

Table of Contents

The Adaptive Gaussian-Sum filter.....	1
Usage.....	1
Parameters.....	1
Limitation of the number of components.....	2
Vertex validity and exceptions.....	2
Review Status.....	2

The Adaptive Gaussian-Sum filter

Complete: 

The Adaptive Gaussian-Sum filter combines the AdaptiveVertexFitter (AVF) the GaussianSumFitter (GSF). Indeed, as in the AVF the computation of the vertex position is independent of the computation of the track weights, the Kalman filter used in the default implementation can be replaced by a GSF. In this way, the complete mixture modeling the track measurements is taken into account, instead of only a single component. This filter is referred to as the *Adaptive-GSF* (A-GSF).

Usage

An easy way to use the A-GSF is through the AdaptiveGsfVertexFitter. A PSet has to be given with the parameters to steer the fitter, and the default linearization point finder can optionally be replaced. As it is a VertexFitter, it is then to be used as such. Example:

```
AdaptiveGsfVertexFitter fitter(pSet);  
TransientVertex myVertex = fitter.vertex(vectorOfRecTracks);=
```

Parameters

The PSet needed for the configuration has the following structure

```
GSFParameters = cms.PSet(  
    maxshift = cms.double(0.0001),  
    maxstep = cms.int32(30),  
    maxlpshift = cms.double(0.1),  
    weightthreshold = cms.double(0.001),  
    limitComponents = cms.bool(True),  
    smoothTracks = cms.bool(True),  
    GsfMergerParameters = cms.PSet(  
        maxNbrComponents= cms.int32(4),  
        merger = cms.string("CloseComponentsMerger"),  
        distance = cms.string("KullbackLeiblerDistance")  
    )  
)
```

The parameters are the following:

Parameter Name	Description	Default
maxshift	Convergence criterion (maximum transverse distance between vertex computed in the previous and the current iterations)	0.0001
maxlpshift	Criterion for the relinearization of the tracks	0.1
maxNbrOfIterations	Maximum number of iterations to perform	30
weightthreshold	Minimum track weight for a track to be considered "significant". If fewer than two tracks are significant, an invalid vertex is returned.	0.001
smoothTracks	Specifies whether the tracks have to be smoothed at the end of the fit	false
limitComponents	Specifies whether the number of components has to be limited to a maximum number during the fit.	true
maxNbrComponents	Maximum number of components to keep	4
merger	Strategy used to limit the number of components	CloseComponentsMerger

distance	Distance measurement between components	KullbackLeiblerDistance
----------	---	-------------------------

Limitation of the number of components

It has to be kept in mind that without limitation of the number of components during the fit, these will increase exponentially, and the computing time will increase likewise! The merger indicated is the most useful. The other implemented merger is the *LargestWeightsStateMerger*. Little difference in performance is seen between the two distance measurements implemented, the *KullbackLeiblerDistance* and the *MahalanobisDistance*.

Vertex validity and exceptions

The vertex returned may not be valid in some cases. The user had to check the validity of the vertex with the method `isValid()`. In each case, an error message is put into the log:

- The maximum number of iterations is exceeded
- The fitted position is out of the tracker bounds
- Too many tracks have been downweighted, and fewer than two significant tracks remain.

Review Status

Editor/Reviewer and date	Comments
Main.speer - 03 Apr 2007	page content last edited
JennyWilliams - 10 Apr 2007	edited for swguide inclusion
ThomasSpeer - 27 Feb 2009	Review and update

Responsible: Main.speer (Thomas Speer)

This topic: CMSPublic > SWGuideAdaptiveGsfVertexFitter

Topic revision: r3 - 2009-02-27 - ThomasSpeer



Copyright & © 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback