Parallel implementation of the HDP-mixture model with applications to topic modeling

Generated by Doxygen 1.8.6

Tue Nov 29 2016 11:32:30

Contents

Chapter 1

Implementation of a parallelizable Hierarchical Dirichlet Process Mixture Model

1.1 Introduction

This code implement the aglorithm propose in the paper of J. Chang, J. W. Fisher III Parallel Sampling of HDPs using sub-clusters splits, NIPS, 2014.

The paper deal the topic modeling problem, but this algorithm could be extended to a differente problem where the goal is to discover the shared cluster between group of data. In this aglorithm there is an intrchange between step of Gibbs sampler with step of Metropolis-Hastings. The Gibbs sampler's step dela only no empty cluster and there is the update the interest quantity (β, π, θ) using their full conditional. During these steps the number of clusters don't change. With Metropolis Hastings' steps there is the propose of marge of two clusters or split of a cluster. For each topic k there is a partition in two sub-topics, kl e kr , which respectively are the left sub-topic and the right sub-topic. The new topics are proposed on the basis of these sub-topics.

This code manage only conjugate model.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 BETA Struct Reference

Clusters' global weights.

`#include <Struct.hpp>`

Public Attributes

- double **a**
- vector< double > **b_c**
- vector< double > **Left**
- vector< double > **Right**
- double **k**

5.1.1 Detailed Description

Clusters' global weights.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

The documentation for this struct was generated from the following file:

- Struct.hpp

5.2 C Struct Reference

Statistics.

`#include <Struct.hpp>`

Public Attributes

- STAT **a**
- STAT **b**
- STAT **c**
- STAT **a_left**
- STAT **b_left**
- STAT **c_left**
- STAT **a_right**
- STAT **b_right**
- STAT **c_right**

5.2.1 Detailed Description

Statistics.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

The documentation for this struct was generated from the following file:

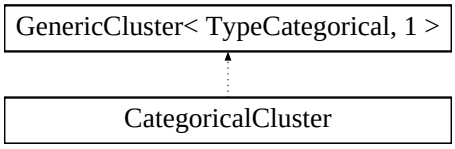
- Struct.hpp

5.3 CategoricalCluster Class Reference

Management of cluster's and subclusters' informations with Categorical Likelihood.

`#include <Cluster.hpp>`

Inheritance diagram for CategoricalCluster:



Public Types

- using THETA = TypeCategorical< 1 >::THETA

  Latent Paramenter, vector of distinct element's weigths which are contained into the cluster.
- using Point = TypeCategorical< 1 >::Point

  Type of single data, repeaterd data.
- using STAT = TypeCategorical< 1 >::STAT

  Statistics' vector. In this container there are the update of latent parameters' hyperparameters of clusters or sub-clusters. Number of data which are contained into cluster and sub-clusters.

Public Member Functions

- CategoricalCluster ()

  Default costructor.
- ~CategoricalCluster ()=default

  Default distructor.
- CategoricalCluster (double _Beta, double _BetaLeft, double _BetaRight, THETA &_Theta, THETA &_ThetaLeft, THETA &_ThetaRight, STAT &_c, STAT &_cLeft, STAT &_cRight, unsigned int _NrTable, unsigned int _NrTableLeft, unsigned int _NrTableRight)

  Costructor which required in input all informations about cluster and sub-clusters.
- void SetTheta (THETA &_Theta)

  Fix the cluster's latent paramenter.
- void SetThetaLeft (THETA &_ThetaLeft)

  Fix the left sub-cluster's latent paramenter.
- void SetThetaRight (THETA &_ThetaRight)

  Fix the right sub-cluster's latent paramenter.
- void SetBeta (double _Beta)

  Fix cluster's global weigth.
- void SetBetaLeft (double _BetaLeft)

  Fix left sub-cluster's global weigth.
- void SetBetaRight (double _BetaRight)

  Fix rigth sub-cluster's global weigth.
- void SetGlobalTable (unsigned int _NrTable)

  Fix the number of global table which characterizes the cluster.
- void SetGlobalTableLeft (unsigned int _NrTableLeft)

  Fix the number of global table which characterizes the left sub-cluster.
- void SetGlobalTableRight (unsigned int _NrTableRight)

  Fix the number of global table which characterizes the right sub-cluster.
- void SetStatistics (STAT &_c)

  Set the statistics. Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del cluster.
- void SetStatisticsLeft (STAT &_cLeft)

  Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del sub-cluster sinistro.
- void SetStatisticsRight (STAT &_cRight)

  Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del sub-cluster destro.
- void ViewTheta (THETA &_Theta) const

  Estrae il parametro latente del cluster.
- void ViewThetaLeft (THETA &_ThetaLeft) const

  Estrae il parametro latente del sub-cluster sinistro.
- void ViewThetaRight (THETA &_ThetaRight) const

  Estrae il parametro latente del sub-cluster destro.
- double ViewThetaId (unsigned int _id) const

  Estrare l'i-esimo elemento del parametro latente del cluster.
- double ViewThetaLeftId (unsigned int _id) const

  Estrare l'i-esimo elemento del parametro latente del sub-cluster sinistro.
- double ViewThetaRightId (unsigned int _id) const

  Estrare l'i-esimo elemento del parametro latente del sub-cluster destro.
- double ViewBeta () const

  Estra il peso globale del cluster.
- double ViewBetaLeft () const

  Estra il peso globale del sub-cluster sinistro.
- double ViewBetaRight () const

Estra il peso globale del sub-cluster destro.

- unsigned int ViewGlobalTable () const

  Estra il numero globale dei tavoli nel cluster.
- unsigned int ViewGlobalTableLeft () const

  Estra il numero globale dei tavoli nel sub-cluster sinistro.
- unsigned int ViewGlobalTableRight () const

  Estra il numero globale dei tavoli nel sub-cluster destro.
- void ViewStatistics (STAT &_c) const

  It sets the statistics, in other words the hyperparameter of cluster's latent paramenter.
- void ViewStatisticsLeft (STAT &_cLeft) const

  It sets the statistics, in other words the hyperparameter of left sub-cluster's latent paramenter.
- void ViewStatisticsRight (STAT &_cRight) const

  It sets the statistics, in other words the hyperparameter of right sub-cluster's latent paramenter.
- void ResetStatistics (unsigned int W)

  It resests the cluster's statistics.
- void ResetStatisticsLeft (unsigned int W)

  It resests the left sub-cluster's statistics.
- void ResetStatisticsRight (unsigned int W)

  It resests the right sub-cluster's statistics.
- void UpdateStatistics (STAT &counts4cleft, STAT &counts4right)

  Update of statistics. This method update the hyperparameter of cluster's and sub-clusters' latent paramenter.
- bool IsEmpty () const

  Check if cluster is empty.

Private Attributes

- double Beta

  Cluster's global weigth.
- double BetaLeft

  Left sub-cluster's global weigth.
- double BetaRight

  Right sub-cluster's global weigth.
- THETA Theta

  Cluster's latent parameter: weigth of distnict element which are conteined into the cluster.
- THETA ThetaLeft

  Left sub-cluster's latent parameter: weigth of distnict element which are conteined into the left sub-cluster.
- THETA ThetaRight

  Right sub-cluster's latent parameter: weigth of distnict element which are conteined into the right sub-cluster.
- STAT c

  Statisitcs to update clustr's latent paramenter: counts of element which are conteined into the cluster.
- STAT cLeft

  Statisitcs to update left sub-clustr's latent paramenter: counts of element which are conteined into the left sub-cluster.
- STAT cRight

  Statisitcs to update right sub-clustr's latent paramenter: counts of element which are conteined into the right sub-cluster.
- unsigned int NrTable

  Number of global table assings to the cluster.
- unsigned int NrTableLeft

  Number of global table assings to the left sub-cluster.
- unsigned int NrTableRight

  Number of global table assings to the right sub-cluster.

Additional Inherited Members

5.3.1 Detailed Description

Management of cluster's and subclusters' informations with Categorical Likelihood.

This class memorizes and extracts information of cluster and sub-clusters about global weight, latent parameters, hyperparameters' update. In this case that the likelihood is categorical, data could be repeted, the latent parameters are the weight of distinct elements and the statistics are conts of elements in the clusters. The latent paramenters are the mixutre's parameters.

Authors

{Debora Parisi and Stefania Perego}

Date

Febbrario 2016

5.3.2 Constructor & Destructor Documentation

5.3.2.1 CategoricalCluster::CategoricalCluster (double *_Beta,* double *_BetaLeft,* double *_BetaRight,* THETA & *_Theta,* THETA & *_ThetaLeft,* THETA & *_ThetaRight,* STAT & *_c,* STAT & *_cLeft,* STAT & *_cRight,* unsigned int *_NrTable,* unsigned int *_NrTableLeft,* unsigned int *_NrTableRight*)

Costructor which required in input all informations about cluster and sub-clusters.

Parameters

_Beta	- Cluster's global weigth
_BetaLeft	- Left sub-luster's global weigth
_BetaRight	- Right sub-luster's global weigth
_Theta	- Cluster's latent paramenter: weigth of distnict element which are conteined into the cluster
_ThetaLeft	- Left sub-cluster's latent parameter: weigth of distnict element which are conteined into the left sub-cluster
_ThetaRight	- Right sub-cluster's latent parameter: weigth of distnict element which are conteined into the right sub-cluster
_c	- Statisitcs to update clustr's latent paramenter: counts of element which are conteined into the cluster
_cLeft	-Statisitcs to update left sub-clustr's latent paramenter: counts of element which are conteined into the left sub-cluster
_cRight	- Statisitcs to update right sub-clustr's latent paramenter: counts of element which are conteined into the right sub-cluster
-NrTable	- Number of global table assings to the cluster
-NrTableLeft	- Number of global table assings to the left sub-cluster
-NrTableRight	- Number of global table assings to the right sub-cluster

5.3.3 Member Function Documentation

5.3.3.1 bool CategoricalCluster::IsEmpty ( ) const [virtual]

Check if cluster is empty.

Returns

TRUE if cluster is empty otherwise FALSE

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.2 void CategoricalCluster::ResetStatistics (unsigned int *W*) [virtual]

It resests the cluster's statistics.

Parameters

W	- dimenstion of statistics with to update hyperparamenters of cluster's latent paramenter

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.3 void CategoricalCluster::ResetStatisticsLeft (unsigned int *W*) [virtual]

It resests the left sub-cluster's statistics.

Parameters

W	- dimenstion of statistics with to update hyperparamenters of left sub-cluster's latent paramenter

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.4 void CategoricalCluster::ResetStatisticsRight (unsigned int *W*) [virtual]

It resests the right sub-cluster's statistics.

Parameters

W	- dimenstion of statistics with to update hyperparamenters of right sub-cluster's latent paramenter

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.5 void CategoricalCluster::SetBeta (double *_Beta*) [virtual]

Fix cluster's global weigth.

Parameters

_Beta	- Weigth in input

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.6 void CategoricalCluster::SetBetaLeft (double *_BetaLeft*) [virtual]

Fix left sub-cluster's global weigth.

Parameters

_Beta	- Weigth in input

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.7 void CategoricalCluster::SetBetaRight (double *_BetaRight*) [virtual]

Fix rigth sub-cluster's global weigth.

Parameters

_Beta	- Weigth in input

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.8 void CategoricalCluster::SetGlobalTable (unsigned int *_NrTable*) [virtual]

Fix the number of global table which characterizes the cluster.

Parameters

NrTable	- number of global table which characterizes the cluster

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.9 void CategoricalCluster::SetGlobalTableLeft (unsigned int *_NrTableLeft*) [virtual]

Fix the number of global table which characterizes the left sub-cluster.

Parameters

NrTable	- number of global table which characterizes the left sub-cluster

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.10 void CategoricalCluster::SetGlobalTableRight (unsigned int *_NrTableRight*) [virtual]

Fix the number of global table which characterizes the right sub-cluster.

Parameters

NrTable	- number of global table which characterizes the right sub-cluster

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.11 void CategoricalCluster::SetStatistics (STAT & *_c*)

Set the statistics. Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del cluster.

Parameters

_c	- statistiche del cluster, conteggi degli elementi finiti nel cluster

5.3.3.12 void CategoricalCluster::SetStatisticsLeft (STAT & *_cLeft*)

Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del sub-cluster sinistro.

Parameters

_cLeft	- statistiche del sub-cluster sinistro, conteggi degli elementi finiti nel sub-cluster sinistro

5.3.3.13 void CategoricalCluster::SetStatisticsRight (STAT & *_cRight*)

Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del sub-cluster destro.

Parameters

_cLeft	- statistiche del sub-cluster destro, conteggi degli elementi finiti nel sub-cluster destro

5.3.3.14 void CategoricalCluster::SetTheta (THETA & *_Theta*)

Fix the cluster's latent paramenter.

Parameters

_Theta	- cluster's latent paramenter. Its type is THETA

5.3.3.15 void CategoricalCluster::SetThetaLeft (THETA & *_ThetaLeft*)

Fix the left sub-cluster's latent paramenter.

Parameters

_Theta	- left sub-cluster's latent paramenter. Its type is THETA

5.3.3.16 void CategoricalCluster::SetThetaRight (THETA & *_ThetaRight*)

Fix the right sub-cluster's latent paramenter.

Parameters

_Theta	- Right sub-cluster's latent paramenter. Its type is THETA

5.3.3.17 void CategoricalCluster::UpdateStatistics (STAT & *counts4cleft,* STAT & *counts4right*)

Update of statistics. This method update the hyperparameter of cluster's and sub-clusters' latent paramenter.

Parameters

counts4cleft	- statistics to update the hyperparameters of left sub-subcluster's latent parameters
counts4cright	- statistics to update the hyperparameters of right sub-subcluster's latent parameters

5.3.3.18 double CategoricalCluster::ViewBeta ( ) const [virtual]

Estra il peso globale del cluster.

Returns

Peso del cluster

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.19 double CategoricalCluster::ViewBetaLeft ( ) const [virtual]

Estra il peso globale del sub-cluster sinistro.

Returns

Peso del sub-cluster sinistro

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.20 double CategoricalCluster::ViewBetaRight ( ) const [virtual]

Estra il peso globale del sub-cluster destro.

Returns

Peso del sub-cluster destro

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.21 unsigned int CategoricalCluster::ViewGlobalTable ( ) const [virtual]

Estra il numero globale dei tavoli nel cluster.

Returns

Numero di tavoli nel cluster

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.22 unsigned int CategoricalCluster::ViewGlobalTableLeft ( ) const [virtual]

Estra il numero globale dei tavoli nel sub-cluster sinistro.

Returns

Numero di tavoli nel sub-cluster sinistro

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.23 unsigned int CategoricalCluster::ViewGlobalTableRight ( ) const [virtual]

Estra il numero globale dei tavoli nel sub-cluster destro.

Returns

Numero di tavoli nel sub-cluster destrp

Implements GenericCluster< TypeCategorical, 1 >.

5.3.3.24 void CategoricalCluster::ViewStatistics (STAT & *_c*) const

It sets the statistics, in other words the hyperparameter of cluster's latent paramenter.

