Flavour Tagging in the LHCb experiment
LHCp Conferences 2018 - Bologna

Davide Fazzini
on behalf of the LHCb Collaboration

Università degli Studi di Milano Bicocca & INFN

June 4-9 2018
LHCb detector is a single-arm forward spectrometer optimised for $b$ and $c$ hadron physics
- pseudorapidity range: $[2,5] \Rightarrow \sim 25\% \; b\bar{b}$ pairs in LHCb acceptance

**High precision measurements** in flavour physics
(e.g. CKM, beyond SM)

Collected data:
- Run1 (2010-2012) $\Rightarrow \approx 3 \; \text{fb}^{-1}$
- Run2 (2015-2018) $\Rightarrow \approx 4$ (already taken) + 2 (expected) $\text{fb}^{-1}$

**Excellent performances**
[Int. J. Mod. Phys. A 30, 1520022 (2015)]:
- **Momentum resolution:**
  \[ \frac{\sigma_p}{p} \approx 0.5 - 0.8\% (p < 100 \; \text{GeV}/c) \]
- **Impact Parameter (IP) resolution:**
  \[ \sigma_{IP} \approx 20 \; \mu m \] (at high $p_T$)
- **Decay time resolution:**
  \[ \sigma_t \approx 50 \; \text{fs} \]
- **Particle Identification (PID):**
  \[ \varepsilon(K) \approx 95\%, \pi \text{ mis-ID} \approx 5\% \; (p < 100 \; \text{GeV}/c) \]
  \[ \varepsilon(\mu) \approx 97\%, \pi \text{ mis-ID} \approx 1-3\% \]
Time-dependent measurements for neutral B mesons

- CP violation measurements involve the study of time-dependent rates asymmetries:

\[
A_{CP}(t) = \frac{N(B^0 \to f)(t) - N(B^0 \to f)(t)}{N(B^0 \to f)(t) + N(B^0 \to f)(t)}
\]

- The measurement of this observable is related to the knowledge of the B flavour at production
- Due to neutral mesons oscillations flavour at decay is different from the flavour at production
- **Key ingredient**: *flavour tagging algorithms*

\[
B^0 \to J/\psi K_S^0 \quad [PRL 115, 031601 (2015)]
\]

\[
B_s^0 \to D_s^- \pi^+ \quad [New J. Phys. 15 (2013) 053021]
\]
Flavour Tagging algorithms

- The p-p collisions produce $b\bar{b}$ pairs through strong interactions.
- Particles in the event are used to identify the B flavour.
- These particles can be classified as:
  - "**Same Side**" (SS) if the tagger is coming from the signal B fragmentation
  - "**Opposite Side**" (OS) if the tagger is coming from opposite B decay

<table>
<thead>
<tr>
<th>SS algorithms</th>
<th>OS algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kaon (SSK) [for $B_s^0$]</td>
<td>Muon (OS$\mu$)</td>
</tr>
<tr>
<td>Pion (SS$\pi$) [for $B^0$]</td>
<td>Electron (OSE)</td>
</tr>
<tr>
<td>Proton (SS$p$) [for $B^0$]</td>
<td>Kaon (OSK)</td>
</tr>
<tr>
<td>SS($\pi + p$) combination</td>
<td>Charm (OSC)</td>
</tr>
<tr>
<td>OS $B_s^0$ combination</td>
<td>Vertex Charge (OS Vtx)</td>
</tr>
</tbody>
</table>

**Diagram:**
- PV
- SV
- SS pion
- SS proton
- SS kaon (for $B_s^0$)
- $B^0$
- Same side
- Opposite side
- $h_b$
- $b \rightarrow c$
- $b \rightarrow Xl^-$
- $c \rightarrow s$
- $K^+$
- OS kaon
- OS muon
- OS electron
- OS vertex charge
- OS Charm
- OS $l^-$
**Definitions**

- **Tagging Efficiency**: fraction of tagged events
  \[ \varepsilon_{tag} = \frac{N_{wrong} + N_{right}}{N_{wrong} + N_{right} + N_{untag}} \]
  - correlated with the transverse momentum of the signal B

- **Mistag probability**: fraction of the events with wrong tag decision.
  \[ \omega = \frac{N_{wrong}}{N_{right} + N_{wrong}} \]
  - Defined in range [0, 0.5]
  - **Dilution**: \( D = (1 - 2\omega) \) of asymmetries and decay rates.
  - Predicted mistag probability \( \eta \) computed by taggers needs calibration \( \omega(\eta) \) to provide unbiased estimate of the mistag probability \( \omega \)

