keep
68 454 MC signal events and
433 621 background CMS data

Available variables

For each event, the following variables are available:

Available variables

For each event, the following variables are available:

Response variable

binary variable, y_i, encoding signal ($y_i = 1$) or background ($y_i = 0$)

Available variables

For each event, the following variables are available:

Response variable

binary variable, y_i, encoding signal $(y_i = 1)$ or background $(y_i = 0)$

Kinematic explanatory variables

- Variables related to the 4 selected jets and to the couples of dijets
  1. Transverse momentum
  2. Pseudorapidity
  3. Centrality

Available variables

Università degli Studi di Padova

For each event, the following variables are available:

Response variable

binary variable, y_i, encoding signal ($y_i = 1$) or background ($y_i = 0$)

Kinematic explanatory variables

- Variables related to the 4 selected jets and to the couples of dijets
  1. Transverse momentum
  2. Pseudorapidity
  3. Centrality
- Variables related to non selected jets
  1. Minimum, mean and maximum transverse momentum
  2. Minimum, mean and maximum pseudorapidity
  3. Minimum, mean and maximum centrality

Available variables

Università degli Studi di Padova

For each event, the following variables are available:

Response variable

binary variable, y_i, encoding signal ($y_i = 1$) or background ($y_i = 0$)

Kinematic explanatory variables

- Variables related to the 4 selected jets and to the couples of dijets
  1. Transverse momentum
  2. Pseudorapidity
  3. Centrality
- Variables related to non selected jets
  1. Minimum, mean and maximum transverse momentum
  2. Minimum, mean and maximum pseudorapidity
  3. Minimum, mean and maximum centrality
- Variables related to pairs of jets

Available variables

For each event, the following variables are available:

Response variable

binary variable, y_i, encoding signal ($y_i = 1$) or background ($y_i = 0$)

Kinematic explanatory variables

- Variables related to the 4 selected jets and to the couples of dijets
  1. Transverse momentum
  2. Pseudorapidity
  3. Centrality
- Variables related to non selected jets
  1. Minimum, mean and maximum transverse momentum
  2. Minimum, mean and maximum pseudorapidity
  3. Minimum, mean and maximum centrality
- Variables related to pairs of jets
- Other variables

Available variables

Variables of the 4 selected jets and to the couples of dijets

Name	Variable
QP_t1	Transverse momentum related to the first jet
QP_t2	Transverse momentum related to the second jet
QP_t3	Transverse momentum related to the third jet
QP_t4	Transverse momentum related to the fourth jet
$QEta1$	Pseudorapidity related to the first jet
$QEta2$	Pseudorapidity related to the second jet
$QEta3$	Pseudorapidity related to the third jet
$QEta4$	Pseudorapidity related to the fourth jet
$QCMVA1$	CMVA related to the first jet
$QCMVA2$	CMVA related to the second jet
$QCMVA3$	CMVA related to the third jet
$QCMVA4$	CMVA related to the fourth jet
$QCent$	Centrality of the 4 jets

Available variables

Variables of non selected jets

Name	Variable
$AP_t min$	minimum p_t among non selected jets
$AP_t mean$	mean p_t among non selected jets
$AP_t max$	maximum p_t among non selected jets
AEtamin	minimum η among non selected jets
AEtamean	mean η among non selected jets
AEtamax	maximum η among non selected jets
ACMVAmin	minimum CMVA among non selected jets
ACMVAmean	mean CMVA among non selected jets
ACMVAmax	maximum CMVA among non selected jets
ACent	centrality of non selected jets

Variables of pairs of jets, corresponding to each Higgs

Name	Variable
$DJ1mass$	mass of the first dijet
$DJ1P_t$	p_t of the first dijet
$DJ1Phi$	$\Delta\Phi$ of the first dijet
$DJ1Eta$	$\Delta\eta$ of the first dijet
$DJ1R$	ΔR of the first dijet
τ_1	twist of the first dijet
$DJ2mass$	mass of the second dijet
$DJ2P_t$	p_t of the second dijet
$DJ2Phi$	$\Delta\Phi$ of the second dijet
$DJ2Eta$	$\Delta\eta$ of the second dijet
$DJ2R$	ΔR of the second dijet
τ_2	twist of the second dijet

Other variables

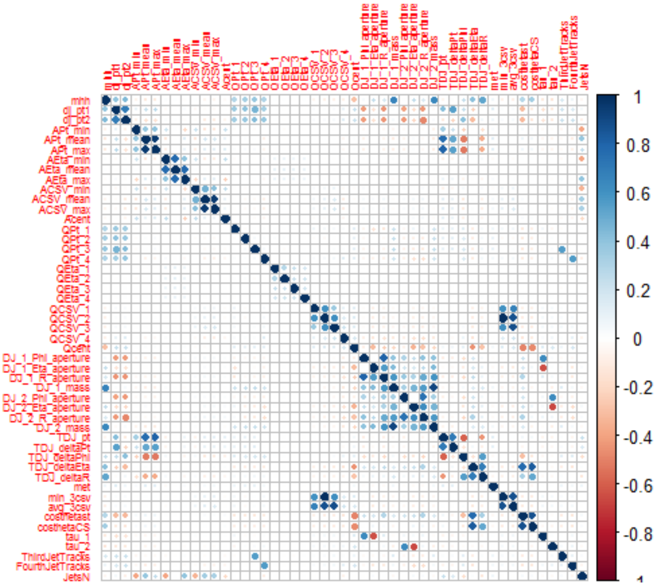
Name	Variable
$TDJP_t$	vectorial sum of the P_t of the two dijet
$TDJ\Delta\Phi$	$\Delta\Phi$ between the two djets
$TDJ\Delta\eta$	$\Delta\eta$ between the two djets
$TDJ\Delta R$	ΔR between the two djets
HHM	invariant mass of the two djets
MET	Missing transverse energy
min3cmva	minimum CMVA among the first 3 jets
avg3cmva	mean CMVA among the first 3 jets
$cos\theta^*$	cosinus of θ on the c.o.m reference system of the two H
$cos\theta_{CS}$	cosinus of θ on the Collins Saper reference system
JetsN	number of jets in the event