Parameters

_c	- Object's type is STA.

5.3.3.25 void CategoricalCluster::ViewStatisticsLeft (STAT & *_cLeft*) const

It sets the statistics, in other words the hyperparameter of left sub-cluster's latent paramenter.

Parameters

_c	- Object's type is STA.

5.3.3.26 void CategoricalCluster::ViewStatisticsRight (STAT & *_cRight*) const

It sets the statistics, in other words the hyperparameter of right sub-cluster's latent paramenter.

Parameters

_c	- Object's type is STA.

5.3.3.27 void CategoricalCluster::ViewTheta (THETA & *_Theta*) const

Estrae il parametro latente del cluster.

Parameters

_Theta	- oggetto di tipo THETA in cui viene memorizzato il parametro latente del cluster, peso degli elementi distinti nel cluster

5.3.3.28 double CategoricalCluster::ViewThetaId (unsigned int *_id*) const

Estrare l'i-esimo elemento del parametro latente del cluster.

Parameters

Elemento	da estrarre

Returns

peso dell'elemento nella posizione indicata del cluster

5.3.3.29 void CategoricalCluster::ViewThetaLeft (THETA & *_ThetaLeft*) const

Estrae il parametro latente del sub-cluster sinistro.

Parameters

_ThetaLeft	- oggetto di tipo THETA in cui viene memorizzato il parametro latente del cluster, peso degli elementi distinti nel sub-cluster sinistro

5.3.3.30 double CategoricalCluster::ViewThetaLeftId (unsigned int *_id*) const

Estrare l'i-esimo elemento del parametro latente del sub-cluster sinistro.

Parameters

Elemento	da estrarre

Returns

peso dell'elemento nella posizione indicata del sub-cluster sinistro

5.3.3.31 void CategoricalCluster::ViewThetaRight (THETA & *_ThetaRight*) const

Estrae il parametro latente del sub-cluster destro.

Parameters

_ThetaRight	- oggetto di tipo THETA in cui viene memorizzato il parametro latente del cluster, peso degli elementi distinti nel sub-cluster destro

5.3.3.32 double CategoricalCluster::ViewThetaRightId (unsigned int *_id*) const

Estrare l'i-esimo elemento del parametro latente del sub-cluster destro.

Parameters

Elemento	da estrarre

Returns

peso dell'elemento nella posizione indicata del subl-cluster destro

The documentation for this class was generated from the following file:

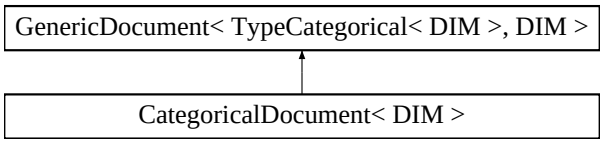
- Cluster.hpp

5.4 CategoricalDocument< DIM > Class Template Reference

Derived class for topic modeling, where data are categorical and the base measure is the Dirichlet distribution. This class represents a document in the topic modeling problem. It manages the words and is in charge of sampling the topic labels. It samples the model's parameters specific to the document: $\alpha, \pi_j, (\bar{\pi})_{jl}, (\bar{\pi})_{jr}, m_j, \bar{m}_{jl}, \bar{m}_{jl}$. It keep track of words' counts in topics.

`#include <Document.hpp>`

Inheritance diagram for CategoricalDocument< DIM >:



Public Types

- using STAT = TypeCategorical< 1 >::STAT

  Statistics for updating hyperparameters of latent parameter's distribution.
- using THETA = TypeCategorical< 1 >::THETA

  Latent parameter: vector of distinct words' weights in the topic.
- using POINT = TypeCategorical< 1 >::Point

  A datum.
- using ClusterID = unsigned int

  Topic id.

Public Member Functions

- CategoricalDocument (double _alpha)

  Constructor.
- CategoricalDocument ()=default

Default constructor.

- ~CategoricalDocument ()

  Destructor.

- CategoricalDocument (CategoricalDocument &&doc)

  Move constructor.

- CategoricalDocument (const CategoricalDocument &doc)=default

  Copy constructor.

- CategoricalDocument & operator= (const CategoricalDocument &doc)

  Copy assignment operator.

- CategoricalDocument & operator= (CategoricalDocument &&doc)

  Move assignment operator.

- void UpdatePi (const vector< double > &_AllBeta, omprng &Gen)

  Update document specific topics' weights.

- void UpdatePiSub (const double _BetaLeft, const double _BetaRight, const ClusterID k, omprng &Gen)

  Updates document specific weights for subtopics of topic k.

- void UpdateAllPiSub (const vector< double > _BetaLeft, const vector< double > _BetaRight, omprng &Gen)

  Updates document specific subtopics' weights for all topics.

- void UpdateLocalTable (const vector< long double > &_stirling, const vector< double > &_Beta, omprng &Gen)

  Updates table counts.

- void UpdateLocalTableSub_OneCluster (const vector< long double > &_stirling, const double _BetaLeft, const double _BetaRight, const ClusterID k, omprng &Gen)

  Updates table counts for subtopics of topic k.

- void UpdateAllLocalTableSub (const vector< long double > &_stirling, const vector< double > &_BetaLeft, const vector< double > &_BetaRight, omprng &Gen)

  Updates table counts for subtopic of all topics.

- void UpdateZeta (const THETA &_ThetaId, const POINT _VettId, omprng &Gen)

  Updates topic label of a datum, sampled with the Sampling method.

- void UpdateZeta_and_Sub (const THETA &_ThetaId, const THETA &_ThetaIdLeft, const THETA &_ThetaIdRight, const unsigned int _VettId, omprng &Gen)

  Updates topic label of a datum, sampled with with the Sampling method.

- void UpdateZetaSub (const THETA &_ThetaIdLeft, const THETA &_ThetaIdRight, const POINT id, const unsigned int nidjk, const ClusterID k, omprng &Gen)

  Assigns the datum to the subtopics of topic k, after sampling the subtopics' label with the Sampling method.

- void UpdateZeta (const ClusterID _k)

  Method needed to remove topic k during M-H moves.

- void UpdateZeta (const ClusterID _k1, const ClusterID _k2)

  Method needed to remove two topics during M-H moves.

- unsigned int ViewNj () const

  Retrieve number of data in document j.

- void ViewData (vector< POINT > &_VettId) const

  Retrieve data id in document j.

- void ViewCounts4c (ClusterID _k, STAT &_counts4cleft, STAT &_counts4cright)

  Retrieve counts necessary for updating subtopics' latent parameters.

- unsigned int ViewNumTableID (const ClusterID _k) const

  Retrieve number of tables for topic k m_{jk}.

- unsigned int ViewNumTableLeftID (const ClusterID _k) const

  Retrieve number of tables for left subtopic of topic k m_{jkl}.

- unsigned int ViewNumTableRightID (const ClusterID _k) const

  Retrieve number of tables for right subtopic of topic k m_{jkr}.

- unsigned int ViewDataCountID (const ClusterID _k) const

Retrieve number of words in topic k $n_{\cdot k}$.

- unsigned int ViewDataCountLeftID (const ClusterID _k) const

  Retrieve number of topic in left subtopic of topic k $n_{\cdot kl}$.
- unsigned int ViewDataCountRightID (const ClusterID _k) const

  Retrieve number of topic in right subtopic of topic k $n_{\cdot kr}$.
- void ResetDataCountSub (const ClusterID k)

  Reset wourds counts in topic k.
- void ViewDataCount (vector< unsigned int > &_WordCount) const

  Retrieve a vector containing the number of words in all topics.
- void ViewDataCountLeft (vector< unsigned int > &_WordCountLeft) const

  Retrieve a vector containing the number of words in all left subtopics.
- void ViewDataCountRight (vector< unsigned int > &_WordCountRight) const

  Retrieve a vector containing the number of words in all right subtopics.
- void ViewIdCounts (vector< pair< POINT, unsigned int >> &_nidjk, const ClusterID _k)

  Retrieve counts and id of words in topic k.
- void ViewCluster (const ClusterID _k, pair< unordered_map< POINT, unsigned int >, unordered_map< POINT, unsigned int >> &_Cluster)

  Retrieve topic k.
- double ViewPiID (const ClusterID _k) const

  Retrieve document specific weight of topic k π_{jk}.
- double ViewPiLeftID (const ClusterID _k) const

  Retrieve document specific weight for the left subtopic of topic k π_{jkl}.
- double ViewPiRightID (const ClusterID _k) const

  Retrieve document specific weight for the right subtopic of topic k π_{jkr}.
- void ViewPi (vector< double > &_pi) const

  Retrieve a vector containing the document specific topic weights.
- void ViewPiLeft (vector< double > &_pi) const

  Retrieve a vector containing the document specific weights for left subtopics.
- void ViewPiRight (vector< double > &_pi) const

  Retrieve a vector containing the document specific weights for right subtopics.
- void SetAlpha (const double _alpha)

  Set the concentration parameter α of the Dirichlet process governing the document.
- void SetNj (const unsigned int _Nj)

  Set the number of words in document j.
- void SetPi (vector< double > &_Pi)

  Set the vector containing the document specific topics weights.
- void InsertNewCluster (const pair< unordered_map< POINT, unsigned int >, unordered_map< POINT, unsigned int > > &NewCluster, const double _Pi, const double _PiLeft, const double _PiRight, const unsigned int _WordCount, const unsigned int _WordCountLeft, const unsigned int _WordCountRight, const unsigned int _LocalTable, const unsigned int _LocalTableLeft, const unsigned int _LocalTableRight)

  Insert a new topic.
- void RemoveCluster (const vector< ClusterID > &_k)

  Remove topics with id contained in the input vector.
- void RemoveCluster (const ClusterID _k)

  Remove topic k.
- void RemoveCluster (ClusterID _k1, ClusterID _k2)

  Remove two topics.
- unsigned int CheckLeftSubcluster (const ClusterID _k)

  Verifies if topic k has an empty left subtopic.
- unsigned int CheckRightSubcluster (const ClusterID _k)

  Verifies if topic k has an empty right subtopic.

- void ViewLabel (vector< pair< POINT, ClusterID >> &Data)

  Retrieve labels assigned to words.
- void SetDataset (std::istringstream &SSTR)

  Acquires data.
- unsigned int SortData (unsigned int _K, omprng &Gen)

  Allocates words in the Zeta container.

Private Member Functions

- void UpdateDataCount ()

  Updates word counts in topics.
- void Sampling (std::vector< unsigned int > &_temp_counts, std::vector< double > &_Weights, unsigned int _nidj, omprng &Gen)

  Samples topic's or subtopic's label from categorical distribution.

Private Attributes

- unordered_map< ClusterID, pair < unordered_map< POINT, unsigned int >, unordered_map < POINT, unsigned int > > > Zeta

  Data container. Each topic has a left and right subtopic: in the subtopic map, the key is the datum and the mapped value is the number of times that datum appear in the document, in the subtopic of that topic.
- unordered_map< POINT, unsigned int > Vocabulary

  Vocabulary of distinct words in the document. The map key is the datum, the mapped value is the number of times the datum appears in the document.
- double alpha

  Concentration parameter for the Dirichlet process ruling the document.
- unsigned int Nj

  Number of words contained in the document.
- vector< double > Pi

  Vector of group specific topics' weights.
- vector< double > PiLeft

  Vector of group specific weights for left subtopics.
- vector< double > PiRight

  Vector of group specific weights for right subtopics.
- vector< unsigned int > WordCount

  K-dimensional vector of counts: the k-th element represents the number of data in topic k (n_{jk})
- vector< unsigned int > WordCountLeft

  K-dimensional vector of counts: the k-th element represents the number of data in the left subtopic of topic k (n_{jkl})
- vector< unsigned int > WordCountRight

  K-dimensional vector of counts: the k-th element represents the number of data in the right subtopic of topic k (n_{jkr})
- vector< unsigned int > LocalTable

  K-dimensional vector for tables: the k-th element represents the number of table in the restaurant serving dish k (m_{jk})
- vector< unsigned int > LocalTableLeft

  K-dimensional vector for tables: the k-th element represents the number of table in the restaurant serving dish k left (m_{jkl})
- vector< unsigned int > LocalTableRight

  K-dimensional vector for tables: the k-th element represents the number of table in the restaurant serving dish k right (m_{jkr})

5.4.1 Detailed Description

template<unsigned int DIM = 1>class CategoricalDocument< DIM >

Derived class for topic modeling, where data are categorical and the base measure is the Dirichlet distribution. This class represents a document in the topic modeling problem. It manages the words and is in charge of sampling the topic labels. It samples the model's parameters specific to the document: $\alpha, \pi_j, (\bar{\pi})_{jl}, (\bar{\pi})_{jr}, m_j, \bar{m}_{jl}, \bar{m}_{jl}$. It keep track of words' counts in topics.

Authors

{Debora Parisi and Stefania Perego}

Date

Febbraio 2016

5.4.2 Constructor & Destructor Documentation

5.4.2.1 template<unsigned int DIM = 1> CategoricalDocument< DIM >::CategoricalDocument (double *_alpha*) [inline]

Constructor.

Parameters

_alpha	- concentration parameter of the Dirichlet process governing the document

5.4.3 Member Function Documentation

5.4.3.1 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::CheckLeftSubcluster (const ClusterID *_k*)

Verifies if topic k has an empty left subtopic.

Parameters

_k	- topic id

5.4.3.2 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::CheckRightSubcluster (const ClusterID *_k*)

Verifies if topic k has an empty right subtopic.

Parameters

_k	- topic id

5.4.3.3 template<unsigned int DIM> void CategoricalDocument< DIM >::InsertNewCluster (const pair< unordered_map< POINT, unsigned int >, unordered_map< POINT, unsigned int > > & *NewCluster,* const double *_Pi,* const double *_PiLeft,* const double *_PiRight,* const unsigned int *_WordCount,* const unsigned int *_WordCountLeft,* const unsigned int *_WordCountRight,* const unsigned int *_LocalTable,* const unsigned int *_LocalTableLeft,* const unsigned int *_LocalTableRight*)

Insert a new topic.

Parameters

NewCluster	- the new topic
_Pi	- document specific weight for the new topic
_PiLeft	- document specific left subtopic weight for the new topic
_PiRight	- document specific right subtopic weight for the new topic
_WordCount	- number of words in the new topic
_WordCountLeft	- number of words in the left subtopic of the new topic
_WordCount-Right	- number of words in the right subtopic of the new topic
_LocalTable	- number of tables serving the new dish in restaurant j (CRF metaphor) m_{jk}
_LocalTableLeft	- number of tables serving the new left dish in restaurant j (CRF metaphor) m_{jkl}
_LocalTable-Right	- number of tables serving the new right dish in restaurant j (CRF metaphor) m_{jkr}

5.4.3.4 template<unsigned int DIM> void CategoricalDocument< DIM >::RemoveCluster (const vector< ClusterID > & *_k*) [virtual]

Remove topics with id contained in the input vector.