- **Tagging power**: statistical degradation of CP asymmetries
  \[ \varepsilon_{eff} = \varepsilon_{tag} D^2 = \varepsilon_{tag} \langle (1 - 2\omega)^2 \rangle \]
  \[ \sigma_{stat}(CP \ asym) \propto \frac{1}{\sqrt{\varepsilon_{eff} N}} \]
FT algorithm development

- FT algorithms are developed following a similar workflow

- **Selection of the tagging candidates:**
  - performed applying rectangular cuts on the most sensitive variables
  - aim: remove most of the background contamination

- **Training of the algorithm:**
  - MVA classifier is used to select the best tagging candidates
  - the charge of the tagging candidate is used to infer the flavour of the signal $B$
  - MVA output turned into a predicted mistag probability

- **Calibration of the mistag probability:**
  - performed on an independent control sample
  - checking the predicted mistag probability $\eta$ match with the mistag probability $\omega$
Flavour tagging in Run2

- FT performance is sensitive to:
  - center-of-mass energy
  - track multiplicity/number of primary vertices
  - trigger efficiency

⇒ differences are expected from Run1 to Run2!

- Using the algorithm optimised on Run1:
  - OS: loss in tagging power
  - SS: small improvements in tagging power

- A re-optimisation was needed, retuning/redesigning the FT algorithms on new data (Run2)
  - \( OS_\mu, OS_e, OS_k & SSK \) taggers revisited and optimised (see next slides)
  - \( OSVtx, OSC, SS_\pi & SS_p \) remained untouched
Re-optimisation of OS taggers: strategy

Tagging candidates selection

- Implementation on $B^+ \to J/\psi K^+$ decay
- Optimise cut-based selection maximizing the average tagging power $\langle \varepsilon_{\text{eff}} \rangle$
  - $\langle \varepsilon_{\text{eff}} \rangle = f(p > x_1, p_T > x_2, ...)$
  - Points $x_i$ determined minimizing $-\langle \varepsilon_{\text{eff}} \rangle$
- At each step, the candidate with higher $p_T$ is taken as tagging particle

![Graph showing the effect of BDT training](image)

**BDT Training**

- BDT trained to discriminate between rightly and wrongly tagged B candidates
- True flavour determined by B charge
- BDT probability turned into a predicted mistag $\eta$
- $\eta$ calibrated to evaluate the performance.
Re-optimisation of OS taggers: performance

- Performance evaluated on Run2 $B^0 \rightarrow D^- \pi^+$ data sample

<table>
<thead>
<tr>
<th>Taggers</th>
<th>$\varepsilon$ [%]</th>
<th>$\omega$ [%]</th>
<th>$\varepsilon \langle D^2 \rangle = \varepsilon (1 - 2\omega)^2$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS$\mu$</td>
<td>0.915 ± 0.053</td>
<td>30.713 ± 0.434</td>
<td>1.361 ± 0.062</td>
</tr>
<tr>
<td>OS$e$</td>
<td>4.451 ± 0.038</td>
<td>34.038 ± 0.604</td>
<td>0.454 ± 0.035</td>
</tr>
<tr>
<td>OS$K$</td>
<td>19.600 ± 0.073</td>
<td>37.557 ± 0.315</td>
<td>1.214 ± 0.061</td>
</tr>
</tbody>
</table>

OS$\mu$ calibration  
OS$e$ calibration  
OS$K$ calibration

- Loss of tagging power recovered!

$\implies$ Stable performances w.r.t. Run1
Other OS taggers & OS combination

- Performance evaluated on Run2 $B^0 \rightarrow D^- \pi^+$ data sample

<table>
<thead>
<tr>
<th>Taggers</th>
<th>$\varepsilon$ [%]</th>
<th>$\omega$ [%]</th>
<th>$\varepsilon\langle D^2 \rangle = \varepsilon(1 - 2\omega)^2$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS Vtx</td>
<td>20.834 ± 0.075</td>
<td>36.994 ± 0.308</td>
<td>1.410 ± 0.067</td>
</tr>
<tr>
<td>OSC</td>
<td>5.025 ± 0.040</td>
<td>34.062 ± 0.620</td>
<td>0.511 ± 0.040</td>
</tr>
<tr>
<td>OS comb</td>
<td>40.154 ± 0.090</td>
<td>35.123 ± 0.211</td>
<td>3.555 ± 0.101</td>
</tr>
</tbody>
</table>