Obtained variable

Name	Variable
$sumQP_t$	sum of the transverse momenta of the selected 4 jets

Available variables: Correlation Plot

Università degli Studi di Padova



Approaches

Two approaches

Approaches

Two approaches

- Choice of 9 most predictive variables.
  To favour interpretation and contain error propagation
- Use of all 39 available variables: focus on the best classification

Two approaches

- Choice of 9 most predictive variables.
  To favour interpretation and contain error propagation
- Use of all 39 available variables: focus on the best classification

Estimation strategy

- Training set: 50 000 balanced events
- Test set: 16 000
- Validation set: 16 000

Statistical learning models

We consider a number of typical statistical learning models to best classify signal and background.
Linear and logistic regression, MARS, GAM, CART, Neural nets, Projection pursuit, etc. (e.g., Azzalini and Scarpa, 2012).

We consider a number of typical statistical learning models to best classify signal and background.
Linear and logistic regression, MARS, GAM, CART, Neural nets, Projection pursuit, etc. (e.g., Azzalini and Scarpa, 2012).

Among the models with best performance on the test set:

- Random forests
- Gradient boosting
- Boosting decision tree

Random forests (Breiman, 1999)

- Refinement of bagged trees; quite popular

Random forests (Breiman, 1999)

- Refinement of bagged trees; quite popular
- At each *tree split*, a random sample of m features (variables) is drawn, and only those m features are considered for splitting (Typically $m = \sqrt{p}$ or $\log_2 p$, where p is the number of features)

Random forests (Breiman, 1999)

- Refinement of bagged trees; quite popular
- At each *tree split*, a random sample of m features (variables) is drawn, and only those m features are considered for splitting (Typically $m = \sqrt{p}$ or $\log_2 p$, where p is the number of features)
- For each tree grown on a bootstrap sample, the error rate for observations left out of the bootstrap sample is monitored (out-of-bag)

Random forests (Breiman, 1999)

- Refinement of bagged trees; quite popular
- At each *tree split*, a random sample of m features (variables) is drawn, and only those m features are considered for splitting (Typically $m = \sqrt{p}$ or $\log_2 p$, where p is the number of features)
- For each tree grown on a bootstrap sample, the error rate for observations left out of the bootstrap sample is monitored (out-of-bag)
- Random forests tries to improve on bagging by "de-correlating" the trees, and reduce the variance.

Random forests (Breiman, 1999)

- Refinement of bagged trees; quite popular
- At each *tree split*, a random sample of m features (variables) is drawn, and only those m features are considered for splitting (Typically $m = \sqrt{p}$ or $\log_2 p$, where p is the number of features)
- For each tree grown on a bootstrap sample, the error rate for observations left out of the bootstrap sample is monitored (out-of-bag)
- Random forests tries to improve on bagging by "de-correlating" the trees, and reduce the variance.
- Each tree has the same expectation, so increasing the number of trees does not alter the bias of bagging or random forests.

Random forests (Breiman, 1999)

- Refinement of bagged trees; quite popular
- At each *tree split*, a random sample of m features (variables) is drawn, and only those m features are considered for splitting (Typically $m = \sqrt{p}$ or $\log_2 p$, where p is the number of features)
- For each tree grown on a bootstrap sample, the error rate for observations left out of the bootstrap sample is monitored (out-of-bag)
- Random forests tries to improve on bagging by "de-correlating" the trees, and reduce the variance.
- Each tree has the same expectation, so increasing the number of trees does not alter the bias of bagging or random forests.

Bias-variance trade off

- *The small m, the lower the variance of the random forest ensemble*
- *Small m is also associated with higher bias, because important variables can be missed by the sampling*

Interactions

- Random forests overemphasizes the role of interactions

Interactions

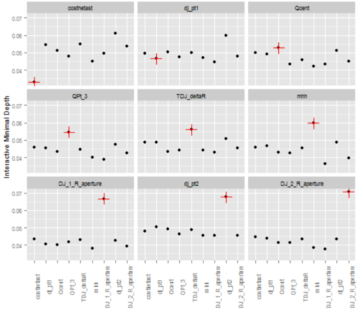
- Random forests overemphasizes the role of interactions
- What are the most interactive kinematic variables in predicting signal from background?

Interactions

- Random forests overemphasizes the role of interactions
- What are the most interactive kinematic variables in predicting signal from background?
- Maximal subtree analysis (Ishwaran et al., 2010,2011) for identify interactions. Smaller minimum depth of a variable j with respect to the maximal subtree for variable i indicates interactions
- we choose to select couples of variables with maximal subtree smaller than 0.045.

Interactions

- Random forests overemphasizes the role of interactions
- What are the most interactive kinematic variables in predicting signal from background?
- Maximal subtree analysis (Ishwaran et al., 2010,2011) for identify interactions. Smaller minimum depth of a variable *j* with respect to the maximal subtree for variable *i* indicates interactions
- we choose to select couples of variables with maximal subtree smaller than 0.045.