Parameters

_k	- vector containing the id of topics to eliminate

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.5 template<unsigned int DIM> void CategoricalDocument< DIM >::RemoveCluster (const ClusterID *_k*)

Remove topic k.

Parameters

_k	- topic id

5.4.3.6 template<unsigned int DIM> void CategoricalDocument< DIM >::RemoveCluster (ClusterID *_k1*, ClusterID *_k2*)

Remove two topics.

Parameters

_k1	- topic id
_k2	- topic id

5.4.3.7 template<unsigned int DIM> void CategoricalDocument< DIM >::ResetDataCountSub (const ClusterID *k*)

Reset wourds counts in topic k.

Parameters

k	- topic id

5.4.3.8 template<unsigned int DIM> void CategoricalDocument< DIM >::Sampling (std::vector< unsigned int > & *_temp_counts*, std::vector< double > & *_Weights*, unsigned int *_nidj*, omprng & *Gen*) [private], [virtual]

Samples topic's or subtopic's label from categorical distribution.

Parameters

_temp_counts	- vector containing count for word id in all topics
_Weights	- weights for sampling labels
_nidj	- number of times word id appears in document j
Gen	- parallel random number generator

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.9 template<unsigned int DIM> void CategoricalDocument< DIM >::SetAlpha (const double *_alpha*) [virtual]

Set the concentration parameter α of the Dirichlet process governing the document.

Parameters

_alpha	- α

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.10 template<unsigned int DIM> void CategoricalDocument< DIM >::SetDataset (std::istringstream & *SSTR*) [virtual]

Acquires data.

Parameters

SSTR	- contains words id and counts in the document

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.11 template<unsigned int DIM> void CategoricalDocument< DIM >::SetNj (const unsigned int *_Nj*) [virtual]

Set the number of words in document j.

Parameters

_Nj	- number of words in document j

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.12 template<unsigned int DIM> void CategoricalDocument< DIM >::SetPi (vector< double > & *_Pi*) [virtual]

Set the vector containing the document specific topics weights.

Parameters

_pi	- vector containing the document specific topics weights

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.13 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::SortData (unsigned int *_K*, omprng & *Gen*) [virtual]

Allocates words in the Zeta container.

Parameters

_K	- initial number of topics
Gen	- parallel random number generator

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.14 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateAllLocalTableSub (const vector< long double > & *_stirling*, const vector< double > & *_BetaLeft*, const vector< double > & *_BetaRight*, omprng & *Gen*) [virtual]

Updates table counts for subtopic of all topics.

Parameters

_stirling	- Stirling numbers
_BetaLeft	- global weight for left subtopics of all topics
_BetaRight	- global weight for right subtopics of all topics
Gen	- parallel random number generator

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.15 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateAllPiSub (const vector< double > *_BetaLeft*, const vector< double > *_BetaRight*, omprng & *Gen*) [virtual]

Updates document specific subtopics' weights for all topics.

Parameters

_BetaLeft	- global weights for left subtopics
_BetaRight	- global weights for right subtopics
Gen	- parallel random number generator

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.16 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateLocalTable (const vector< long double > & *_stirling*, const vector< double > & *_Beta*, omprng & *Gen*) [virtual]

Updates table counts.

Parameters

_stirling	- Stirling numbers
_Beta	- topics' global weights
Gen	- parallel random number generator

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.17 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateLocalTableSub_OneCluster (const vector< long double > & *_stirling*, const double *_BetaLeft*, const double *_BetaRight*, const ClusterID *k*, omprng & *Gen*)

Updates table counts for subtopics of topic k.

Parameters

_stirling	- Stirling numbers
_BetaLeft	- global weight for left subtopic of topic k
_BetaRight	- global weight for right subtopic of topic k
k	- topic id
Gen	- parallel random number generator

5.4.3.18 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdatePi (const vector< double > & *_AllBeta,* omprng & *Gen*) [virtual]

Update document specific topics' weights.

Parameters

_AllBeta	- global topics' weights
Gen	- parallel random number generator

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.19 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdatePiSub (const double *_BetaLeft,* const double *_BetaRight,* const ClusterID *k,* omprng & *Gen*)

Updates document specific weights for subtopics of topic k.

Parameters

_BetaLeft	- global weight for left subtopic of topic k
_BetaRight	- global weight for right subtopic of topic k
k	- topic id
Gen	- parallel random number generator

5.4.3.20 template<unsigned int DIM = 1> void CategoricalDocument< DIM >::UpdateZeta (const THETA & *_ThetaId,* const POINT *_VettId,* omprng & *Gen*)

Updates topic label of a datum, sampled with the Sampling method.

Parameters

_ThetaId	- vector of a datum's weights in all topics
_VettId	- datum id
Gen	- parallel random number generator

5.4.3.21 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateZeta (const ClusterID *_k*)

Method needed to remove topic k during M-H moves.

Parameters

_k	- topic id

5.4.3.22 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateZeta (const ClusterID *_k1,* const ClusterID *_k2*)

Method needed to remove two topics during M-H moves.

Parameters

_k1	- topic id
_k2	- topic id

5.4.3.23 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateZeta_and_Sub (const THETA & *_ThetaId,* const THETA & *_ThetaIdLeft,* const THETA & *_ThetaIdRight,* const unsigned int *_VettId,* omprng & *Gen*)

Updates topic label of a datum, sampled with with the Sampling method.

Parameters

_ThetaId	- vector of a datum's weights in all topics
_ThetaIdLeft	- vector of a datum's weights in all left subtopics
_ThetaIdRight	- vector of a datum's weights in all right subtopics
_VettId	- datum id
Gen	- parallel random number generator

5.4.3.24 template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateZetaSub (const THETA & *_ThetaIdLeft,* const THETA & *_ThetaIdRight,* const POINT *id,* const unsigned int *nidjk,* const ClusterID *k,* omprng & *Gen*)

Assigns the datum to the subtopics of topic k, after sampling the subtopics' label with the Sampling method.

Parameters

_ThetaIdLeft	- vector of a datum's weights in all left subtopics
_ThetaIdRight	- vector of a datum's weights in all right subtopics
id	- datum id
nidjk	- number of times datum id in document j is assigned to topic k
k	- topic id
Gen	- parallel random number generator

5.4.3.25 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewCluster (const ClusterID *_k,* pair< unordered_map< POINT, unsigned int >, unordered_map< POINT, unsigned int >> & *_Cluster*)

Retrieve topic k.

Parameters

_k	- topic id
_Cluster	- structure that will contain topic k

5.4.3.26 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewCounts4c (ClusterID *_k,* STAT & *_counts4cleft,* STAT & *_counts4cright*)

Retrieve counts necessary for updating subtopics' latent parameters.

Parameters

_k	- topic id

_counts4cleft	- counts for left subtopic's latent parameter
_counts4cright	- counts for right subtopic's latent parameter

5.4.3.27 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewData (vector< POINT > & *_VettId*) const

Retrieve data id in document j.

Parameters

_VettId	- filled with data id in document j

5.4.3.28 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewDataCount (vector< unsigned int > & *_WordCount*) const [virtual]

Retrieve a vector containing the number of words in all topics.

Parameters

_WordCount	- vector containing the number of words in all

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.29 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewDataCountID (const ClusterID *_k*) const

Retrieve number of words in topic k $n_{.k}$.

Parameters

_k	- topic k

Returns

$n_{.k}$

5.4.3.30 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewDataCountLeft (vector< unsigned int > & *_WordCountLeft*) const [virtual]

Retrieve a vector containing the number of words in all left subtopics.

Parameters

_WordCountLeft	- vector containing the number of words in all left subtopics

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.31 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewDataCountLeftID (const ClusterID *_k*) const

Retrieve number of topic in left subtopic of topic k $n_{.kl}$.

Parameters

_k	- topic id

Returns

$n_{.kl}$

5.4.3.32 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewDataCountRight (vector< unsigned int > & *_WordCountRight*) const [virtual]

Retrieve a vector containing the number of words in all right subtopics.

Parameters

_WordCount-Right	- vector containing the number of words in all right subtopics

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.33 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewDataCountRightID (const ClusterID *_k*) const

Retrieve number of topic in right subtopic of topic k $n_{.kr}$.

Parameters

_k	- topic id

Returns

$n_{.kr}$

5.4.3.34 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewIdCounts (vector< pair< POINT, unsigned int >> & *_nidjk,* const ClusterID *_k*)

Retrieve counts and id of words in topic k.

Parameters

_nidjk	- structure that will contain counts and id of words in topic k
_k	- topic id

5.4.3.35 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewLabel (vector< pair< POINT, ClusterID >> & *Data*)

Retrieve labels assigned to words.

Parameters

Data	- structure to store retrieved labels

5.4.3.36 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewNj ( ) const [virtual]

Retrieve number of data in document j.

Returns

Number of data in document j

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.37 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewNumTableID (const ClusterID _k) const

Retrieve number of tables for topic k m_{jk}.

Parameters

_k	- topic id

Returns

m_{jk}

5.4.3.38 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewNumTableLeftID (const ClusterID _k) const

Retrieve number of tables for left subtopic of topic k m_{jkl}.

Parameters

_k	- topic id

Returns

m_{jk}

5.4.3.39 template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewNumTableRightID (const ClusterID _k) const

Retrieve number of tables for right subtopic of topic k m_{jkr}.

Parameters

_k	- topic id

Returns

m_{jkr}

5.4.3.40 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewPi (vector< double > & _pi) const [virtual]

Retrieve a vector containing the document specific topic weights.

Parameters

_pi	- vector containing the document specific topic weights

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.41 template<unsigned int DIM> double CategoricalDocument< DIM >::ViewPiID (const ClusterID _k) const

Retrieve document specific weight of topic k π_{jk}.

Parameters

_k	- topic id

Returns

π_{jk}

5.4.3.42 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewPiLeft (vector< double > & *_pi*) const [virtual]

Retrieve a vector containing the document specific weights for left subtopics.

Parameters

_pi_left	- vector containing the document specific weights for left subtopics

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.43 template<unsigned int DIM> double CategoricalDocument< DIM >::ViewPiLeftID (const ClusterID _k) const

Retrieve document specific weight for the left subtopic of topic k π_{jkl}.

Parameters

_k	- topic id

Returns

π_{jkl}

5.4.3.44 template<unsigned int DIM> void CategoricalDocument< DIM >::ViewPiRight (vector< double > & *_pi*) const [virtual]

Retrieve a vector containing the document specific weights for right subtopics.

Parameters

_pi_right	- vector containing the document specific weights for right subtopics

Implements GenericDocument< TypeCategorical< DIM >, DIM >.

5.4.3.45 template<unsigned int DIM> double CategoricalDocument< DIM >::ViewPiRightID (const ClusterID _k) const

Retrieve document specific weight for the right subtopic of topic k π_{jkr}.

Parameters

_k	- topic id

Returns

π_{jkr}

The documentation for this class was generated from the following file:

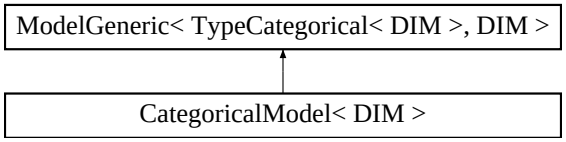
- Document.hpp

5.5 CategoricalModel< DIM > Class Template Reference

Specialized class for topic modeling, where data are categorical and H is the Dirichlet distribution. This class is used to sample latent parameters from the Dirichlet distribution, to compute likelihood and marginal distribution for categorical data. It also removes and adds topics and print values of latent parameters on file.

`#include <Model.hpp>`

Inheritance diagram for CategoricalModel< DIM >:



Public Types

- using THETA = TypeCategorical< 1 >::THETA

  Latent parameter: weights of distintc words in a topic.
- using POINT = TypeCategorical< 1 >::Point

  A datum.
- using STAT = TypeCategorical< 1 >::STAT

  Statistics to updated hyperparameters for latent parameter's distribution, namely the number of data assigned to topic and subtopic.
- using HYP = TypeCategorical< 1 >::HYP

  Hyperparameters for latent parameter's distribution.

Public Member Functions

- CategoricalModel ()

  Default constructor.
- CategoricalModel (const CategoricalModel &mod)=default

  Copy costructor.
- CategoricalModel (const CategoricalModel &&mod)

  Move constructor.
- ~CategoricalModel ()=default

  Destructor.
- CategoricalCluster & operator[] (unsigned int _K)

  Operator to access directly to methods of cluster k.
- CategoricalModel & operator= (const CategoricalModel &mod)

Assignment operator.

- CategoricalModel & operator= (CategoricalModel &&mod)

  Move assignement operator.

- void SetHyperparameter (const HYP &_Lambda)

  Fixes hyperparameters for the latent parameters' distribution.

- void DefaultHyperparameter (size_t W)

  Fixes hyperparameters for the latent parameters' distribution with default values.

- void SetInitialClusters (unsigned int _K)

  Sets topics assigning initial values for their weights.

- double Marginalized_Loglikelihood (const unsigned int _K)

  Computes the logarithm of the topic's marginal likelihood, given the topic id.

- double Loglikelihood (const POINT X, const unsigned int _K)

  Computes the loglikelihood of datum X, given the topic id.

- double LoglikelihoodLeft (const POINT X, const unsigned int _K)

  Computes the loglikelihood of datum X, given the left subtopic id.

- double LoglikelihoodRight (const POINT X, const unsigned int _K)

  Computes the loglikelihood of datum X, given the right subtopic id.

- long double LogDensity (const unsigned int _K)

  Computes the latent parameter's density, given the topic id.

- void UpdateThetaCluster (omprng &Gen)

  Updates latent parameters of all topics.

- void UpdateThetaSubCluster (omprng &Gen)

  Updates latent parameters of left and right subtopics of all topics.

- void UpdateOneThetaCluster (const unsigned int _K, omprng &Gen)

  Updates latent parameters of one topic, given its id.

- void UpdateOneThetaSubCluster (const unsigned int _K, omprng &Gen)

  Updates latent parameters of left and right subtopics of one topic, given its id.

- void AddOneCluster (const unsigned int _k)

  Adds an empty topic to the current topics and updates the current number of topics.

- void RemoveOneCluster (const unsigned int _K)

  Removes a topic given its id and updates the current number of topics.

- void RemoveClusters (const vector< unsigned int > &_K)

  Removes multiple topics given their id and updates the current number of topics.

- unsigned int ViewK () const

  Retrieves current number of topics.

- void ViewKey (vector< unsigned int > &Key) const

  Retrieves current topics' ids.

- void ViewBeta (vector< double > &_AllBeta)

  Retrieves current topics' global weights.

- void ViewBetaLeft (vector< double > &_AllBetaLeft)

  Retrieves global weights for left subtopics of current topics.

- void ViewBetaRight (vector< double > &_AllBetaRight)

  Retrieves global weights for right subtopics of current topics.

- void PrintTheta (const std::string &FileName)

  Print to file values of current topics' latent parameters.