- OS Vtx calibration
- OSC calibration
- OS comb calibration

- Stable performances w.r.t. Run1
Re-optimisation of SSK tagger: strategy

- Fast $B^0_s$ oscillations makes training impossible on data:
  \[ \Rightarrow \] optimisation performed on simulated $B^0_s \rightarrow D^-_s \pi^+$ sample
- Training consist of two classifiers:
  1. separating *fragmentation kaons* from *underlying event tracks*
  2. separating $B^0_s$ and $\bar{B}^0_s$ events and determining tagging decision and mistag probability

![Graph](image)
Re-optimisation of $SSK$ tagger: performance

- $SSK$ calibration and performance evaluated on $B_{s}^{0} \rightarrow D_{s}^{-}\pi^{+}$ data

<table>
<thead>
<tr>
<th>Tagger</th>
<th>$\varepsilon$ [%]</th>
<th>$\omega$ [%]</th>
<th>$\varepsilon \langle D^{2} \rangle$ = $\varepsilon (1 - 2\omega)^{2}$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS$k$</td>
<td>$68.190 \pm 0.177$</td>
<td>$39.667 \pm 0.507$</td>
<td>$2.912 \pm 0.286$</td>
</tr>
</tbody>
</table>

- $SSK$ $\varepsilon \langle D^{2} \rangle$ increase w.r.t. Run1: $+ \sim 45\%$ rel.
Other SS taggers

- Performance evaluated on Run2 $B^0 \rightarrow D^- \pi^+$ data sample

<table>
<thead>
<tr>
<th>Taggers</th>
<th>$\varepsilon$ [%]</th>
<th>$\omega$ [%]</th>
<th>$\varepsilon\langle D^2 \rangle = \varepsilon(1 - 2\omega)^2$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS$\pi$</td>
<td>83.486 ± 0.068</td>
<td>42.561 ± 0.145</td>
<td>1.848 ± 0.072</td>
</tr>
<tr>
<td>SS$p$</td>
<td>37.767 ± 0.089</td>
<td>43.645 ± 0.221</td>
<td>0.610 ± 0.042</td>
</tr>
<tr>
<td>SS comb</td>
<td>87.590 ± 0.061</td>
<td>41.787 ± 0.142</td>
<td>2.364 ± 0.081</td>
</tr>
</tbody>
</table>

- SS$\pi$ calibration
- SS$p$ calibration
- SS comb calibration

- Performance improved by $\sim$ 10% w.r.t Run1
Inclusive tagger: strategy

- Improving classical tagging algorithms is becoming harder and harder
- **New idea**: infer the $b$ flavour at production by exploiting the full event: \( \Rightarrow \) letting the algorithm to choose:
  - which are the tagging candidates
  - how they are connected to $B$ flavour

- Algorithm developed using a **Recursive Neural Network** (RNN):
  - natural approach to handling variable-sized (tracks, vertices)

- **Implementation**:
  - RNN trained using simulated events of $B^+ \rightarrow J/\psi K^+$
  - input variables consist of kinematic, topological, PID and tracking informations
  - the aim is distinguishing between $B^+$ and $B^-$ given a set of tracks
  - developed still on-going
Inclusive tagger: performance

- Comparing inclusive tagger performances w.r.t classical taggers

- Inclusive tagger separation power between $B^+$ and $B^-$

Next steps:
- refining the RNN architecture trying to improve the discriminating power
- applying the algorithm to neutral B meson
Summary and conclusions

- Flavour tagging in LHCb experiment allows precision measurements in b-hadron physics
- Performance depends on data taking condition
- **Successful re-optimisation of FT algorithms**
- **Significant** improvements achieved:
  - OS tagging power now similar to Run1
  - SSK tagging power increased by $\sim 45\%$ w.r.t. Run1
- **Overall FT performances better in Run2 than Run1!**

- New idea under development: **Inclusive tagger**
- Studies are still in a preliminary stage but seem to be very promising
Thank you for your attention!
Backup
Tagger Combination

- If more than one algorithm tag the same signal $B_{0}^{0(d,s)}$ meson, the informations provided by each tagger are combined together

\[ p(b) = \prod_{i} \left( \frac{1+d_i}{2} - d_i(1-\eta_i) \right), \quad p(b) = \prod_{i} \left( \frac{1-d_i}{2} + d_i(1-\eta_i) \right) \]

where

- $p(b)$ ($p(b)$) is the probability that the signal $B_{0}^{0(d,s)}$ contains a $b$-quark ($\bar{b}$-quark)
- $d_i$ is the decision taken by the $i$-th tagger
  - $d_i = 1 \rightarrow$ signal $\bar{b}$-quark, is a $B_{0}^{0(d,s)}$
  - $d_i = -1 \rightarrow$ signal $b$-quark, is a $\bar{B}_{0}^{0(d,s)}$
- $\eta_i$ is the predicted mistag probability of the $i$-th tagger
- The probabilities are normalized as:

\[ P(\bar{b}) = \frac{p(\bar{b})}{p(b)+p(b)}, \quad P(b) = 1 - P(\bar{b}) \]
OS\textsubscript{e} strategy (I)

