NAME

SpotRecoveryLatentModel< copulaPolicy > − Random spot recovery rate latent variable portfolio model.

SYNOPSIS

```
#include <ql/experimental/credit/spotlosslatentmodel.hpp>
```

Inherits **LatentModel< copulaPolicy >**.

Public Member Functions

SpotRecoveryLatentModel (const std::vector< std::vector< **Real** > > &**factorWeights**, const std::vector< **Real** > &recoveries, **Real** modelA, LatentModelIntegrationType::LatentModelIntegrationType integralType, const initTraits &ini=initTraits())

void **resetBasket** (const boost::shared_ptr< **Basket** > basket) const

Probability conditionalDefaultProbability (const **Date** &date, **Size** iName, const std::vector< **Real** > &mktFactors) const

Probability conditionalDefaultProbability (**Probability** prob, **Size** iName, const std::vector< **Real** > &mktFactors) const

Probability conditionalDefaultProbabilityInvP (**Real** invCumYProb, **Size** iName, const std::vector< **Real** > &m) const

Real expCondRecovery (const **Date** &d, **Size** iName, const std::vector< **Real** > &mktFactors) const

Real expCondRecoveryP (**Real** uncondDefP, **Size** iName, const std::vector< **Real** > &mktFactors) const

Real expCondRecoveryInvPinvRR (**Real** invUncondDefP, **Real** invUncondRR, **Size** iName, const std::vector< **Real** > &mktFactors) const

Real conditionalRecovery (**Real** latentVarSample, **Size** iName, const **Date** &d) const

Real latentRRVarValue (const std::vector< **Real** > &allFactors, **Size** iName) const

Real conditionalExpLossRR (const **Date** &d, **Size** iName, const std::vector< **Real** > &mktFactors) const

Real conditionalExpLossRRInv (**Real** invP, **Real** invRR, **Size** iName, const std::vector< **Real** > &mktFactors) const

Real expectedLoss (const **Date** &d, **Size** iName) const

Protected Member Functions

const boost::shared_ptr< LMIntegration > & **integration** () const
access to integration:

Additional Inherited Members

Detailed Description

template<class copulaPolicy>

class QuantLib::SpotRecoveryLatentModel< copulaPolicy >" Random spot recovery rate latent variable portfolio model.

See:

A Spot Stochastic Recovery Extension of the Gaussian Copula N.Bennani and J.Maetz, MPRA July 2009

Extension of Spot Recovery model for Gaussian Copula H.Li, October 2009, MPRA

The model is adpated here for a multifactor set up and a generic copula so it can be used for pricing in single factor mode or for risk metrics in its multifactor version.

Member Function Documentation

Real expCondRecovery (const Date & d, Size iName, const std::vector< Real > & mktFactors) const

Expected conditional spot recovery rate. Conditional on a set of systemic factors and default returns the integrated attainable recovery values.

Corresponds to a multifactor generalization of the model in eq. 44 on p.15 of **Extension of Spot Recovery Model for Gaussian Copula** Hui Li. 2009 Only remember that $ ho_i^2 $ there is here (multiple betas): $ _k re is here: $ _k

(d,l corresponds to first and last set of betas)

Real conditionalRecovery (Real latentVarSample, Size iName, const Date & d) const

Implements equation 42 on p.14 (second). Remember that for this call to make sense the sample used must be one leading to a default. Theres no check on this. This member typically to be used within a simulation.

Real latentRRVarValue (const std::vector< Real > & allFactors, Size iName) const

Due to the way the latent model is splitted in two parts, we call the base class for the default sample and the LM owned here for the RR model sample. This sample only makes sense if it led to a default.

Parameters:

allFactors All sampled factors, default and RR valiables.
iName The index of the name for which we want the RR sample

Real expectedLoss (const Date & d, Size iName) const

Single name expected loss.

The main reason of this method is for the testing of this model. The model is coherent in that it preserves the single name expected loss and thus is coherent with the single name CDS market when used in the pricing context. i.e. it should match: $pdef_i(d) imes RR_i $

Author

Generated automatically by Doxygen for QuantLib from the source code.