- void PrintLambdaInfo () const

  Print to screen information about hyperparameters of latent parameters' distribution.

Private Attributes

- HYP Lambda

  Vector of hyperparameters.
- unsigned int K

  Current number of clusters.
- unordered_map< unsigned int, CategoricalCluster > Clusters

  Container for clusters.
- unsigned int OMP_NUM_THREADS

  Number of threads.

5.5.1 Detailed Description

template<unsigned int DIM = 1>class CategoricalModel< DIM >

Specialized class for topic modeling, where data are categorical and H is the Dirichlet distribution. This class is used to sample latent parameters from the Dirichlet distribution, to compute likelihood and marginal distribution for categorical data. It also removes and adds topics and print values of latent parameters on file.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.5.2 Member Function Documentation

5.5.2.1 template<unsigned int DIM> void CategoricalModel< DIM >::AddOneCluster (const unsigned int _k) [virtual]

Adds an empty topic to the current topics and updates the current number of topics.

Parameters

_K	- new topic id

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.2 template<unsigned int DIM> void CategoricalModel< DIM >::DefaultHyperparameter (size_t W) [virtual]

Fixes hyperparameters for the latent parameters' distribution with default values.

Parameters

W	- hyperparameters' dimension

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.3 template<unsigned int DIM> long double CategoricalModel< DIM >::LogDensity (const unsigned int _K) [virtual]

Computes the latent parameter's density, given the topic id.

Parameters

_K	- topic id

Returns

logdensity of topic k

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.4 template<unsigned int DIM> double CategoricalModel< DIM >::Loglikelihood (const POINT X, const unsigned int _K)

Computes the loglikelihood of datum X, given the topic id.

Parameters

X	- datum
_K	- topic id

Returns

loglikelihood of datum X in topic k

5.5.2.5 template<unsigned int DIM> double CategoricalModel< DIM >::LoglikelihoodLeft (const POINT X, const unsigned int _K)

Computes the loglikelihood of datum X, given the left subtopic id.

Parameters

X	- datum
_K	- left subtopic id

Returns

loglikelihood of datum X in left subtopic of topic k

5.5.2.6 template<unsigned int DIM> double CategoricalModel< DIM >::LoglikelihoodRight (const POINT X, const unsigned int _K)

Computes the loglikelihood of datum X, given the right subtopic id.

Parameters

X	- datum
_K	- right subtopic id

Returns

loglikelihood of datum X in right subtopic of topic k

5.5.2.7 template<unsigned int DIM> double CategoricalModel< DIM >::Marginalized_Loglikelihood (const unsigned int _K) [virtual]

Computes the logarithm of the topic's marginal likelihood, given the topic id.

Parameters

_K	- topic id

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.8 template<unsigned int DIM> CategoricalModel< DIM > & CategoricalModel< DIM >::operator= (const CategoricalModel< DIM > & *mod*)

Assignment operator.

Parameters

mod	- object of class CategoricalModel

5.5.2.9 template<unsigned int DIM> CategoricalModel< DIM > & CategoricalModel< DIM >::operator= (CategoricalModel< DIM > && *mod*)

Move assignement operator.

Parameters

mod	- object of class CategoricalModel

5.5.2.10 template<unsigned int DIM> CategoricalCluster & CategoricalModel< DIM >::operator[] (unsigned int *_K*)

Operator to access directly to methods of cluster k.

Parameters

_K	- cluster id

5.5.2.11 template<unsigned int DIM> void CategoricalModel< DIM >::RemoveClusters (const vector< unsigned int > & *_K*) [virtual]

Removes multiple topics given their id and updates the current number of topics.

Parameters

_K	- vector of topics' ids to be removed

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.12 template<unsigned int DIM> void CategoricalModel< DIM >::RemoveOneCluster (const unsigned int *_K*) [virtual]

Removes a topic given its id and updates the current number of topics.

Parameters

_K	- id of the topic to be removed

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.13 template<unsigned int DIM> void CategoricalModel< DIM >::SetHyperparameter (const HYP & *_Lambda*)

Fixes hyperparameters for the latent parameters' distribution.

Parameters

_Lambda	- hyperparameters for the latent parameters' distribution

5.5.2.14 template<unsigned int DIM> void CategoricalModel< DIM >::SetInitialClusters (unsigned int *_K*) [virtual]

Sets topics assigning initial values for their weights.

Parameters

_K	- Initial number of topics

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.15 template<unsigned int DIM> void CategoricalModel< DIM >::UpdateOneThetaCluster (const unsigned int *_K*, ompprng & *Gen*) [virtual]

Updates latent parameters of one topic, given its id.

Parameters

_K	- topic id
Gen	- parallel random number generator

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.16 template<unsigned int DIM> void CategoricalModel< DIM >::UpdateOneThetaSubCluster (const unsigned int *_K*, ompprng & *Gen*) [virtual]

Updates latent parameters of left and right subtopics of one topic, given its id.

Parameters

_K	- topic id
Gen	- parallel random number generator

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.17 template<unsigned int DIM> void CategoricalModel< DIM >::UpdateThetaCluster (ompprng & *Gen*) [virtual]

Updates latent parameters of all topics.

Parameters

Gen	- parallel random number generator

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.18 template<unsigned int DIM> void CategoricalModel< DIM >::UpdateThetaSubCluster (ompprng & *Gen*) [virtual]

Updates latent parameters of left and right subtopics of all topics.

Parameters

Gen	- parallel random number generator

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.19 template<unsigned int DIM> void CategoricalModel< DIM >::ViewBeta (vector< double > & *_AllBeta*) [virtual]

Retrieves current topics' global weights.

Parameters

AllBeta	- vector that will be filled with global weights

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.20 template<unsigned int DIM> void CategoricalModel< DIM >::ViewBetaLeft (vector< double > & *_AllBetaLeft*) [virtual]

Retrieves global weights for left subtopics of current topics.

Parameters

AllBetaLeft	- vector that will be filled with global weights of left subtopics

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.21 template<unsigned int DIM> void CategoricalModel< DIM >::ViewBetaRight (vector< double > & *_AllBetaRight*) [virtual]

Retrieves global weights for right subtopics of current topics.

Parameters

AllBetaRight	- vector that will be filled with global weights of right subtopics

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.22 template<unsigned int DIM> unsigned int CategoricalModel< DIM >::ViewK ( ) const [virtual]

Retrieves current number of topics.

Returns

current number of topics

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

5.5.2.23 template<unsigned int DIM> void CategoricalModel< DIM >::ViewKey (vector< unsigned int > & *Key*) const [virtual]

Retrieves current topics' ids.

Parameters

Key	- object where current topics' ids are stored

Implements ModelGeneric< TypeCategorical< DIM >, DIM >.

The documentation for this class was generated from the following file:

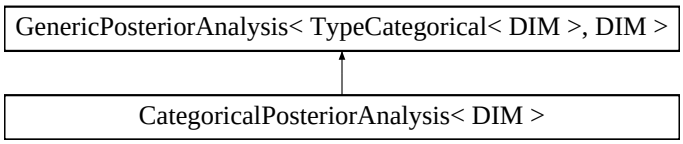
- Model.hpp

5.6 CategoricalPosteriorAnalysis< DIM > Class Template Reference

Class for the posterior analysis when data are categorical and H is the Dirichlet distribution. Reads and stores the results in suitable structures.

`#include <PosteriorAnalysis.hpp>`

Inheritance diagram for CategoricalPosteriorAnalysis< DIM >:



Public Types

- using THETA = TypeCategorical< 1 >::THETA

  Latent parameter: vector of topic's weights for the distinct words.
- using ClusterId = unsigned int

  Topic id.
- using GroupId = unsigned int

  Document id.
- using DataId = unsigned int

  Word id.
- using Check = unsigned int

  Control variable.

Public Member Functions

- CategoricalPosteriorAnalysis ()=default

  Default constructor.
- ~CategoricalPosteriorAnalysis ()=default

  Destructor.
- void SetAllK ()

  Sets the chain of the number of topics.
- void SetAllAlpha ()

  Sets the chain of the values for α.
- void SetAllGamma ()

  Sets the chain of the values for γ.
- void SetVocabulary ()

  Sets the vocabulary.
- void KPosteriorAnalysis (RInside &R, const unsigned int Burnin, const unsigned int Thinning)

  Calls the R script for the analysis of the K chain.

- void AGPosteriorAnalysis (RInside &R, const unsigned int AlphaBurnin, const unsigned int AlphaThinning, const unsigned int GammaBurnin, const unsigned int GammaThinning, const char AlphaTry, const char GammaTry)

  Calls the R script for the analyis of Alpha and Gamma chains.
- void Setwd (const std::string &_wd)

  Sets the R working directory.
- void SetW (const unsigned int _W)

  Sets the number of distintc words.
- void SetD (const unsigned int _D)

  Sets the number of documents.
- void SetN (const unsigned int _N)

  Set the total number of words.
- void UnioneLabels (RInside &R)

  Skims the Labels∗.bin files and joins them in the Labels.bin file.
- void LeastSquareClustering ()

  Computes least-square clustering.
- void WriteBestParams ()

  Writes on binary files θ and β parameters of the optimal iteration.
- void WordClouds (RInside &R)

  Draws the wordclouds representing the estimated topics in the optimal iteration.
- void SetDocs ()

  Acquires documents' names.
- void AssociatingDocs ()

  Associates documents to estimated topics.
- void SetIterations (const unsigned int _Iterations)

  Sets the number of iterations for the MCMC chain.
- void Setburnin (const unsigned int _burnin)

  Sets the number of iterations to discard in order to compute the LPML.
- void SetBT (const unsigned int _Burnin, const unsigned int _Thinning)

  Sets burnin and thinning for the K chain.
- void LPML ()

  Computes and prints on the terminal the LPML index.

Private Attributes

- unordered_map< DataId, std::string > Vocabulary

  Corpus' vocabulary.
- vector< double > Alpha

  Containes the values taken on by α in each iteration, when α is random.
- vector< double > Gamma

  Containes the values taken on by γ in each iteration, when γ is random.
- vector< unsigned int > AllK

  Contains the number of topics inferred in each iteration.
- std::string wd

  R working directory.
- unsigned int W

  Number of distinct words.
- unsigned int N

  Total number of words.
- unsigned int D

Number of documents.

- unsigned int Iterations

  Number of iteration in the MCMC chain.

- unsigned int burnin

  Number of initial iteration to discard in order to compute the LPML.

- unsigned int Burnin

  Number of initial iteration to discard in the K chain.

- unsigned int Thinning

  Thinning for the K chain.

- unsigned int Iter_after

  Letf iterations after skimming the chain.

- unsigned int iter_opt

  Optimal iteration according to the least square clustering method.

- unsigned int file_nr

  Files where looking for best parameters.

- unsigned int move

  Number of iterations to be skipped in those files.

- unsigned int K_opt

  Optimal number of topics.

- vector< std::string > docs

  Structure that stores the documents' names.

5.6.1 Detailed Description

template<unsigned int DIM = 1>class CategoricalPosteriorAnalysis< DIM >

Class for the posterior analysis when data are categorical and H is the Dirichlet distribution. Reads and stores the results in suitable structures.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.6.2 Member Function Documentation

5.6.2.1 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::AGPosteriorAnalysis (RInside & *R*, const unsigned int *AlphaBurnin*, const unsigned int *AlphaThinning*, const unsigned int *GammaBurnin*, const unsigned int *GammaThinning*, const char *AlphaTry*, const char *GammaTry*) [virtual]

Calls the R script for the analyis of Alpha and Gamma chains.

Parameters

R	- R istance
AlphaBurnin	- number of initial values in the Alpha chain to be discarded
AlphaThinning	- keep a value in the Alpha chain every AlphaThinning values

GammaBurnin	- number of initial values in the Gamma chain to be discarded
GammaThinning	- keep a value in the Gamma chain every GammaThinning values
AlphaTry	- yes if you want to repeat the Alpha chain's analysis, no otherwise
GammaTry	- yes if you want to repeat the Gamma chain's analysis, no otherwise

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.2 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::KPosteriorAnalysis (RInside & *R*, const unsigned int *Burnin*, const unsigned int *Thinning*) [virtual]

Calls the R script for the analysis of the K chain.

Parameters

R	- R istance
Burnin	- number of initial values in the chain to be discarded
Thinning	- keep a value in the chain every thinning values

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.3 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::SetBT (const unsigned int *_Burnin*, const unsigned int *_Thinning*)

Sets burnin and thinning for the K chain.

Parameters

_Burnin	- burnin
_Thinning	- thinning

5.6.2.4 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::Setburnin (const unsigned int *_burnin*) [virtual]

Sets the number of iterations to discard in order to compute the LPML.

Parameters

_burnin	- iterations to discard in order to compute the LPML

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.5 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::SetD (const unsigned int *_D*) [virtual]

Sets the number of documents.

Parameters

_D	- number of documents

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.6 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::SetIterations (const unsigned int *_Iterations*) [virtual]

Sets the number of iterations for the MCMC chain.

Parameters

_Iterations	- number of iterations

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.7 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::SetN (const unsigned int *_N*) [virtual]

Set the total number of words.

Parameters

_N	- total number of words

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.8 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::SetW (const unsigned int *_W*) [virtual]

Sets the number of distintc words.

Parameters

_W	- number of distintc words

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.9 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::Setwd (const std::string & *_wd*) [virtual]

Sets the R working directory.

Parameters

_wd	- R working directory

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.10 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::UnioneLabels (RInside & *R*) [virtual]

Skims the Labels∗.bin files and joins them in the Labels.bin file.

Parameters

R	- object of class RInside

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

5.6.2.11 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::WordClouds (RInside & *R*)

Draws the wordclouds representing the estimated topics in the optimal iteration.

Parameters

R	- object of class RInside

5.6.2.12 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::WriteBestParams ( ) [virtual]

Writes on binary files θ and β parameters of the optimal iteration.

Parameters

file_nr	- files in which looking for parameters
move	- in those files, number of initial iterations to skip

Returns

number of topics in the best clustering

Implements GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >.

The documentation for this class was generated from the following file:

- PosteriorAnalysis.hpp

5.7 CLUSTER Struct Reference

Structure for a cluster.

`#include <Struct.hpp>`

Public Attributes

- pair< unordered_map< POINT, unsigned int >, unordered_map < POINT, unsigned int > > **b**
- pair< unordered_map< POINT, unsigned int >, unordered_map < POINT, unsigned int > > **c**
- pair< unordered_map< POINT, unsigned int >, unordered_map < POINT, unsigned int > > **a**
- unordered_map< POINT, unsigned int > **a_sx**
- unordered_map< POINT, unsigned int > **a_dx**

5.7.1 Detailed Description

Structure for a cluster.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

The documentation for this struct was generated from the following file:

- Struct.hpp

5.8 DATACOUNT Struct Reference

Structure for data counts.

