LHCb 2008-002

LPHE 2008-002

Clone Track Identification using the Kullback-Liebler Distance

M. Needham

Laboratoire de Physique des Hautes Energies,
École Polytechnique Fédérale de Lausanne

January 22, 2008

Abstract

The implementation and performance of an algorithm that identifies and removes clone tracks using the Kullback-Liebler distance is discussed. For long tracks this algorithm reduces the clone rate to the level of 3 per mille for a 2 per mille loss in efficency.

1 Introduction

Often in combinatoric problems such as pattern recognition clones are created. In track reconstruction one way to remove clones is by comparison of hits. If a pair of tracks share hits they are considered to be clones and the track with the lower quality is removed [1]. Such a a procedure is used to identify clone tracks produced by the LHCb long track reconstruction algorithms the forward tracking and the track matching [2, 3]. However, this approach will fail to identify clones if they do not have hits in common. The current VELO track reconstruction is known to create clones of this type. In some cases the algorithm splits the clusters coming from one particle into two tracks, one consisting of the hits from the forward stations and the other the hits of the remaining stations. In this note an algorithm, based on the Kullback-Liebler distance, to identify clones of this type is described.

This note is organized as follows. First, the Kullback-Liebler distance is defined. Next the implementation of clone killing algorithm based on this is in the LHCb software framework discussed. Finally, the performance of this procedure is evaluated.

2 Definitions

In order to proceed it is helpful to define what we mean by a clone track. Two tracks are clones if they provide the same information. A track i is characterized by a state vector x_i and an associated covariance matrix C_i at a given z in the detector. The Shannon information content of the track is [4]:

$$H(X) = -log(p(x)) \tag{1}$$

where $p(x_i)$ is a multi-dimensional Gaussian probability density function that represents the track state. The expectation of this quantity:

$$H(X) = -\int p(x) \cdot log(p(x))dx \tag{2}$$

is the entropy associated with the knowledge of the track parameters. The relative entropy between the two tracks is then:

$$\begin{aligned} D_{KL}(p_1||p_2) &= H(p_1,p_2) - H(p_1) \\ &= -\int p_1(x)\cdot log(p_2(x))dx + \int p_1(x)\cdot log(p_1(x))dx \\ &= \int p_1(x)\cdot log\left(\frac{p_1(x)}{p_2(x)}\right)dx \end{aligned} \tag{3}$$

The quantity D_{KL} is also known as the Kullback-Liebler divergence [5] and measures the difference in information content between p_1 and p_2. Therefore, if this distance is small then two tracks are likely to be clones. Since this quantity is not symmetric under the interchange of p_1 and p_2 it is not a true distance metric. However, a distance measure can trivially be constructed:

$$D(p_1,p_2) = 2\cdot(D_{KL}(p_1||p_2) + D_{KL}(p_2||p_1)) \tag{4}$$

For multi-dimensional Gaussian pdf it can be shown that $D(p_1,p_2)$ is given by [6]:

$$D(p_1,p_2) = tr[(C_1 - C_2)(G_2 - G_1)] + (x_1 - x_2)^T(G_1 + G_2)(x_1 - x_2) \tag{5}$$

where $G_i = C_i^{-1}$.

3 Implementation

Two algorithms have been implemented in the TrackUtils package to allow the evaluation of the Kullback-Liebler distance. The first, **TrackBuildCloneTable**, builds from a container of tracks a linker table. This table contains all pairs of tracks with a Kullback-Liebler distance below some value that is defined via jobOptions. The comparison can be made at any position along the track. Again this choice is driven by jobOptions. In order to ensure the CPU performance of the algorithm is not prohibitive some optimizations are performed. For example, for each track, the inverse covariance matrix G_i that is needed in Eqn 5 is cached locally so that only one matrix inversion is performed per track.

The second algorithm, **TrackCloneCleaner**, flags clone tracks for removal. To choose between tracks identified as clones the numbers of hits is used. In the case that the number of hits is equal the track with the lower χ^2 is considered to be the one of higher quality. At the end of this procedure all

clone tracks are tagged and can trivially be removed from any subsequent analysis by accessing the **CloneDist** flag in the **ExtraInfo** map of the **Track** class. Since the Kullback-Liebler distance is stored in the map it is possible to make stronger cuts on this quantity at a later stage.

4 Results

The performance of the algorithm was tested using a sample of 12000 $B_d \to J/\psi(\mu^+\mu^-)K_S(\pi^+\pi^-)$ events generated at the default LHCb luminosity of 2×10^{32} cm^{-2}s^{-1} and reconstructed with Brunel v31r11. The studies were made for long tracks using the definitions of acceptance and efficiency and ghost rate given in [7]. The clone rate is defined as:

$$\text{clone rate} = \frac{\text{N(reconstructed } \cap \text{ accepted)}}{\text{N(accepted)}} - 1 \tag{6}$$

Fig. 1 shows the distribution of $log(D(p_1, p_2))$ for all possible two track combinations together with the distribution for pairs that are identified as clones using Monte Carlo truth information. A clear separation between clone pairs and random combinations can be seen on this plot. By cutting on this distribution at $log(5000) = 8.52$ the clone rate can be reduced from 2.2 % to 3 per mille with a loss in efficiency of 2 per mille. Applying this cut the ghost rate is reduced from 14.6 % to 13.8 %. This is not suprising. First, it is expected that the clone rate for ghosts is around 2%, as is the case for real tracks, though in this case they can not be identified using the Monte Carlo truth. In addition, as discussed in [8], around 10 % of the ghost rate is due to processes such a photon conversion or hadronic interactions in the detector which lead to the creation of particles that are close in space and hence more likely to cause clones.