$DJ1P_t$	$DJ1R$	$QCent$	QP_t3
QP_t3	$TDJ\Delta R$	QP_t3	mhh
$TDJ\Delta R$	mhh	$TDJ\Delta R$	$DJ1R$
$DJ1R$	$DJ2P_t$	$DJ1R$	$DJ2R$
$QCent$	$DJ1R$	QP_t3	$DJ2R$
$QCent$	mhh	mhh	$DJ2R$
mhh	$DJ1R$	QP_t3	$DJ1R$

Boosting (Freund e Schapire, 1997)

- *Idea*: Put more weight on observations that are misclassified, to make the classifier work harder on those points

Boosting (Freund e Schapire, 1997)

- *Idea*: Put more weight on observations that are misclassified, to make the classifier work harder on those points
- Average many trees, each grown to reweighted versions of the training data.

Boosting (Freund e Schapire, 1997)

- *Idea*: Put more weight on observations that are misclassified, to make the classifier work harder on those points
- Average many trees, each grown to reweighted versions of the training data.
- Weighting *decorrelates* the trees, by focussing on regions missed by past trees.

Boosting (Freund e Schapire, 1997)

- *Idea*: Put more weight on observations that are misclassified, to make the classifier work harder on those points
- Average many trees, each grown to reweighted versions of the training data.
- Weighting *decorrelates* the trees, by focussing on regions missed by past trees.
- Final classifier is weighted average of classifiers

$$C(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m C_m(x)\right]$$

Note: $C_m(x) \in \{-1, +1\}$.

Boosting (Freund e Schapire, 1997)

Università degli Studi di Padova

- *Idea*: Put more weight on observations that are misclassified, to make the classifier work harder on those points
- Average many trees, each grown to reweighted versions of the training data.
- Weighting *decorrelates* the trees, by focussing on regions missed by past trees.
- Final classifier is weighted average of classifiers

$$C(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m C_m(x)\right]$$

  Note: $C_m(x) \in \{-1, +1\}$.
- Boosting Decision Tree (BDT; Drucker, 1997): each error is normalized with the maximum of errors

Adaboost

Details

- Start with equal observation weights $p_i = 1/n$
- At iteration t, draw a bootstrap sample with the current probabilities $p_1, p_2, \ldots, p_n$, compute the classifier and e_t, the error rate of the classifier on the original sample (for BDT normalized with the maximum error).

  Let $\beta_t = e_t/(1 - e_t)$
- For those points that are classified correctly, decrease their probabilities by

$$p_i \leftarrow p_i \cdot \beta_t$$

  and renormalise them
- Do this for many (say 1000) iterations.

At the end, take a weighted vote of the classifications, with weights $\log(1/\beta_t)$ (more weight on classifiers with lower error).

Boosting can improve bagging in many instances

Gradient boosting (Friedman, 2002)

- Gradient boosting builds additive tree models, for example, for representing logits in logistic regression.

Gradient boosting (Friedman, 2002)

- Gradient boosting builds additive tree models, for example, for representing logits in logistic regression.
- *Tree size* is a parameter that determines the order of interaction

Gradient boosting (Friedman, 2002)

- Gradient boosting builds additive tree models, for example, for representing logits in logistic regression.
- *Tree size* is a parameter that determines the order of interaction
- Gradient Boosting inherits all the good features of trees (variable selection, missing data, mixed predictors), and improves on the weak features, such as prediction performance.

Gradient boosting (Friedman, 2002)

- Gradient boosting builds additive tree models, for example, for representing logits in logistic regression.
- *Tree size* is a parameter that determines the order of interaction
- Gradient Boosting inherits all the good features of trees (variable selection, missing data, mixed predictors), and improves on the weak features, such as prediction performance.
- Idea: Fit a weighted additive model (ensemble) in a forward stage-wise manner.

Gradient boosting (Friedman, 2002)

Università degli Studi di Padova

- Gradient boosting builds additive tree models, for example, for representing logits in logistic regression.
- *Tree size* is a parameter that determines the order of interaction
- Gradient Boosting inherits all the good features of trees (variable selection, missing data, mixed predictors), and improves on the weak features, such as prediction performance.
- Idea: Fit a weighted additive model (ensemble) in a forward stage-wise manner.
- In each stage, introduce a 'weak learner' to compensate the shortcomings of existing weak learners.

Gradient boosting (Friedman, 2002)

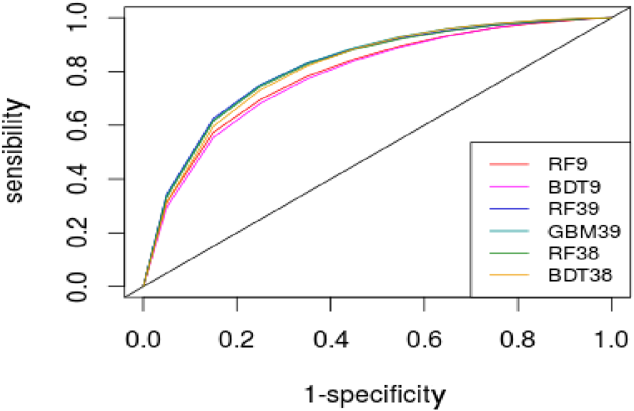
- Gradient boosting builds additive tree models, for example, for representing logits in logistic regression.
- *Tree size* is a parameter that determines the order of interaction
- Gradient Boosting inherits all the good features of trees (variable selection, missing data, mixed predictors), and improves on the weak features, such as prediction performance.
- Idea: Fit a weighted additive model (ensemble) in a forward stage-wise manner.
- In each stage, introduce a ‘weak learner’ to compensate the shortcomings of existing weak learners.
- in Gradient Boosting, ‘shortcomings’ are identified by gradients (in Adaboost, ‘shortcomings’ are identified by high-weight data points).