`#include <Struct.hpp>`

Public Attributes

- vector< unsigned int > **b**
- vector< unsigned int > **c**
- vector< unsigned int > **a**

5.8.1 Detailed Description

Structure for data counts.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

The documentation for this struct was generated from the following file:

- Struct.hpp

5.9 GenericCluster< ClassType, DIM > Class Template Reference

Generic Model of Cluster.

`#include <Cluster.hpp>`

Public Member Functions

- virtual void SetTheta (typename ClassType< DIM >::THETA &)=0

  It sets the latent parameter of cluster.
- virtual void SetThetaLeft (typename ClassType< DIM >::THETA &)=0

  It sets the left sub-cluster latent parameter.
- virtual void SetThetaRight (typename ClassType< DIM >::THETA &)=0

  It sets the right sub-cluster latent parameter.
- virtual void SetBeta (double)=0

  It sets the weigth of cluster.
- virtual void SetBetaLeft (double)=0

  It sets the weigth of left subcluster.
- virtual void SetBetaRight (double)=0

  It sets the weigth of right subcluster.
- virtual void SetGlobalTable (unsigned int)=0

  It sets the number of global tables which characterize the cluster.
- virtual void SetGlobalTableLeft (unsigned int)=0

  It sets the number of global tables which characterize the left sub-cluster.

- virtual void SetGlobalTableRight (unsigned int)=0

  It sets the number of global tables which characterize the right sub-cluster.
- virtual void SetStatistics (typename ClassType< DIM >::STAT &)=0

  It sets the statistics, in other words the hyperparameter of cluster's latent paramenter.
- virtual void SetStatisticsLeft (typename ClassType< DIM >::STAT &)=0

  It sets the statistics, in other words the hyperparameter of left sub-cluster's latent paramenter.
- virtual void SetStatisticsRight (typename ClassType< DIM >::STAT &)=0

  It sets the statistics, in other words the hyperparameter of right sub-cluster's latent paramenter.
- virtual void ViewTheta (typename ClassType< DIM >::THETA &) const =0

  It extracts information about cluster's latent parameter.
- virtual void ViewThetaLeft (typename ClassType< DIM >::THETA &) const =0

  It extracts information about left sub-cluster's latent parameter.
- virtual void ViewThetaRight (typename ClassType< DIM >::THETA &) const =0

  It extracts information about right sub-cluster's latent parameter.
- virtual double ViewBeta () const =0
- virtual double ViewBetaLeft () const =0
- virtual double ViewBetaRight () const =0
- virtual void ViewStatistics (typename ClassType< DIM >::STAT &) const =0

  It extracts infromation about statistics of cluster's latent paramenter.
- virtual void ViewStatisticsLeft (typename ClassType< DIM >::STAT &) const =0

  It extracts infromation about statistics of left sub-cluster's latent paramenter.
- virtual void ViewStatisticsRight (typename ClassType< DIM >::STAT &) const =0

  It extracts infromation about statistics of right sub-cluster's latent paramenter.
- virtual unsigned int ViewGlobalTable () const =0

  It extracts the number of cluster's global table.
- virtual unsigned int ViewGlobalTableLeft () const =0

  It extracts the number of left sub-cluster's global table.
- virtual unsigned int ViewGlobalTableRight () const =0

  It extracts the number of right sub-cluster's global table.
- virtual void ResetStatistics (unsigned int)=0

  It resests the cluster's statistics.
- virtual void ResetStatisticsLeft (unsigned int)=0

  It resests the left sub-cluster's statistics.
- virtual void ResetStatisticsRight (unsigned int)=0

  It resests the right sub-cluster's statistics.
- virtual void UpdateStatistics (typename ClassType< DIM >::STAT &, typename ClassType< DIM >::STAT &)=0

  It updates the statistics, in other words hyperparamenters of cluster and sub-clusters.
- virtual bool IsEmpty () const =0

  Check if cluster is empty.

5.9.1 Detailed Description

template<template< unsigned int > class ClassType, unsigned int DIM>class GenericCluster< ClassType, DIM >

Generic Model of Cluster.

Abstract class where all methods are virtual and they are null. All classes that inherit form Cluster Generic are used to extract, to memorize and to set all data which describe a cluster and its subclusters. This classes have the role only to manage the clusters, in this class there anern't any sample function.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.9.2 Member Function Documentation

5.9.2.1 template<template< unsigned int > class ClassType, unsigned int DIM> virtual bool GenericCluster< ClassType, DIM >::IsEmpty ( ) const [pure virtual]

Check if cluster is empty.

Returns

TRUE if cluster is empty otherwise FALSE

Implemented in CategoricalCluster.

5.9.2.2 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ResetStatistics (unsigned *int*) [pure virtual]

It resests the cluster's statistics.

Parameters

W	- dimenstion of statistics with to update hyperparamenters of cluster's latent paramenter

Implemented in CategoricalCluster.

5.9.2.3 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ResetStatisticsLeft (unsigned *int*) [pure virtual]

It resests the left sub-cluster's statistics.

Parameters

W	- dimenstion of statistics with to update hyperparamenters of left sub-cluster's latent paramenter

Implemented in CategoricalCluster.

5.9.2.4 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ResetStatisticsRight (unsigned *int*) [pure virtual]

It resests the right sub-cluster's statistics.

Parameters

W	- dimenstion of statistics with to update hyperparamenters of right sub-cluster's latent paramenter

Implemented in CategoricalCluster.

5.9.2.5 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetBeta (double) [pure virtual]

It sets the weigth of cluster.

Parameters

_Beta	- Weigth in input

Implemented in CategoricalCluster.

5.9.2.6 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetBetaLeft (double) [pure virtual]

It sets the weigth of left subcluster.

Parameters

_BetaLeft	- Weigth in input

Implemented in CategoricalCluster.

5.9.2.7 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetBetaRight (double) [pure virtual]

It sets the weigth of right subcluster.

Parameters

_BetaRight	- Weigth in input

Implemented in CategoricalCluster.

5.9.2.8 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetGlobalTable (unsigned *int*) [pure virtual]

It sets the number of global tables which characterize the cluster.

Parameters

NrTable	- number of tables which characterize the cluster

Implemented in CategoricalCluster.

5.9.2.9 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetGlobalTableLeft (unsigned *int*) [pure virtual]

It sets the number of global tables which characterize the left sub-cluster.

Parameters

NrTableLeft	- number of tables which characterize the left sub-cluster

Implemented in CategoricalCluster.

5.9.2.10 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetGlobalTableRight (unsigned *int*) [pure virtual]

It sets the number of global tables which characterize the right sub-cluster.

Parameters

NrTableRight	- number of tables which characterize the right sub-cluster

Implemented in CategoricalCluster.

5.9.2.11 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetStatistics (typename ClassType< DIM >::STAT &) [pure virtual]

It sets the statistics, in other words the hyperparameter of cluster's latent paramenter.

Parameters

_c	- cluster's statistics

5.9.2.12 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetStatisticsLeft (typename ClassType< DIM >::STAT &) [pure virtual]

It sets the statistics, in other words the hyperparameter of left sub-cluster's latent paramenter.

Parameters

_cLeft	- left sub-cluster's statistics

5.9.2.13 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetStatisticsRight (typename ClassType< DIM >::STAT &) [pure virtual]

It sets the statistics, in other words the hyperparameter of right sub-cluster's latent paramenter.

Parameters

_cRight	- right sub-cluster's statistics

5.9.2.14 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetTheta (typename ClassType< DIM >::THETA &) [pure virtual]

It sets the latent parameter of cluster.

Parameters

_Theta	- latent parameter in input. Its type is THETA

5.9.2.15 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetThetaLeft (typename ClassType< DIM >::THETA &) [pure virtual]

It sets the left sub-cluster latent parameter.

Parameters

_ThetaLeft	- latent parameter in input. Its type is THETA

5.9.2.16 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::SetThetaRight (typename ClassType< DIM >::THETA &) [pure virtual]

It sets the right sub-cluster latent parameter.

Parameters

_ThetaRight	- latent parameter in input. Its type is THETA

5.9.2.17 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::UpdateStatistics (typename ClassType< DIM >::STAT & , typename ClassType< DIM >::STAT &) [pure virtual]

It updates the statistics, in other words hyperparamenters of cluster and sub-clusters.

Parameters

counts4cleft	- statistics to update the hyperparameters of left sub-subcluster's latent parameters
counts4cright	- statistics to update the hyperparameters of right sub-subcluster's latent parameters

5.9.2.18 template<template< unsigned int > class ClassType, unsigned int DIM> virtual double GenericCluster< ClassType, DIM >::ViewBeta ( ) const [pure virtual]

It extracts information about cluster's global weight

Returns

cluster's global weight

Implemented in CategoricalCluster.

5.9.2.19 template<template< unsigned int > class ClassType, unsigned int DIM> virtual double GenericCluster< ClassType, DIM >::ViewBetaLeft ( ) const [pure virtual]

It extracts information about left sub-cluster's global weight

Returns

left sub-cluster's global weight

Implemented in CategoricalCluster.

5.9.2.20 template<template< unsigned int > class ClassType, unsigned int DIM> virtual double GenericCluster< ClassType, DIM >::ViewBetaRight ( ) const [pure virtual]

It extracts information about right sub-cluster's global weight

Returns

right sub-cluster's global weight

Implemented in CategoricalCluster.

5.9.2.21 template<template< unsigned int > class ClassType, unsigned int DIM> virtual unsigned int GenericCluster< ClassType, DIM >::ViewGlobalTable ( ) const [pure virtual]

It extracts the number of cluster's global table.

Returns

Number of global table

Implemented in CategoricalCluster.

5.9.2.22 template<template< unsigned int > class ClassType, unsigned int DIM> virtual unsigned int GenericCluster< ClassType, DIM >::ViewGlobalTableLeft ( ) const [pure virtual]

It extracts the number of left sub-cluster's global table.

Returns

Number of global table in left sub-cluster

Implemented in CategoricalCluster.

5.9.2.23 template<template< unsigned int > class ClassType, unsigned int DIM> virtual unsigned int GenericCluster< ClassType, DIM >::ViewGlobalTableRight ( ) const [pure virtual]

It extracts the number of right sub-cluster's global table.

Returns

Number of global table in right sub-cluster

Implemented in CategoricalCluster.

5.9.2.24 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ViewStatistics (typename ClassType< DIM >::STAT &) const [pure virtual]

It extracts infromation about statistics of cluster's latent paramenter.

Parameters

_c	- It takes in input an object, which its type is STAT. In this object It saves the statistics of cluster's latent paramenter

5.9.2.25 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ViewStatisticsLeft (typename ClassType< DIM >::STAT &) const [pure virtual]

It extracts infromation about statistics of left sub-cluster's latent paramenter.

Parameters

_cLeft	- It takes in input an object, which its type is STAT. In this object It saves the statistics of left sub-cluster's latent paramenter

5.9.2.26 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ViewStatisticsRight (typename ClassType< DIM >::STAT &) const [pure virtual]

It extracts infromation about statistics of right sub-cluster's latent paramenter.

Parameters

_cRight	- It takes in input an object, which its type is STAT. In this object It saves the statistics of right sub-cluster's latent paramenter

5.9.2.27 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ViewTheta (typename ClassType< DIM >::THETA &) const [pure virtual]

It extracts information about cluster's latent parameter.

Parameters

_Theta	- It takes in input an object, which its type is THETA. In this object It saves the value of cluster's latent paramenter

5.9.2.28 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ViewThetaLeft (typename ClassType< DIM >::THETA &) const [pure virtual]

It extracts information about left sub-cluster's latent parameter.

Parameters

_ThetaLeft	- It takes in input an object, which its type is THETA. In this object It saves the value of left sub-cluster's latent paramenter

5.9.2.29 template<template< unsigned int > class ClassType, unsigned int DIM> virtual void GenericCluster< ClassType, DIM >::ViewThetaRight (typename ClassType< DIM >::THETA &) const [pure virtual]

It extracts information about right sub-cluster's latent parameter.

Parameters

_ThetaRight	- It takes in input an object, which its type is THETA. In this object It saves the value of right sub-cluster's latent paramenter

The documentation for this class was generated from the following file:

- Cluster.hpp

5.10 GenericDocument< Type, DIM > Class Template Reference

Generic class for groups of data.

`#include <Document.hpp>`

Public Member Functions

- virtual void UpdatePi (const vector< double > &, omprng &)=0

  Updates group specific clusters' weights.
- virtual void UpdatePiSub (const double, const double, const unsigned int, omprng &)=0

  Updates group specific weights for subclusters of cluster k.
- virtual void UpdateAllPiSub (const vector< double >, const vector< double >, omprng &)=0

  Updates group specific subclusters' weights for all cluster.
- virtual void UpdateLocalTable (const vector< long double > &, const vector< double > &, omprng &)=0

  Updates table counts.
- virtual void UpdateLocalTableSub_OneCluster (const vector< long double > &, const double _, const double, const unsigned int, omprng &)=0

  Updates table counts for subclusters of cluster k.
- virtual void UpdateAllLocalTableSub (const vector< long double > &, const vector< double > &, const vector< double > &, omprng &)=0

  Updates table counts for subclusters of all clusters.
- virtual void UpdateZeta (const typename Type::THETA &, const unsigned int, omprng &)=0

  Updates cluster label of a datum, sampled with the Sampling method.
- virtual void UpdateZeta_and_Sub (const typename Type::THETA &, const typename Type::THETA &, const typename Type::THETA &, const unsigned int, omprng &)=0

  Updates cluster label of a datum, sampled with with the Sampling method.
- virtual void UpdateZetaSub (const typename Type::THETA &, const typename Type::THETA &, const typename Type::Point, const unsigned int, const unsigned int, omprng &)=0

  Assigns the datum to the subclusters of cluster k, after sampling the subclusters' label with the Sampling method.
- virtual void UpdateZeta (const unsigned int)=0

  Method needed to remove cluster k during M-H moves.
- virtual void UpdateZeta (const unsigned int, const unsigned int)=0

  Method needed to remove two clusters during M-H moves.