- Selection optimised minimizing $-\varepsilon_{\text{eff}}$
- Optimisation performed numerically by using gradient boosted regression trees as a function of the applied cuts
- During each step the track with the highest transverse momentum is taken in order to evaluate the tagging power
- Variables used in the optimisation:
  - ghost\_prob
  - PID\textsubscript{e}
  - Tr\_PT
  - Tr\_P
  - min(deltaPhi)
  - sigma\_IP
  - sigma\_IPPU
  - Tr\_PROBNNk
  - Tr\_PROBNNpi
  - Tr\_PROBNNp
  - Tr\_PROBNNe
  - Tr\_PROBNNe - Tr\_PROBNNpi
  - Tr\_PROBNNmu
OS$_e$ strategy (II)

- Classifier trained to discriminate between:
  - **Signal**: rightly tagged B candidates
  - **Background**: wrongly tagged B candidates

- Input variables:
  - Tr$_P$
  - Tr$_{PT}$
  - sigma$_{IPPU/IPPU}$
  - ghost$_{prob}$
  - Tr$_{CHI2NDOF}$
  - nTracks
  - deltaEta
  - deltaR
  - B$_{PT}$
  - e/P
  - B$_{IP}$
  - sigma$_{IP}$
  - BPVIPCHI2
  - ProbNN ghost
  - deltaQ
SSK strategy (I)

- Loose preselection cuts applied to tagging tracks
- First MVA classifier to separate fragmentation tracks from underlying event tracks:
  - **Signal**: fragmentation track $\rightarrow$ track which ancestor is identical to the one of the signal $B_s^0$
  - **Background**: underlying track $\rightarrow$ otherwise (originating from soft QCD processes)

- Input variables:
  - $\log(n\text{Tracks})$
  - $n\text{PVs}$
  - $\log(B_{PT})$
  - $\log(Tr_P)$
  - $\log(Tr_{PT})$
  - $\log(\text{abs(deltaPhi)})$
  - $\log(\text{abs(deltaEta)})$
  - $\log(p\text{Trel})$
  - $\log(Tr_{BPVIP})$
  - $\log(\sqrt{Tr_{BPVIPCHI2}})$
  - $\log(Tr_{TRCHI2DOF})$
  - $\log(Tr_{PROBNNk})$
  - $\log(Tr_{PROBNNpi})$
  - $\log(Tr_{PROBNNp})$
SSK strategy (II)

- From first classifier output:
  - select 3 tracks with highest output value above a given cut
  - tested several cuts [0.5, 0.8]

- Second classifier to distinguish $B_s^0$ from $\overline{B_s^0}$
  - **Signal**: B_ID_prod > 0
  - **Background**: B_ID_prod < 0

- Input variables:
  - log(nTracks)
  - nPVs
  - log(B_PT)
  - 1st_Tr_charge $\cdot$ 1st_Tr_BDTG
  - 2nd_Tr_charge $\cdot$ 2nd_Tr_BDTG
  - 3rd_Tr_charge $\cdot$ 3rd_Tr_BDTG
RNN consists of nodes with an internal connection, in addition to a connection to the next node.

In practice, the recurrence is truncated and the network 'unrolled' into a finite (variable) length.

This allows for a variable length (number of track, vertices, etc.) inputs and outputs.

Weights common to each step.

Used in all domains for sequence classification/labelling: machine translation, speech recognition, image captioning, etc.
Inclusive tagger: current configuration

- **RNN configuration:**
  - preprocessing: 1 feed-forward layer
  - recurrent network: 2 GRU layers

- **Track features:**
  - PROBNNe
  - PROBNNghost
  - PROBNNk
  - PROBNNmu
  - PROBNNpi
  - PROBNNp
  - Tr_P
  - Tr_PT
  - Tr_charge
  - BPVIP
  - BPVIPCHI2
  - SumBDT_ult
  - zfirst - B_OWNPV_Z
  - Tr_P · B_P
  - Tr_PT - B_PT
  - cos(Tr_phi - B_phi)
  - B_eta - Tr_eta