Gradient boosting (Friedman, 2002)

Università degli Studi di Padova

- Gradient boosting builds additive tree models, for example, for representing logits in logistic regression.
- *Tree size* is a parameter that determines the order of interaction
- Gradient Boosting inherits all the good features of trees (variable selection, missing data, mixed predictors), and improves on the weak features, such as prediction performance.
- Idea: Fit a weighted additive model (ensemble) in a forward stage-wise manner.
- In each stage, introduce a 'weak learner' to compensate the shortcomings of existing weak learners.
- in Gradient Boosting, 'shortcomings' are identified by gradients (in Adaboost, 'shortcomings' are identified by high-weight data points).
- Both high-weight data points and gradients tell us how to improve the model.

ROC curve of the principal statistical learning classification models



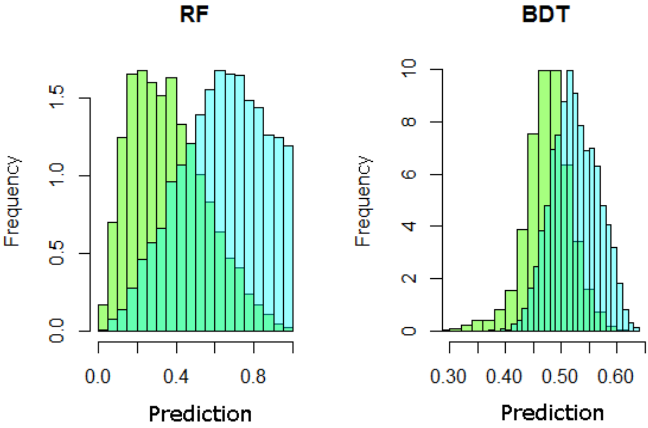
Performances of the best statistical learning classification models

Model	Error test set	Error validation set	AUC test set	AUC validation set
BDT9	0.2821	0.2695	0.7969	0.7934
RF9	0.2756	0.2685	0.7982	0.8004
GBM9	0.2851	0.2858	0.7888	0.7911
BDT38	0.2596	0.2637	0.8204	0.8227
RF38	0.2540	0.2521	0.8224	0.8263
BDT39	0.2340	0.2464	0.8349	0.8400
RF39	0.2320	0.2424	0.8369	0.8424
GBM39	0.2431	0.2477	0.8278	0.8328

Performances of the best statistical learning classification models

Model	Error test set	Error validation set	AUC test set	AUC validation set
RF9	0.2756	0.2685	0.7982	0.8004
BDT9	0.2821	0.2695	0.7969	0.7934
GBM9	0.2851	0.2858	0.7888	0.7911
RF38	0.2540	0.2521	0.8224	0.8263
BDT38	0.2596	0.2637	0.8204	0.8227
RF39	0.2320	0.2424	0.8369	0.8424
BDT39	0.2340	0.2464	0.8349	0.8400
GBM39	0.2431	0.2477	0.8278	0.8328

Comparison RF and BDT



Green: background
Blue: signal

A generalized mixed effects model

- Let y_i be the binary variable encoding signal or background,
- the classical generalized mixed model formulation assumes

$$
\begin{aligned}
y_i|\pi_i &\sim \text{Bern}(\pi_i) \\
logit(\pi_i) &= \eta_i \\
\eta_i &= \mu_i + f(\mathbf{x}_i)
\end{aligned}
$$

where

- $\mathbf{x}_i$ is the vector including all the explanatory variables for each event i
- β is a vector of parameters
- μ_i is a random effect for each event
- p is the number of available explanatory variables

A generalized mixed effects model

- Let y_i be the binary variable encoding signal or background,
- the classical generalized mixed model formulation assumes

$$
\begin{aligned}
y_i|\pi_i &\sim \text{Bern}(\pi_i) \\
logit(\pi_i) &= \eta_i \\
\eta_i &= \mu_i + f(\mathbf{x}_i)
\end{aligned}
$$

  where

  - $\mathbf{x}_i$ is the vector including all the explanatory variables for each event i
  - β is a vector of parameters
  - μ_i is a random effect for each event
  - p is the number of available explanatory variables

- Bayesian approach, assuming *priors* on parameters.
  (*still "intrinsically" frequentist* - not a *subjective* approach)

- **Dirichlet process** (DP):
  assume $\mu_i \sim P$ with $P \sim DP(\alpha P_0)$, $\alpha > 0$
  We also assume fixed effects for explanatory variables $f(\mathbf{x}_i) = \mathbf{x}_i^T \beta$

- **Dirichlet process** (DP):
  assume $\mu_i \sim P$ with $P \sim DP(\alpha P_0)$, $\alpha > 0$
  We also assume fixed effects for explanatory variables $f(\mathbf{x}_i) = \mathbf{x}_i^T \beta$
- **Additive model with P-splines**

$$f(\mathbf{x}_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

$f_1(\cdot), \ldots, f_p(\cdot)$ fitted via Bayesian P-splines
and μ_i assumed constant and fixed