- virtual unsigned int ViewNj () const =0

  Retrieve number of data in group j.
- virtual void ViewData (vector< typename Type::Point > &) const =0

  Retrieve data id in group j.
- virtual void ViewCounts4c (const unsigned int, typename Type::STAT &, typename Type::STAT &)=0

  Retrieve counts necessary for updating subclusters' latent parameters.
- virtual unsigned int ViewNumTableID (const unsigned int) const =0

  Retrieve number of tables for cluster k m_{jk}.
- virtual unsigned int ViewNumTableLeftID (const unsigned int) const =0

  Retrieve number of tables for left subcluster of cluster k m_{jkl}.
- virtual unsigned int ViewNumTableRightID (const unsigned int) const =0

  Retrieve number of tables for right subcluster of cluster k m_{jkr}.
- virtual unsigned int ViewDataCountID (const unsigned int) const =0

  Retrieve number of data in cluster k $n_{\cdot k}$.
- virtual unsigned int ViewDataCountLeftID (const unsigned int) const =0

  Retrieve number of data in left subcluster of cluster k $n_{\cdot kl}$.
- virtual unsigned int ViewDataCountRightID (const unsigned int) const =0

  Retrieve number of data in right subcluster of cluster k $n_{\cdot kr}$.
- virtual void ResetDataCountSub (const unsigned int)=0

  Reset data counts in cluster k.
- virtual void ViewDataCount (vector< unsigned int > &) const =0

  Retrieve a vector containing the number of data in all clusters.
- virtual void ViewDataCountLeft (vector< unsigned int > &) const =0

  Retrieve a vector containing the number of data in all left subclusters.
- virtual void ViewDataCountRight (vector< unsigned int > &) const =0

  Retrieve a vector containing the number of data in all right subclusters.
- virtual void ViewIdCounts (vector< pair< typename Type::Point, unsigned int >> &, const unsigned int)=0

  Retrieve counts and id of data in cluster k.
- virtual void ViewCluster (const unsigned int, pair< unordered_map< typename Type::Point, unsigned int >, unordered_map< typename Type::Point, unsigned int >> &)=0

  Retrieve cluster k.
- virtual double ViewPiID (const unsigned int) const =0

  Retrieve group specific weight of cluster k π_{jk}.
- virtual double ViewPiLeftID (const unsigned int) const =0

  Retrieve group specific weight for the left subcluster of cluster k π_{jkl}.
- virtual double ViewPiRightID (const unsigned int) const =0

  Retrieve group specific weight for the right subcluster of cluster k π_{jkr}.
- virtual void ViewPi (vector< double > &) const =0

  Retrieve a vector containing the group specific cluster weights.
- virtual void ViewPiLeft (vector< double > &) const =0

  Retrieve a vector containing the group specific weights for left subclusters.
- virtual void ViewPiRight (vector< double > &) const =0

  Retrieve a vector containing the group specific weights for right subclusters.
- virtual void SetAlpha (const double)=0

  Set the concentration parameter α of the Dirichlet process governing the group.
- virtual void SetNj (const unsigned int)=0

  Set the number of data in group j.
- virtual void SetPi (vector< double > &)=0

  Set the vector containing the group specific cluster weights.

- virtual void InsertNewCluster (const pair< unordered_map< typename Type::Point, unsigned int >, unordered_map< typename Type::Point, unsigned int > > &, const double, const double, const double, const unsigned int, const unsigned int, const unsigned int, const unsigned int, const unsigned int, const unsigned int)=0

  Insert a new cluster.
- virtual void RemoveCluster (const vector< unsigned int > &)=0

  Remove clusters with id contained in the input vector.
- virtual void RemoveCluster (const unsigned int)=0

  Remove cluster k.
- virtual void RemoveCluster (unsigned int, unsigned int)=0

  Remove two clusters.
- virtual unsigned int CheckLeftSubcluster (const unsigned int)=0

  Verifies if cluster k has an empty left subcluster.
- virtual unsigned int CheckRightSubcluster (const unsigned int)=0

  Verifies if cluster k has an empty right subcluster.
- virtual void ViewLabel (vector< pair< typename Type::Point, unsigned int >> &)=0

  Retrieve labels assigned to data.
- virtual void SetDataset (std::istringstream &)=0

  Acquires data.
- virtual unsigned int SortData (unsigned int, omprng &)=0

  Allocates data in the Zeta container.

Private Member Functions

- virtual void UpdateDataCount ()=0

  Updates data counts in clusters.
- virtual void Sampling (std::vector< unsigned int > &, std::vector< double > &, unsigned int, omprng &)=0

  Samples cluster's or subcluster's label from categorical distribution.

5.10.1 Detailed Description

template<typename Type, unsigned int DIM>class GenericDocument< Type, DIM >

Generic class for groups of data.

Abstract class in which all methods are virtual. It contains methods that must be defined in all derived classes.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.10.2 Member Function Documentation

5.10.2.1 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::CheckLeftSubcluster (const unsigned *int*) [pure virtual]

Verifies if cluster k has an empty left subcluster.

Parameters

_k	- cluster id

5.10.2.2 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::CheckRightSubcluster (const unsigned *int*) [pure virtual]

Verifies if cluster k has an empty right subcluster.

Parameters

_k	- cluster id

5.10.2.3 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::InsertNewCluster (const pair< unordered_map< typename Type::Point, unsigned int >, unordered_map< typename Type::Point, unsigned int > > & , const double , const double , const double , const unsigned *int*, const unsigned *int*, const unsigned *int*, const unsigned *int*, const unsigned *int*) [pure virtual]

Insert a new cluster.

Parameters

NewCluster	- the new cluster
_Pi	- group specific weight for the new cluster
_PiLeft	- group specific left subcluster weight for the new cluster
_PiRight	- group specific right subcluster weight for the new cluster
_WordCount	- number of data in the new cluster
_WordCountLeft	- number of data in the left subcluster of the new cluster
_WordCount-Right	- number of data in the right subcluster of the new cluster
_LocalTable	- number of tables serving the new dish in restaurant j (CRF metaphor) m_{jk}
_LocalTableLeft	- number of tables serving the new left dish in restaurant j (CRF metaphor) m_{jkl}
_LocalTable-Right	- number of tables serving the new right dish in restaurant j (CRF metaphor) m_{jkr}

5.10.2.4 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::RemoveCluster (const vector< unsigned int > &) [pure virtual]

Remove clusters with id contained in the input vector.

Parameters

_k	- vector containing the id of clusters to eliminate

Implemented in CategoricalDocument< DIM >.

5.10.2.5 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::RemoveCluster (const unsigned *int*) [pure virtual]

Remove cluster k.

Parameters

_k	- cluster id

5.10.2.6 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::RemoveCluster (unsigned *int,* unsigned *int*) [pure virtual]

Remove two clusters.

Parameters

_k1	- cluster id
_k2	- cluster id

5.10.2.7 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ResetDataCountSub (const unsigned *int*) [pure virtual]

Reset data counts in cluster k.

Parameters

k	- cluster id

5.10.2.8 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::Sampling (std::vector< unsigned int > & , std::vector< double > & , unsigned *int,* omprng &) [private], [pure virtual]

Samples cluster's or subcluster's label from categorical distribution.

Parameters

_temp_counts	- vector containing count for datum id in all clusters
_Weights	- weights for sampling labels
_nidj	- number of times datum id appears in group j
Gen	- parallel random number generator

Implemented in CategoricalDocument< DIM >.

5.10.2.9 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::SetAlpha (const double) [pure virtual]

Set the concentration parameter α of the Dirichlet process governing the group.

Parameters

_alpha	- α

Implemented in CategoricalDocument< DIM >.

5.10.2.10 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::SetDataset (std::istringstream &) [pure virtual]

Acquires data.

Parameters

SSTR	- contains data id and counts in the group

Implemented in CategoricalDocument< DIM >.

5.10.2.11 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::SetNj (const unsigned *int*) [pure virtual]

Set the number of data in group j.

Parameters

_Nj	- number of data in group j

Implemented in CategoricalDocument< DIM >.

5.10.2.12 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::SetPi (vector< double > &) [pure virtual]

Set the vector containing the group specific cluster weights.

Parameters

_pi	- vector containing the group specific cluster weights

Implemented in CategoricalDocument< DIM >.

5.10.2.13 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::SortData (unsigned *int*, ompprng &) [pure virtual]

Allocates data in the Zeta container.

Parameters

_K	- initial number of clusters
Gen	- parallel random number generator

Implemented in CategoricalDocument< DIM >.

5.10.2.14 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateAllLocalTableSub (const vector< long double > & , const vector< double > & , const vector< double > & , ompprng &) [pure virtual]

Updates table counts for subclusters of all clusters.

Parameters

_stirling	- Stirling numbers
_BetaLeft	- global weight for left subclusters of all cluster
_BetaRight	- global weight for right subclusters of all cluster
Gen	- parallel random number generator

Implemented in CategoricalDocument< DIM >.

5.10.2.15 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateAllPiSub (const vector< double > , const vector< double > , ompprng &) [pure virtual]

Updates group specific subclusters' weights for all cluster.

Parameters

_BetaLeft	- global weights for left subclusters
_BetaRight	- global weights for right subclusters
Gen	- parallel random number generator

Implemented in CategoricalDocument< DIM >.

5.10.2.16 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateLocalTable (const vector< long double > & , const vector< double > & , omprng &) [pure virtual]

Updates table counts.

Parameters

_stirling	- Stirling numbers
_Beta	- clusters' global weights
Gen	- parallel random number generator

Implemented in CategoricalDocument< DIM >.

5.10.2.17 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateLocalTableSub_OneCluster (const vector< long double > & , const double _, const double , const unsigned *int,* omprng &) [pure virtual]

Updates table counts for subclusters of cluster k.

Parameters

_stirling	- Stirling numbers
_BetaLeft	- global weight for left subcluster of cluster k
_BetaRight	- global weight for right subcluster of cluster k
k	- cluster id
Gen	- parallel random number generator

5.10.2.18 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdatePi (const vector< double > & , omprng &) [pure virtual]

Updates group specific clusters' weights.

Parameters

_AllBeta	- clusters' global weights
Gen	- parallel random number generator

Implemented in CategoricalDocument< DIM >.

5.10.2.19 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdatePiSub (const double , const double , const unsigned *int,* omprng &) [pure virtual]

Updates group specific weights for subclusters of cluster k.

Parameters

_BetaLeft	- global weight for left subcluster of cluster k
_BetaRight	- global weight for right subcluster of cluster k
k	- cluster id
Gen	- parallel random number generator

5.10.2.20 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateZeta (const typename Type::THETA & , const unsigned *int,* omprng &) [pure virtual]

Updates cluster label of a datum, sampled with the Sampling method.

Parameters

_ThetaId	- vector of a datum's weights in all clusters
_VettId	- datum id
Gen	- parallel random number generator

5.10.2.21 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateZeta (const unsigned *int*) [pure virtual]

Method needed to remove cluster k during M-H moves.

Parameters

_k	- cluster id

5.10.2.22 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateZeta (const unsigned *int,* const unsigned *int*) [pure virtual]

Method needed to remove two clusters during M-H moves.

Parameters

_k1	- cluster id
_k2	- cluster id

5.10.2.23 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateZeta_and_Sub (const typename Type::THETA & , const typename Type::THETA & , const typename Type::THETA & , const unsigned *int,* omprng &) [pure virtual]

Updates cluster label of a datum, sampled with with the Sampling method.

Parameters

_ThetaId	- vector of a datum's weights in all clusters
_ThetaIdLeft	- vector of a datum's weights in all left subclusters
_ThetaIdRight	- vector of a datum's weights in all right subclusters
_VettId	- datum id
Gen	- parallel random number generator

5.10.2.24 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::UpdateZetaSub (const typename Type::THETA & , const typename Type::THETA & , const typename Type::Point , const unsigned *int,* const unsigned *int,* omprng &) [pure virtual]

Assigns the datum to the subclusters of cluster k, after sampling the subclusters' label with the Sampling method.

Parameters

_ThetaIdLeft	- vector of a datum's weights in all left subclusters
_ThetaIdRight	- vector of a datum's weights in all right subclusters
id	- datum id
nidjk	- number of times datum id in group j is assigned to cluster k
k	- cluster id
Gen	- parallel random number generator

5.10.2.25 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewCluster (const unsigned *int,* pair< unordered_map< typename Type::Point, unsigned int >, unordered_map< typename Type::Point, unsigned int >> &) [pure virtual]

Retrieve cluster k.

Parameters

_k	- cluster id
_Cluster	- structure that will contain cluster k

5.10.2.26 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewCounts4c (const unsigned *int,* typename Type::STAT & , typename Type::STAT &) [pure virtual]

Retrieve counts necessary for updating subclusters' latent parameters.

Parameters

_k	- cluster id
_counts4cleft	- counts for left subcluster's latent parameter
_counts4cright	- counts for right subcluster's latent parameter

5.10.2.27 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewData (vector< typename Type::Point > &) const [pure virtual]

Retrieve data id in group j.

Parameters

_VettId	- filled with data id in group j

5.10.2.28 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewDataCount (vector< unsigned int > &) const [pure virtual]

Retrieve a vector containing the number of data in all clusters.

Parameters

_WordCount	- vector containing the number of data in all clusters

Implemented in CategoricalDocument< DIM >.

5.10.2.29 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::ViewDataCountID (const unsigned *int*) const [pure virtual]

Retrieve number of data in cluster k $n_{\cdot k}$.

Parameters

_k	- cluster k

Returns

$n_{\cdot k}$

5.10.2.30 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewDataCountLeft (vector< unsigned int > &) const [pure virtual]

Retrieve a vector containing the number of data in all left subclusters.

Parameters

_WordCountLeft	- vector containing the number of data in all left subclusters

Implemented in CategoricalDocument< DIM >.

5.10.2.31 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::ViewDataCountLeftID (const unsigned *int*) const [pure virtual]

Retrieve number of data in left subcluster of cluster k $n_{\cdot kl}$.

Parameters

_k	- cluster id

Returns

$n_{\cdot kl}$

5.10.2.32 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewDataCountRight (vector< unsigned int > &) const [pure virtual]

Retrieve a vector containing the number of data in all right subclusters.

Parameters

_WordCount-Right	- vector containing the number of data in all right subclusters

Implemented in CategoricalDocument< DIM >.

5.10.2.33 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::ViewDataCountRightID (const unsigned *int*) const [pure virtual]

Retrieve number of data in right subcluster of cluster k $n_{\cdot kr}$.

Parameters

_k	- cluster id

Returns

$n_{\cdot kr}$

5.10.2.34 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewIdCounts (vector< pair< typename Type::Point, unsigned int >> & , const unsigned *int*) [pure virtual]

Retrieve counts and id of data in cluster k.

Parameters

_nidjk	- structure that will contain counts and id of data in cluster k
_k	- cluster id

5.10.2.35 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewLabel (vector< pair< typename Type::Point, unsigned int >> &) [pure virtual]

Retrieve labels assigned to data.

Parameters

Data	- structure to store retrieved labels

5.10.2.36 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::ViewNj ( ) const [pure virtual]

Retrieve number of data in group j.

Returns

Number of data in group j

Implemented in CategoricalDocument< DIM >.

5.10.2.37 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::ViewNumTableID (const unsigned *int*) const [pure virtual]

Retrieve number of tables for cluster k m_{jk}.

Parameters

_k	- cluster id

Returns

m_{jk}

5.10.2.38 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::ViewNumTableLeftID (const unsigned *int*) const [pure virtual]

Retrieve number of tables for left subcluster of cluster k m_{jkl}.

Parameters

_k	- cluster id

Returns

m_{jk}

5.10.2.39 template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::ViewNumTableRightID (const unsigned *int*) const [pure virtual]

Retrieve number of tables for right subcluster of cluster k m_{jkr}.

Parameters

_k	- cluster id

Returns

m_{jkr}

5.10.2.40 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewPi (vector< double > &) const [pure virtual]

Retrieve a vector containing the group specific cluster weights.

Parameters

_pi	- vector containing the group specific cluster weights

Implemented in CategoricalDocument< DIM >.