As a second test the standard clone killing algorithm was removed and the output of the matching and the forward tracking merged into one container. This procedure leads to a clone rate of 86.6 % for long tracks. Fig. 2 shows the distribution of $log(D(p_1, p_2))$ in this case. Again clone pairs are easily distinguished from random combinations of tracks. If a cut is applied on this distribution at 8.52 the clone decreases to 5 per mille with a loss in efficiency of 2 per mille. After applying this cut the ghost rate is again 13.8 %. The performance numbers for the four runs described in this note are summarized in Table 1.

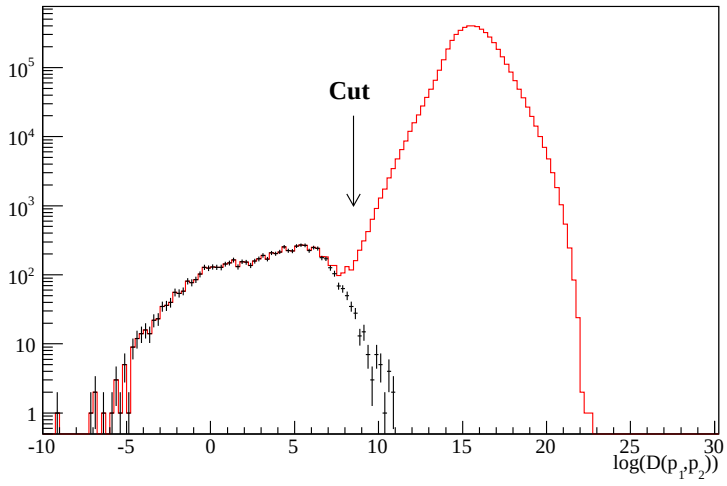

Figure 1: $log(D(p_1, p_2))$ for all pairs of long tracks (solid line) and those pairs identified as clones using Monte Carlo truth (points).

The CPU performance of the algorithm has also been evaluated for the second case. Running on a 64 bit 2.8 GHz Opteron processor the algorithm takes 7.1 ms per event. With further work it should be possible to reduce this time.

Run	Efficiency (%)	Ghost rate (%)	Clone rate (%)
Standard	91.4	14.5	2.2
Standard + KL clone killer	91.2	13.8	0.3
No clone killer	91.4	10.3	86.6
KL clone killer	91.2	13.8	0.5

Table 1: Summary of the tests made. *Nota Bene*, the ghost rate in the run with no clone killer is artificially reduced by the high clone rate.

5 Summary

In this note the performance of an algorithm that uses the Kullback-Liebler distance to identify and remove clones has been presented. It has been shown that that this approach easily removes clones that do not share hits for the

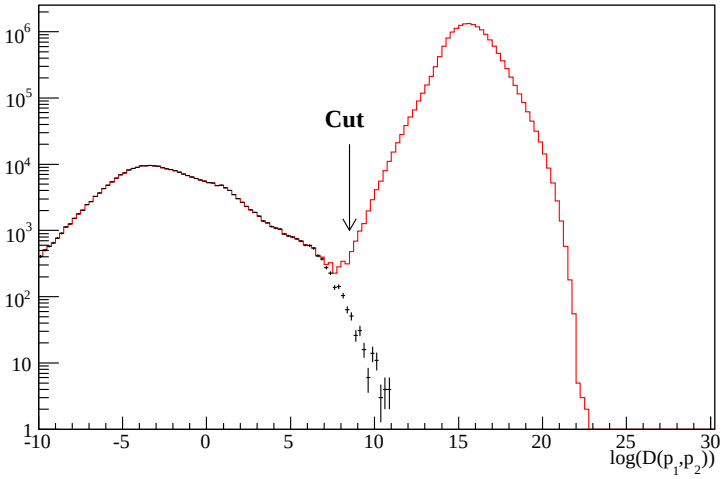

Figure 2: $log(D(p_1, p_2))$ for all pairs of tracks (solid line) and those identified as clones using Monte Carlo truth (points). In this case the standard clone killing algorithm [1] was not run.

case of the long tracking. The approach is generic and can easily be extended to other track types. Such studies will be described in a future note. As well as removing clones this approach also reduces the ghost rate from 14.5 % to 13.8 %

References

[1] E. Rodrigues. Dealing with clones in the Tracking. LHCb-note 2006-065.

[2] O. Callot and S. Menzemer. Performance of the forward tracking. LHCb-note 2007-015.

[3] M. Needham. Performance of the Track Matching. LHCb-note 2007-129.

[4] D.MacKay. *Information Theory. Inference and Learning Algorithms.* Cambridge University Press, 2003.

[5] S. Kullback and R. Liebler. On information and sufficiency. *Annals of Mathematical Statistics*, (79-86), 1951.

[6] R. Fruehwirth. Track Fitting with non-Gaussian noise. *Computer Physics Communications*, (100):1–16, 1997.

[7] M. Needham. Performance of the LHCb Track Reconstruction Software. LHCb-note 2007-144.

[8] M. Needham. Classification of Ghost Tracks. LHCb-Note 2007-128.