Bayesian non parametric models - BNP

- **Dirichlet process** (DP):
  assume $\mu_i \sim P$ with $P \sim DP(\alpha P_0)$, $\alpha > 0$
  We also assume fixed effects for explanatory variables $f(\mathbf{x}_i) = \mathbf{x}_i^T \beta$
- **Additive model with *P*-splines**

$$f(\mathbf{x}_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

  $f_1(\cdot), \ldots, f_p(\cdot)$ fitted via Bayesian P-splines
  and μ_i assumed constant and fixed
- **BART model (Bayesian Additive Regression Tree)**

$$f(\mathbf{x}_i) = \sum_{j=1}^{m} g(\mathbf{x}_i; T_j, M_j)$$

  where $g(\mathbf{x}_i, T_j, M_j)$ denotes the predicting function assigning a value to $\mathbf{x}_i$ given the Bayesian tree T_j and parameters M_j
  μ_i assumed constant and fixed.

Mixed effect model with Dirichlet process

Let

$$\eta_i = \mu_i + \mathbf{x}_i^T \beta$$

Taking a Bayesian approach we specify prior distributions for the parameters (μ_i, β)

Let

$$\eta_i = \mu_i + \mathbf{x}_i^T \beta$$

Taking a Bayesian approach we specify prior distributions for the parameters (μ_i, β)

- to provide a flexible formulation for adaptively modeling the determinants of signal

Mixed effect model with Dirichlet process

Let

$$\eta_i = \mu_i + \mathbf{x}_i^T \beta$$

Taking a Bayesian approach we specify prior distributions for the parameters (μ_i, β)

- to provide a flexible formulation for adaptively modeling the determinants of signal
- to allow uncertainty in the distribution of the random intercepts, while avoiding over-shrinking and favouring clustering effects

Let

$$\eta_i = \mu_i + \mathbf{x}_i^T \beta$$

Taking a Bayesian approach we specify prior distributions for the parameters (μ_i, β)

- to provide a flexible formulation for adaptively modeling the determinants of signal
- to allow uncertainty in the distribution of the random intercepts, while avoiding over-shrinking and favouring clustering effects
- needs for an efficient algorithm for posterior computation

Mixed effect model with Dirichlet process

Università degli Studi di Padova

Let

$$\eta_i = \mu_i + \mathbf{x}_i^T \beta$$

Taking a Bayesian approach we specify prior distributions for the parameters (μ_i, β)

- to provide a flexible formulation for adaptively modeling the determinants of signal
- to allow uncertainty in the distribution of the random intercepts, while avoiding over-shrinking and favouring clustering effects
- needs for an efficient algorithm for posterior computation

Priors distributions

- $\beta \sim \mathcal{N}_p(\mathbf{b}, \mathbf{B})$
- $\mathbf{b}_{p\times 1}, \mathbf{B}_{p\times p}$ are prior mean vector and covariance matrix
- $\mu_i \sim P$ with $P \sim DP(\alpha P_0)$, $\alpha > 0$,
  where DP indicates the Dirichlet Process.

The Dirichlet process $DP(\alpha P_0)$ represents a fully flexible prior with support on the set of distributions on the real line, allowing P to be unknown with

- P_0 indicating the best guess for such distribution and
- α quantifying the confidence in such guess.

In our case, we define P_0 as a normal distribution $\mathcal{N}(0, \sigma^2)$ where

- $\sigma^{-2} \sim Gamma(a, b)$ (i.e. prior for σ is inverse Gamma)
- $\alpha \sim Gamma(a_\alpha, b_\alpha)$ to favor learning of cluster effects in the data

We exploit the stick-breaking representation of the Dirichlet Process

Stick-breaking representation (Sethuraman, 1994)

Let

$$V_h \overset{iid}{\sim} Beta(1, \alpha) \qquad \theta_h \overset{iid}{\sim} G_0$$

$$\pi_h \sim V_h \prod_{l<h} (1 - V_l) \qquad G = \sum_{h=1}^{\infty} \pi_h \delta_{\theta_h}$$

therefore $G \sim DP(\alpha, G_0)$, where δ_θ indicates a mass point concentrated in θ.

Dirichlet Process

We exploit the stick-breaking representation of the Dirichlet Process

Stick-breaking representation (Sethuraman, 1994)

Let

$$V_h \overset{iid}{\sim} Beta(1, \alpha) \qquad \theta_h \overset{iid}{\sim} G_0$$

$$\pi_h \sim V_h \prod_{l<h}(1 - V_l) \qquad G = \sum_{h=1}^{\infty} \pi_h \delta_{\theta_h}$$

therefore $G \sim DP(\alpha, G_0)$, where δ_θ indicates a mass point concentrated in θ.

- Key result: a realization of the Dirichlet process is discrete in nature

We exploit the stick-breaking representation of the Dirichlet Process

Stick-breaking representation (Sethuraman, 1994)

Let

$$V_h \overset{iid}{\sim} Beta(1, \alpha) \qquad \theta_h \overset{iid}{\sim} G_0$$

$$\pi_h \sim V_h \prod_{l<h}(1 - V_l) \qquad G = \sum_{h=1}^{\infty} \pi_h \delta_{\theta_h}$$

therefore $G \sim DP(\alpha, G_0)$, where δ_θ indicates a mass point concentrated in θ.