5.10.2.41 template<typename Type, unsigned int DIM> virtual double GenericDocument< Type, DIM >::ViewPiID (const unsigned *int*) const [pure virtual]

Retrieve group specific weight of cluster k π_{jk}.

Parameters

_k	- cluster id

Returns

π_{jk}

5.10.2.42 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewPiLeft (vector< double > &) const [pure virtual]

Retrieve a vector containing the group specific weights for left subclusters.

Parameters

_pi_left	- vector containing the group specific weights for left subclusters

Implemented in CategoricalDocument< DIM >.

5.10.2.43 template<typename Type, unsigned int DIM> virtual double GenericDocument< Type, DIM >::ViewPiLeftID (const unsigned *int*) const [pure virtual]

Retrieve group specific weight for the left subcluster of cluster k π_{jkl}.

Parameters

_k	- cluster id

Returns

π_{jkl}

5.10.2.44 template<typename Type, unsigned int DIM> virtual void GenericDocument< Type, DIM >::ViewPiRight (vector< double > &) const [pure virtual]

Retrieve a vector containing the group specific weights for right subclusters.

Parameters

_pi_right	- vector containing the group specific weights for right subclusters

Implemented in CategoricalDocument< DIM >.

5.10.2.45 template<typename Type, unsigned int DIM> virtual double GenericDocument< Type, DIM >::ViewPiRightID (const unsigned *int*) const [pure virtual]

Retrieve group specific weight for the right subcluster of cluster k π_{jkr}.

Parameters

_k	- cluster id

Returns

π_{jkr}

The documentation for this class was generated from the following file:

- Document.hpp

5.11 GenericPosteriorAnalysis< Type, DIM > Class Template Reference

Generic class for the posterior analysis. Virtual class where all methods are null. Each inherited class must define all methods in the base class and, if necessary, add other methods. Calls R scripts. Computes LPML index, identifies the topics and detect the best clustering of data according to the least square criteria.

`#include <PosteriorAnalysis.hpp>`

Public Member Functions

- virtual void SetAllK ()=0

  Sets the MCMC chain of the number of clusters K.
- virtual void SetAllAlpha ()=0

  Sets the MCMC chain of the values for concentration parameter α.
- virtual void SetAllGamma ()=0

  Sets the MCMC chain of the values for concentration parameter γ.
- virtual void KPosteriorAnalysis (RInside &, const unsigned int, const unsigned int)=0

  Calls the R script for the analysis of the K chain.
- virtual void AGPosteriorAnalysis (RInside &, const unsigned int, const unsigned int, const unsigned int, const unsigned int, const char, const char)=0

  Calls the R script for the analyis of Alpha and Gamma chains.
- virtual void Setwd (const std::string &)=0

  Sets the R working directory.
- virtual void SetW (const unsigned int)=0

  Sets the dimension of the hyperparameter fo the laten parameter's distribution.
- virtual void SetD (const unsigned int)=0

  Sets the number of groups.
- virtual void SetN (const unsigned int)=0

  Sets the total number of data.
- virtual void SetIterations (const unsigned int)=0

  Sets the number of iterations for the MCMC chain.
- virtual void Setburnin (const unsigned int)=0

  Sets the number of iterations to discard in order to compute the LPML.
- virtual void SetBT (const unsigned int, const unsigned)=0

  Sets burnin and thinning for the K chain.
- virtual void LeastSquareClustering ()=0

  Finds the best clustering according to the least squares criteria.
- virtual void LPML ()=0

  Computes and prints on the terminal the LPML index.
- virtual void UnioneLabels (RInside &R)=0

Skims the Labels∗.bin files and joins them in the Labels.bin file.

- virtual void WriteBestParams ()=0

  Writes on binary files θ and β parameters of the optimal iteration.

- virtual void SetDocs ()=0

  Acquires documents' names.

- virtual void AssociatingDocs ()=0

  Associates documents to estimated topics.

5.11.1 Detailed Description

template<typename Type, unsigned int DIM>class GenericPosteriorAnalysis< Type, DIM >

Generic class for the posterior analysis. Virtual class where all methods are null. Each inherited class must define all methods in the base class and, if necessary, add other methods. Calls R scripts. Computes LPML index, identifies the topics and detect the best clustering of data according to the least square criteria.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.11.2 Member Function Documentation

5.11.2.1 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::AGPosteriorAnalysis (RInside & , const unsigned *int*, const unsigned *int*, const unsigned *int*, const unsigned *int*, const char , const char) [pure virtual]

Calls the R script for the analyis of Alpha and Gamma chains.

Parameters

R	- R istance
AlphaBurnin	- number of initial values in the Alpha chain to be discarded
AlphaThinning	- keep a value in the Alpha chain every AlphaThinning values
GammaBurnin	- number of initial values in the Gamma chain to be discarded
GammaThinning	- keep a value in the Gamma chain every GammaThinning values
AlphaTry	- yes if you want to repeat the Alpha chain's analysis, no otherwise
GammaTry	- yes if you want to repeat the Gamma chain's analysis, no otherwise

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.2 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::KPosteriorAnalysis (RInside & , const unsigned *int*, const unsigned *int*) [pure virtual]

Calls the R script for the analysis of the K chain.

Parameters

R	- R istance

Burnin	- number of initial values in the chain to be discarded
Thinning	- keep a value in the chain every thinning values

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.3 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::SetBT (const unsigned *int*, const unsigned) [pure virtual]

Sets burnin and thinning for the K chain.

Parameters

_Burnin	- burnin
_Thinning	- thinning

5.11.2.4 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::Setburnin (const unsigned *int*) [pure virtual]

Sets the number of iterations to discard in order to compute the LPML.

Parameters

_burnin	- iterations to discard in order to compute the LPML

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.5 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::SetD (const unsigned *int*) [pure virtual]

Sets the number of groups.

Parameters

_D	- number of groups

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.6 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::SetIterations (const unsigned *int*) [pure virtual]

Sets the number of iterations for the MCMC chain.

Parameters

_Iterations	- number of iterations

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.7 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::SetN (const unsigned *int*) [pure virtual]

Sets the total number of data.

Parameters

_N	- total number of data

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.8 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::SetW (const unsigned *int*) [pure virtual]

Sets the dimension of the hyperparameter fo the laten parameter's distribution.

Parameters

_W	- dimension of the hyperparameter fo the laten parameter's distribution

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.9 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::Setwd (const std::string &) [pure virtual]

Sets the R working directory.

Parameters

_wd	- R working directory

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.10 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::UnioneLabels (RInside & *R*) [pure virtual]

Skims the Labels$*$.bin files and joins them in the Labels.bin file.

Parameters

R	- object of class RInside

Implemented in CategoricalPosteriorAnalysis< DIM >.

5.11.2.11 template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::WriteBestParams ( ) [pure virtual]

Writes on binary files θ and β parameters of the optimal iteration.

Parameters

file_nr	- files in which looking for parameters
move	- in those files, number of initial iterations to skip

Returns

number of clusters in the best clustering

Implemented in CategoricalPosteriorAnalysis< DIM >.

The documentation for this class was generated from the following file:

- PosteriorAnalysis.hpp

5.12 greater_for_pair Struct Reference

Sorting operator for <unsigned int,double> pairs. The order is based on the second element in the pair.

`#include <Functions.hpp>`

Public Member Functions

- bool **operator()** (const std::pair< unsigned int, double > &x, const std::pair< unsigned int, double > &y) const
- bool **operator()** (const std::pair< unsigned int, double > &x, const std::pair< unsigned int, double > &y) const

5.12.1 Detailed Description

Sorting operator for <unsigned int,double> pairs. The order is based on the second element in the pair.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

The documentation for this struct was generated from the following files:

- Functions.hpp
- PosteriorAnalysis.hpp

5.13 HDP_MCMC< MODEL, DOCUMENT, DIM > Class Template Reference

HDP_MCMC.

`#include <HDP_MCMC.hpp>`

Public Types

- using **THETA** = typename MODEL< DIM >::THETA
- using **POINT** = typename MODEL< DIM >::POINT
- using **HYP** = typename MODEL< DIM >::HYP
- using **Corpus** = vector< DOCUMENT< DIM >>
- using **ClusterID** = unsigned int

Public Member Functions

- HDP_MCMC ()

  Default constructor: all attribute are inizialized with their default constructor. Scalar variable are equal to zero All Flags are inizialized as FALSE All object are inizialize as empty The number of threads is inizialized from command line.
- ~HDP_MCMC ()=default

  Default distructor.
- void SetK_init (unsigned int _K)

Set the initial number of cluster, It is decide from users.

- void SetDataset (const std::string &Dataset, const std::string &MainVariable)

  It acquire dataset and its dimentions form file It uses Corpus' method which generate structure that manage dataset.
- void SetAlphaFixed (double _Alpha)

  Set the fixed vale of Alpha. It is decided by users.
- void SetAlphaPrior (double _AA, double _AB)

  It sets a prior on Alpha.
- void SetGammaFixed (double _Gamma)

  Set the fixed vale of Gamma. It is decided by users.
- void SetGammaPrior (double _GA, double _GB)

  It sets a prior on Gamma.
- void SetLambdaInfo (HYP Lambda)

  It sets the value of lambda.
- void SetSeed (const unsigned long Seed)

  It sets the seed. It is the same during the execution of algorithm.
- void Check_Model (unsigned long burnin)

  It set the check of model with LPML.
- unsigned int ViewW ()

  It pull out the latent paramenter's dimention.
- unsigned int ViewK ()

  it pull out the current number of clusters
- unsigned int ViewD ()

  It pull out the total number of groups.
- unsigned int ViewN ()

  It pull out the dataset dimension.
- void Algorithm (unsigned int Iterations, unsigned int Iterations_Sub)

  Algorithm.

Private Member Functions

- template<class T >
  void Swap (T &Old, T &New)

  It swaps places the old proposes with new proposes When the method finishes, the new poroposes which weren't accepted during the M-H step, are destroyed. It is a template method because it is used with Corpus and with Model.
- void Summary ()

  It print out a summary about the set of algorithm.
- void SetClusters ()

  It inizialize the structur which manage clusters.
- void UpdateClusterCounts ()

  Update the hyperparameters of latent parameters, on the base of data distribution in each clusters.
- void UpdateTable ()

  Update m_{jk} in each group. After $m_{.k} = \sum_j m_{jk}$, it possible to update the table in each clusters.
- void UpdateSubTable ()

  Update table in each sub-clusters of clusters, m_{jkl}, m_{jkr} in each group. After $m_{.kh} = \sum_j m_{jkh}$ t possible to update the table sub-clusters.
- void UpdateDocWeights ()

  Update clusters'weight in each group.
- void UpdateDocWeights_Sub ()

  Update all sub-clusters'weight in each group.
- void UpdateAssignment_Cluster_and_Subcluster ()

  It assigns the new lables of cluster and sub-clusters to each data.

- void UpdateAssignment_Cluster ()

  It assigns the new lables of cluster to each data It is used in global Merge/split moves to propose the new lables.
- void UpdateBeta ()

  It update clusters' global weight.
- void UpdateAllBetaSub ()

  It update sub-clusters' global weight.
- void UpdateBetaSub (const ClusterID k)

  It update global weight of cluster and its sub-clusters which is identified by its ID.
- void EmptyCluster ()

  Check which clusters are empty and erase them.
- bool IsEmptySubcluster (const ClusterID _k)

  Chek if one of two sub-clusters is empty.
- void computeLogL ()

  Compute a part of Hastings ratio.
- long double logq ()

  Compute a part of Hastings ratio.
- void Gibbs_SubCluster (const vector< ClusterID > &ProposedClusters)

  Gibbs sampler steps to sample new propose for sub-topics of new clusters.
- void LocalSplit ()

  Local split move. It is proposed the split of each cluster in two new clusters. This propose is the same in all group. If the new propose is accept, there is the sampler of sub-clusters of each new clusters.
- void LocalMerge ()

  Local merge move. The pair of clusters to merge and propose new cluster are sampled. If the new propose is accept, there is the sampler of sub-clusters of each new clusters.
- void GlobalMerge ()

  Global merge move One pair of clusters is proposed in a casual way. It is proposed the merge of this pair to realise a new cluster. In this step the new quantity of all variable of all clusters are sampled. If the new propose is accept, there is the sampler of sub-clusters of each new clusters.
- void GlobalSplit ()

  Mosse di split globale. Two clusters are choosen in a casual way. It is proposed the union of these two clusters in order to create a new clustr. In this step the new quantity of all variable of all clusters are sampled. If the new propose is accept, there is the sampler of sub-clusters of each new clusters.
- void AlphaPrior ()

  It Samples new ALpha.
- void GammaPrior ()

  It Samples new Gamma.
- void UpdateK ()

  Update of K, which contains information about current number of clusters.
- void UpdateAllK ()

  Uodate of history of K. It is done at the end of each iteration.
- void SaveAllK ()

  It saves on file the chain of K.
- void SaveAllAlpha ()

  It saves on file the chain of Alpha.
- void SaveAllGamma ()

  It saves on file the chain of Gamma.
- void SaveRunTime ()

  It saves runtime all chain.
- void SaveLastBeta (const std::string &Filename)

  It saves on file clusters' global weight.
- void SaveLastPi (const std::string &Filename)

  It saves on file clusters' weight of each group.

- void SaveLastTheta (const std::string &Filename)

  It saves on file clusters' latent parameters.
- void LPML ()

  It saves on file the necessary quantity to calculate LPML.
- void SaveLabels (const std::string &Filename)

  It save on file data' lables.

Private Attributes

- Corpus corpus

  Object that manage data.
- std::tuple< bool, bool, bool, bool, bool, bool, bool, bool, bool > Flags

  Flags that provide the necessary information to perform algorithm with the required characteristics from users First element is true when dataset has been already loaded. Second element is true if the information about Alpha has been already entered. Third element is ture if the user wants to put a prior on Alpha. Third element is è false if the user wants to perform algorithm with a fixed Alpha Fourth element is true if the information about Gamma has already been enetered. Fivth element is true if user wants to put a prior on Gamma. Fivth element is false if unser wants to perform algorithm with fixed Gamma Sixth element is true if the informtation about Lambda has already been entered. Seventh element is true if the initial number of cluster is already eneterd Eighth element is true if uses has chose the seed Ninth element is true if user wants to monitor the model with LPML.
- unsigned int D

  Number of total group.
- unsigned int N

  NNumber of total data.
- unsigned int W

  Dimension of hyperparameter of latent paramenter's distribution.
- unsigned long It

  Current iteration.
- unsigned long MaxIt

  Maximun number of iteration (criterion of stop)
- unsigned long MaxIt_SubCluster

  Maximun number of iteration to sample sub-cluaters (criterion of stop)
- double Gamma

  Concentration parameter of dirichlet process which describes the global topics (at iterations It)
- vector< double > AllGamma

  History of Gamma when prior was setted.
- double GA

  Shape parameter of Gamma prior.
- double GB

  Rate parameter on Gamma prior.
- double Alpha

  Concentration parameter of dirichlet process which describe the topic in each group of data (at interation It)
- vector< double > AllAlpha

  History of Alpha when prior was setted.
- double AA

  Shape parameter of Alpha prior.
- double AB

  Rate parameter on Alpha prior.
- unsigned int K

  Current number of clusters.