- Key result: a realization of the Dirichlet process is discrete in nature
- It favours ties among random intercepts: events in the same cluster have equal random intercept values

Dirichlet Process

We exploit the stick-breaking representation of the Dirichlet Process

Stick-breaking representation (Sethuraman, 1994)

Let

$$V_h \overset{iid}{\sim} Beta(1, \alpha) \qquad \theta_h \overset{iid}{\sim} G_0$$

$$\pi_h \sim V_h \prod_{l<h} (1 - V_l) \qquad G = \sum_{h=1}^{\infty} \pi_h \delta_{\theta_h}$$

therefore $G \sim DP(\alpha, G_0)$, where δ_θ indicates a mass point concentrated in θ.

- Key result: a realization of the Dirichlet process is discrete in nature
- It favours ties among random intercepts: events in the same cluster have equal random intercept values
- Denoting with S_i the grouping variable, the stick-breaking representation shows clustering effects among events, providing $\mu_i = \theta_{S_i}$, with the number of clusters stochastically increasing with α

We exploit the stick-breaking representation of the Dirichlet Process

Stick-breaking representation (Sethuraman, 1994)

Let

$$V_h \stackrel{iid}{\sim} Beta(1, \alpha) \qquad \theta_h \stackrel{iid}{\sim} G_0$$

$$\pi_h \sim V_h \prod_{l<h} (1 - V_l) \qquad G = \sum_{h=1}^{\infty} \pi_h \delta_{\theta_h}$$

therefore $G \sim DP(\alpha, G_0)$, where δ_θ indicates a mass point concentrated in θ.

- Key result: a realization of the Dirichlet process is discrete in nature
- It favours ties among random intercepts: events in the same cluster have equal random intercept values
- Denoting with S_i the grouping variable, the stick-breaking representation shows clustering effects among events, providing $\mu_i = \theta_{S_i}$, with the number of clusters stochastically increasing with α
- This clustering property is particularly useful in our signal detection, favouring events with common kinematic features to share the same effect

Dirichlet Process

Università degli Studi di Padova

We exploit the stick-breaking representation of the Dirichlet Process

Stick-breaking representation (Sethuraman, 1994)

Let

$$V_h \overset{iid}{\sim} Beta(1, \alpha) \qquad \theta_h \overset{iid}{\sim} G_0$$

$$\pi_h \sim V_h \prod_{l<h} (1 - V_l) \qquad G = \sum_{h=1}^{\infty} \pi_h \delta_{\theta_h}$$

therefore $G \sim DP(\alpha, G_0)$, where δ_θ indicates a mass point concentrated in θ.

- Key result: a realization of the Dirichlet process is discrete in nature
- It favours ties among random intercepts: events in the same cluster have equal random intercept values
- Denoting with S_i the grouping variable, the stick-breaking representation shows clustering effects among events, providing $\mu_i = \theta_{S_i}$, with the number of clusters stochastically increasing with α
- This clustering property is particularly useful in our signal detection, favouring events with common kinematic features to share the same effect
- Conditionally on the grouping indicator S_i, the Gaussian base measure P_0 is conjugate, favoring the implementation of a Gibbs sampler

For posterior computation we exploit

- *blocked Gibbs sampler* algorithm by Ishwaran and James (2001)

Posterior computation

Università degli Studi di Padova

For posterior computation we exploit

- *blocked Gibbs sampler* algorithm by Ishwaran and James (2001)
- a recently proposed data-augmentation scheme based on Pólya-Gamma (PG) distribution

Pólya-Gamma data-augmentation

Assuming a Bayesian logistic regression setting where $y_i \sim Bern(1/[1+e^{-\phi_i}]), i = 1, ..., n, \phi_i = x_i^T \beta$ and $\beta \sim \mathcal{N}_p(\mathbf{b}, \mathbf{B})$, the resulting Gibbs sampler alternates between two full conditional conjugate steps

- $\omega_i \sim PG(1, x_i^T \beta)$
- $\beta | y, \omega, x \sim \mathcal{N}_p(\mu_\beta, \Sigma_\beta)$

where $\Sigma_\beta = (X^T \Omega X + B^{-1})$ and $\mu_\beta = \Sigma_\beta (X^T z + B^{-1b})$, $z = [y_1 - 1/2, \ldots, y_n - 1/2]$ and $\Omega = \text{diag}(\omega_1, \ldots, \omega_n)$

Results

	Test set classification error	False positives	False negatives	AUC
Logistic BNP	0.492	0.4121	0.5663	0.5118

Results

	Test set classification error	False positives	False negatives	AUC
Logistic BNP	0.492	0.4121	0.5663	0.5118
RF39	0.320	0.2678	0.3678	0.7393

- Nonlinearities are not allowed (variables are related to response by linear functions)

Drawbacks of the Dirichlet process model

- Nonlinearities are not allowed (variables are related to response by linear functions)
- Interactions among variables are not allowed

Drawbacks of the Dirichlet process model

Università degli Studi di Padova

- Nonlinearities are not allowed (variables are related to response by linear functions)
- Interactions among variables are not allowed

back to the generalized mixed model

$$
\begin{aligned}
y_i|\pi_i &\sim \text{Bern}(\pi_i) \\
logit(\pi_i) &\sim \eta_i \\
\eta_i &\sim \mu_i + f(\mathbf{x}_i)
\end{aligned}
$$

Additive model with P-splines

Additive model with P-splines

- Bayesian additive model

$$f(\mathbf{x}_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

$f_1(\cdot), \dots, f_p(\cdot)$ estimated via Bayesian P-splines, and $\mu_i = 0$.

Additive model with P-splines

- Bayesian additive model

$$f(\mathbf{x}_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

  $f_1(\cdot), \ldots, f_p(\cdot)$ estimated via Bayesian P-splines, and $\mu_i = 0$.

- P-splines (Bayesian Penalized Splines, Lang and Brezger, 2004)

$$f_j(x_j) = \sum_{r=1}^{M_j} \beta_{jr} B_{jr}(x_j)$$

  where B_{jr} is the r-th base function and $\beta_j = (\beta_{j1}, \ldots, \beta_{jM_j})$ is a parameter vector.

Additive model with *P*-splines

- Bayesian additive model

$$f(\mathbf{x}_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

$f_1(\cdot), \ldots, f_p(\cdot)$ estimated via Bayesian P-splines, and $\mu_i = 0$.

- *P*-splines (Bayesian Penalized Splines, Lang and Brezger, 2004)

$$f_j(x_j) = \sum_{r=1}^{M_j} \beta_{jr} B_{jr}(x_j)$$

where B_{jr} is the r-th base function and $\beta_j = (\beta_{j1}, \ldots, \beta_{jM_j})$ is a parameter vector.

- Include interactions identified with Random Forests, as new variables.

Additive model with P-splines

- Bayesian additive model

$$f(\mathbf{x}_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

$f_1(\cdot), \ldots, f_p(\cdot)$ estimated via Bayesian P-splines, and $\mu_i = 0$.

- P-splines (Bayesian Penalized Splines, Lang and Brezger, 2004)

$$f_j(x_j) = \sum_{r=1}^{M_j} \beta_{jr} B_{jr}(x_j)$$

where B_{jr} is the r-th base function and $\beta_j = (\beta_{j1}, \ldots, \beta_{jM_j})$ is a parameter vector.

- Include interactions identified with Random Forests, as new variables.

	Test set classification error	False positives	False negatives	AUC
P-splines	0.3400	0.3975	0.2874	0.7102

Additive model with P-splines

- Bayesian additive model

$$f(\mathbf{x}_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

$f_1(\cdot), \ldots, f_p(\cdot)$ estimated via Bayesian P-splines, and $\mu_i = 0$.

- P-splines (Bayesian Penalized Splines, Lang and Brezger, 2004)

$$f_j(x_j) = \sum_{r=1}^{M_j} \beta_{jr} B_{jr}(x_j)$$

where B_{jr} is the r-th base function and $\beta_j = (\beta_{j1}, \ldots, \beta_{jM_j})$ is a parameter vector.

- Include interactions identified with Random Forests, as new variables.

	Test set classification error	False positives	False negatives	AUC
P-splines	0.3400	0.3975	0.2874	0.7102
RF39	0.320	0.2678	0.3678	0.7393

BART - Bayesian Additive Regression Tree

- BART (Chipman, George and McCulloch, 2010)

$$f(\mathbf{x}_i) = \sum_{j=1}^{m} g(\mathbf{x}_i; T_j, M_j)$$

where $g(\mathbf{x}_i, T_j, M_j)$ denotes the predicting function assigning a value to $\mathbf{x}_i$ given the Bayesian tree T_j and parameters M_j

BART - Bayesian Additive Regression Tree

- BART (Chipman, George and McCulloch, 2010)

$$f(\mathsf{x}_i) = \sum_{j=1}^{m} g(\mathsf{x}_i; T_j, M_j)$$

  where $g(\mathsf{x}_i, T_j, M_j)$ denotes the predicting function assigning a value to x_i given the Bayesian tree T_j and parameters M_j
- Sum of trees where variable selection at each node, cut points and depth are parameters (needing for a prior distribution).

BART - Bayesian Additive Regression Tree

- BART (Chipman, George and McCulloch, 2010)

$$f(\mathsf{x}_i) = \sum_{j=1}^{m} g(\mathsf{x}_i; T_j, M_j)$$

  where $g(\mathsf{x}_i, T_j, M_j)$ denotes the predicting function assigning a value to x_i given the Bayesian tree T_j and parameters M_j
- Sum of trees where variable selection at each node, cut points and depth are parameters (needing for a prior distribution).
- To fit a BART: tailored version of Bayesian backfitting MCMC (Hastie and Tibshirani, 2000) that iteratively constructs and fits successive residuals

BART - Bayesian Additive Regression Tree

- BART (Chipman, George and McCulloch, 2010)

$$f(\mathbf{x}_i) = \sum_{j=1}^{m} g(\mathbf{x}_i; T_j, M_j)$$

  where $g(\mathbf{x}_i, T_j, M_j)$ denotes the predicting function assigning a value to $\mathbf{x}_i$ given the Bayesian tree T_j and parameters M_j
- Sum of trees where variable selection at each node, cut points and depth are parameters (needing for a prior distribution).
- To fit a BART: tailored version of Bayesian backfitting MCMC (Hastie and Tibshirani, 2000) that iteratively constructs and fits successive residuals
- Interactions are included in the tree model

BART - Bayesian Additive Regression Tree

- BART (Chipman, George and McCulloch, 2010)

$$f(\mathbf{x}_i) = \sum_{j=1}^{m} g(\mathbf{x}_i; T_j, M_j)$$

  where $g(\mathbf{x}_i, T_j, M_j)$ denotes the predicting function assigning a value to $\mathbf{x}_i$ given the Bayesian tree T_j and parameters M_j
- Sum of trees where variable selection at each node, cut points and depth are parameters (needing for a prior distribution).
- To fit a BART: tailored version of Bayesian backfitting MCMC (Hastie and Tibshirani, 2000) that iteratively constructs and fits successive residuals
- Interactions are included in the tree model

	Test set classification error	False positives	False negatives	AUC
BART	0.3480	0.3083	0.3846	0.7078

BART - Bayesian Additive Regression Tree

- BART (Chipman, George and McCulloch, 2010)

$$f(\mathbf{x}_i) = \sum_{j=1}^{m} g(\mathbf{x}_i; T_j, M_j)$$

  where $g(\mathbf{x}_i, T_j, M_j)$ denotes the predicting function assigning a value to $\mathbf{x}_i$ given the Bayesian tree T_j and parameters M_j
- Sum of trees where variable selection at each node, cut points and depth are parameters (needing for a prior distribution).
- To fit a BART: tailored version of Bayesian backfitting MCMC (Hastie and Tibshirani, 2000) that iteratively constructs and fits successive residuals
- Interactions are included in the tree model

	Test set classification error	False positives	False negatives	AUC
BART	0.3480	0.3083	0.3846	0.7078
RF39	0.320	0.2678	0.3678	0.7393

- **Dirichlet process and P-splines**

$$f(\mathbf{x}_i) = \mu_i + f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

where μ_i is a DP with Gaussian atoms, and the f_j are Bayesian P-splines

Combinations of models

Università degli Studi di Padova

- **Dirichlet process and P-splines**

$$f(\mathbf{x}_i) = \mu_i + f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

where μ_i is a DP with Gaussian atoms, and the f_j are Bayesian P-splines

- **Dirichlet process with BART athoms**
the atoms of the Dirichlet process depends on the $\mathbf{x}_i$ and are described by a BART.

Combinations of models

- **Dirichlet model with BART atoms and P-splines**

$$f(\mathbf{x}_i) = \mu_i + f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

where μ_i is a DP with atoms described by a BART and f_j are Bayesian P-splines

Combinations of models

- **Dirichlet model with BART atoms and *P*-splines**

$$f(\mathbf{x}_i) = \mu_i + f_1(x_{i1}) + \cdots + f_p(x_{ip})$$

where μ_i is a DP with atoms described by a BART and f_j are Bayesian *P*-splines

- **BART with *P*-SPLINES**

Si stima il modello

$$f(\mathbf{x}_i) = f_1(x_{i1}) + \cdots + f_p(x_{ip}) + \sum_{j=1}^{m} g(x; T_j, M_j)$$

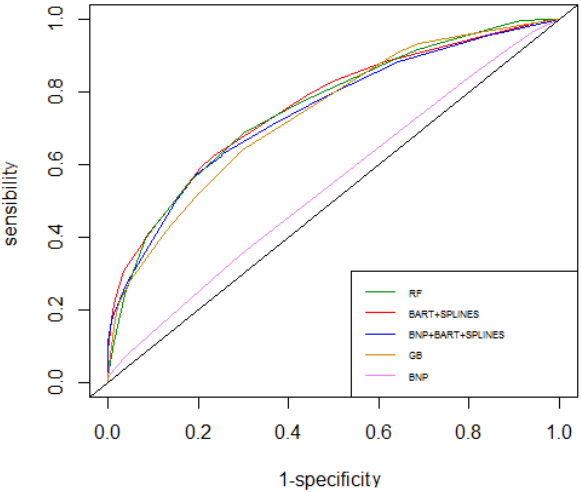
where f_j are estimated via Bayesian *P*-splines, and sum of g is a BARTmodel.

Results

Model	Classification Error	False Positives	False Negatives	AUC
RF	0.3200	0.2678	0.3678	0.7393
GB	0.3300	0.2971	0.3602	0.7234
BDT	0.3351	0.3461	0.3389	0.7198
DP	0.4920	0.4121	0.5663	0.5118
P-splines	0.3400	0.3975	0.2874	0.7102
BART	0.3480	0.3083	0.3846	0.7078
BART as atoms of DP	0.4620	0.4346	0.4867	0.5429
DP+P-splines	0.3300	0.3347	0.3295	0.7259
BART as atoms of DP + P-splines	0.3240	0.2970	0.3455	0.7321
BART+P-splines	0.3140	0.3138	0.3141	0.7417

Model	Classification Error	False Positives	False Negatives	AUC
RF	0.3200	0.2678	0.3678	0.7393
GB	0.3300	0.2971	0.3602	0.7234
BDT	0.3351	0.3461	0.3389	0.7198
DP	0.4920	0.4121	0.5663	0.5118
P-splines	0.3400	0.3975	0.2874	0.7102
BART	0.3480	0.3083	0.3846	0.7078
BART as atoms of DP	0.4620	0.4346	0.4867	0.5429
DP+P-splines	0.3300	0.3347	0.3295	0.7259
BART as atoms of DP + P-splines	0.3240	0.2970	0.3455	0.7321
BART+P-splines	0.3140	0.3138	0.3141	0.7417

ROC curves


sensibility
1-specificity
0.0
0.2
0.4
0.6
0.8
1.0
RF
BART+SPLINES
BNP+BART+SPLINES
GB
BNP

Discussion

- The choice of the most appropriate model can be driven by knowledge of its strength. However, it is always better to fit different models and compare them on different data, in order to choose the best classifier.
- Combination of models may work better than single models if each model has a different strength compared to the others: committees can help when different learners have complementary strengths for a given task.
- Choice of 9 most predictive variables has been done by physicists by looking to marginal correlation and meaning of the variables. Most of models with 39 variables have some procedure of automatic selection of complexity and of variables, compromising between bias and variance. We used a test set to train and select complexity level of each model.