- vector< unsigned int > AllK

  History of all K.
- vector< long double > LogStirlingNumbers

  Logaritmic Stirling Number.
- MODEL< DIM > Model

  Model for the initial distribution .
- double Beta_empty

  Empty cluster's global weight at It iterataion. This joins all weight of cluster (in this moment empty), which could appear in the next iterations.
- vector< double > logL

  Matrix to calculate the Hasting Ratio.
- unsigned int m

  Total number of table which are used to sample Gamma.
- vector< double > CPO

  Object that memorizes $\sum_{It=1}^{MaxIt} f_{ij}(y_{ij}|\theta_{z_{ij}}^{(g)})$.
- unsigned long burnin_CPO

  Iteration when user wants star to calculate CPO.
- omprng Gen

  Parallel random number generator.
- unsigned int OMP_NUM_THREADS

5.13.1 Detailed Description

template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM = 1>class HDP_MCMC< MODEL, DOCUMENT, DIM >

HDP_MCMC.

Implementation of algorithm. Gibbs Sampler on clusters and i sub-cluster, Metropolis-Hastings to propose Merge/-Split local amd global moves. It is implemented the sample equation to update global weigth and table. In this class there are method that update the cluster's count in Model, in particular check the cluster situation. This class use the class implement in Cluster.hpp to manage cluster and to define the likelihood to use, classes in Model.hpp to do the inference on latent paramenter and to define the prior. Moreover this class use the classed define in Document.-hpp to manage data and to sample their lables. In particular Model is the class that define the latent paramenter MODEL. In Model.hpp is implemented CategoricalModel (prior Dirichlet)

Authors

{Debora Parisi and Stefania Perego}

Date

Febbrario 2016

5.13.2 Member Function Documentation

5.13.2.1 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::Algorithm (unsigned int *Iterations,* unsigned int *Iterations_Sub*)

Algorithm.

Parameters

Iterations	- Maximun nuber of iterations
Iterations_Sub	- Maximun nuber of iterations to gibbs samplere which sample sub-topics

5.13.2.2 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::Check_Model (unsigned long *burnin*)

It set the check of model with LPML.

Parameters

burnin	- When start the cacluclation on LPML

5.13.2.3 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::Gibbs_SubCluster (const vector< ClusterID > & *ProposedClusters*) [private]

Gibbs sampler steps to sample new propose for sub-topics of new clusters.

Parameters

Proposed-Clusters	- Labels of new clusters

5.13.2.4 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> bool HDP_MCMC< MODEL, DOCUMENT, DIM >::IsEmptySubcluster (const ClusterID *_k*) [private]

Chek if one of two sub-clusters is empty.

Parameters

_k	- Cluster's ID to check.

Returns

TRUE if one of two sub-cluster is empty.

5.13.2.5 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::SetAlphaFixed (double *_Alpha*)

Set the fixed vale of Alpha. It is decided by users.

Parameters

_Alpha	- Value assigned to alpha

5.13.2.6 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::SetAlphaPrior (double *_AA*, double *_AB*)

It sets a prior on Alpha.

Parameters

_AA	- Shape parameter, it is decided by users
_AB	- Rate parameter it is decided by users

5.13.2.7 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::SetDataset (const std::string & *Dataset,* const std::string & *MainVariable*)

It acquire dataset and its dimentions form file It uses Corpus' method which generate structure that manage dataset.

Parameters

Dataset	- File name of Dataset
MainVariable	- File name which contain information about dimenstion of dataset

5.13.2.8 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::SetGammaFixed (double *_Gamma*)

Set the fixed vale of Gamma. It is decided by users.

Parameters

_Gamma	- Value assigned to gamma

5.13.2.9 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::SetGammaPrior (double *_GA,* double *_GB*)

It sets a prior on Gamma.

Parameters

_AA	- Shape parameter, it is decided by users
_AB	- Rate parameter it is decided by users

5.13.2.10 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::SetK_init (unsigned int *_K*)

Set the initial number of cluster, It is decide from users.

Parameters

_K	- Initial cluster

5.13.2.11 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::SetLambdaInfo (HYP *Lambda*)

It sets the value of lambda.

Parameters

Lambda	- Value decideb by users

5.13.2.12 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::SetSeed (const unsigned long *Seed*)

It sets the seed. It is the same during the execution of algorithm.

Parameters

Seed	- seed

5.13.2.13 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> template<class T > void HDP_MCMC< MODEL, DOCUMENT, DIM >::Swap (T & *Old,* T & *New*) [private]

It swaps places the old proposes with new proposes When the method finishes, the new poroposes which weren't accepted during the M-H step, are destroyed. It is a template method because it is used with Corpus and with Model.

Parameters

Old	- Before the exchange
New	- New proposes

5.13.2.14 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> void HDP_MCMC< MODEL, DOCUMENT, DIM >::UpdateBetaSub (const ClusterID *k*) [private]

It update global weight of cluster and its sub-clusters which is identified by its ID.

Parameters

k	- Cluster's ID

5.13.2.15 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> unsigned int HDP_MCMC< MODEL, DOCUMENT, DIM >::ViewD ( )

It pull out the total number of groups.

Returns

Total number of groups

5.13.2.16 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> unsigned int HDP_MCMC< MODEL, DOCUMENT, DIM >::ViewK ( )

it pull out the current number of clusters

Returns

Number of clusters

5.13.2.17 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> unsigned int HDP_MCMC< MODEL, DOCUMENT, DIM >::ViewN ( )

It pull out the dataset dimension.

Returns

Dataset dimention

5.13.2.18 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM> unsigned int HDP_MCMC< MODEL, DOCUMENT, DIM >::ViewW ( )

It pull out the latent paramenter's dimention.

Returns

latent paramenter's dimention

5.13.3 Member Data Documentation

5.13.3.1 template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM = 1> unsigned int HDP_MCMC< MODEL, DOCUMENT, DIM >::OMP_NUM_THREADS [private]

Number of threads

The documentation for this class was generated from the following file:

- HDP_MCMC.hpp

5.14 ModelGeneric< Type, DIM > Class Template Reference

Interface for the Model class Abstract class where all methods are virtual. Classes that inherit from ModelGeneric sample latent parameters and manages related hyperparameters, which are specific to the chosen model. Manages model specific functions, such as likelihood, marginals and other densities. Removes and adds clusters.

`#include <Model.hpp>`

Public Member Functions

- virtual unsigned int ViewK () const =0

  Retrieves current number of clusters.
- virtual void ViewKey (vector< unsigned int > &) const =0

  Retrieves current clusters' ids.
- virtual void SetHyperparameter (const typename Type::HYP &)=0

  Fixes hyperparameters for the latent parameters' distribution.
- virtual void DefaultHyperparameter (size_t)=0

  Fixes hyperparameters for the latent parameters' distribution with default values.
- virtual void SetInitialClusters (unsigned int)=0

  Sets clusters assigning initial values for their weights.
- virtual double Marginalized_Loglikelihood (const unsigned int)=0

  Computes the logarithm of the cluster's marginal likelihood, given the cluster id.
- virtual double Loglikelihood (const typename Type::Point, const unsigned int)=0

  Computes the loglikelihood of datum X, given the cluster id.
- virtual double LoglikelihoodLeft (const typename Type::Point, const unsigned int)=0

  Computes the loglikelihood of datum X, given the left subcluster id.
- virtual double LoglikelihoodRight (const typename Type::Point, const unsigned int)=0

  Computes the loglikelihood of datum X, given the right subcluster id.
- virtual long double LogDensity (const unsigned int)=0

  Computes the latent parameter's density, given the cluster id.
- virtual void UpdateThetaCluster (omprng &)=0

  Updates latent parameters of all clusters.
- virtual void UpdateThetaSubCluster (omprng &)=0

Updates latent parameters of left and right subclusters of all clusters.

- virtual void UpdateOneThetaCluster (const unsigned int, omprng &)=0

  Updates latent parameters of one cluster, given its id.
- virtual void UpdateOneThetaSubCluster (const unsigned int, omprng &)=0

  Updates latent parameters of left and right subclusters of one cluster, given its id.
- virtual void AddOneCluster (const unsigned int)=0

  Adds an empty cluster to the current clusters and updates the current number of clusters.
- virtual void RemoveOneCluster (const unsigned int)=0

  Removes a cluster given its id and updates the current number of clusters.
- virtual void RemoveClusters (const vector< unsigned int > &)=0

  Removes multiple clusters given their id and updates the current number of clusters.
- virtual void ViewBeta (vector< double > &)=0

  Retrieves current clusters' global weights.
- virtual void ViewBetaLeft (vector< double > &)=0

  Retrieves global weights for left subclusters of current clusters.
- virtual void ViewBetaRight (vector< double > &)=0

  Retrieves global weights for right subclusters of current clusters.
- virtual void PrintTheta (const std::string &)=0

  Print to file values of current clusters' latent parameters.
- virtual void PrintLambdaInfo () const =0

  Print to screen information about hyperparameters of latent parameters' distribution.

5.14.1 Detailed Description

template<typename Type, unsigned int DIM = 1>class ModelGeneric< Type, DIM >

Interface for the Model class Abstract class where all methods are virtual. Classes that inherit from ModelGeneric sample latent parameters and manages related hyperparameters, which are specific to the chosen model. Manages model specific functions, such as likelihood, marginals and other densities. Removes and adds clusters.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.14.2 Member Function Documentation

5.14.2.1 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::AddOneCluster (const unsigned *int*) [pure virtual]

Adds an empty cluster to the current clusters and updates the current number of clusters.

Parameters

_K	- new cluster id

Implemented in CategoricalModel< DIM >.

5.14.2.2 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::DefaultHyperparameter (size_t) [pure virtual]

Fixes hyperparameters for the latent parameters' distribution with default values.

Parameters

W	- hyperparameters' dimension

Implemented in CategoricalModel< DIM >.

5.14.2.3 template<typename Type, unsigned int DIM = 1> virtual long double ModelGeneric< Type, DIM >::LogDensity (const unsigned *int*) [pure virtual]

Computes the latent parameter's density, given the cluster id.

Parameters

_K	- cluster id

Returns

logdensity of cluster k

Implemented in CategoricalModel< DIM >.

5.14.2.4 template<typename Type, unsigned int DIM = 1> virtual double ModelGeneric< Type, DIM >::Loglikelihood (const typename Type::Point , const unsigned *int*) [pure virtual]

Computes the loglikelihood of datum X, given the cluster id.

Parameters

X	- datum
_K	- cluster id

Returns

loglikelihood of datum X in cluster k

5.14.2.5 template<typename Type, unsigned int DIM = 1> virtual double ModelGeneric< Type, DIM >::LoglikelihoodLeft (const typename Type::Point , const unsigned *int*) [pure virtual]

Computes the loglikelihood of datum X, given the left subcluster id.

Parameters

X	- datum
_K	- left subcluster id

Returns

loglikelihood of datum X in left subcluster of cluster k

5.14.2.6 template<typename Type, unsigned int DIM = 1> virtual double ModelGeneric< Type, DIM >::LoglikelihoodRight (const typename Type::Point , const unsigned *int*) [pure virtual]

Computes the loglikelihood of datum X, given the right subcluster id.

Parameters

X	- datum
_K	- right subcluster id

Returns

loglikelihood of datum X in right subcluster of cluster k

5.14.2.7 template<typename Type, unsigned int DIM = 1> virtual double ModelGeneric< Type, DIM >::Marginalized_Loglikelihood (const unsigned *int*) [pure virtual]

Computes the logarithm of the cluster's marginal likelihood, given the cluster id.

Parameters

_K	- cluster id

Implemented in CategoricalModel< DIM >.

5.14.2.8 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::RemoveClusters (const vector< unsigned int > &) [pure virtual]

Removes multiple clusters given their id and updates the current number of clusters.

Parameters

_K	- vector of clusters' ids to be removed

Implemented in CategoricalModel< DIM >.

5.14.2.9 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::RemoveOneCluster (const unsigned *int*) [pure virtual]

Removes a cluster given its id and updates the current number of clusters.

Parameters

_K	- id of the cluster to be removed

Implemented in CategoricalModel< DIM >.

5.14.2.10 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::SetHyperparameter (const typename Type::HYP &) [pure virtual]

Fixes hyperparameters for the latent parameters' distribution.

Parameters

_Lambda	- hyperparameters for the latent parameters' distribution

5.14.2.11 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::SetInitialClusters (unsigned *int*) [pure virtual]

Sets clusters assigning initial values for their weights.

Parameters

_K	- Initial number of clusters

Implemented in CategoricalModel< DIM >.

5.14.2.12 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::UpdateOneThetaCluster (const unsigned *int,* omprng &) [pure virtual]

Updates latent parameters of one cluster, given its id.

Parameters

_K	- cluster id
Gen	- parallel random number generator

Implemented in CategoricalModel< DIM >.

5.14.2.13 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::UpdateOneThetaSubCluster (const unsigned *int,* omprng &) [pure virtual]

Updates latent parameters of left and right subclusters of one cluster, given its id.

Parameters

_K	- cluster id
Gen	- parallel random number generator

Implemented in CategoricalModel< DIM >.

5.14.2.14 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::UpdateThetaCluster (omprng &) [pure virtual]

Updates latent parameters of all clusters.

Parameters

Gen	- parallel random number generator

Implemented in CategoricalModel< DIM >.

5.14.2.15 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::UpdateThetaSubCluster (omprng &) [pure virtual]

Updates latent parameters of left and right subclusters of all clusters.

Parameters

Gen	- parallel random number generator

Implemented in CategoricalModel< DIM >.

5.14.2.16 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::ViewBeta (vector< double > &) [pure virtual]

Retrieves current clusters' global weights.

Parameters

AllBeta	- vector that will be filled with global weights

Implemented in CategoricalModel< DIM >.

5.14.2.17 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::ViewBetaLeft (vector< double > &) [pure virtual]

Retrieves global weights for left subclusters of current clusters.

Parameters

AllBetaLeft	- vector that will be filled with global weights of left subclusters

Implemented in CategoricalModel< DIM >.

5.14.2.18 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::ViewBetaRight (vector< double > &) [pure virtual]

Retrieves global weights for right subclusters of current clusters.

Parameters

AllBetaRight	- vector that will be filled with global weights of right subclusters

Implemented in CategoricalModel< DIM >.

5.14.2.19 template<typename Type, unsigned int DIM = 1> virtual unsigned int ModelGeneric< Type, DIM >::ViewK ( ) const [pure virtual]

Retrieves current number of clusters.

Returns

current number of clusters

Implemented in CategoricalModel< DIM >.

5.14.2.20 template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::ViewKey (vector< unsigned int > &) const [pure virtual]

Retrieves current clusters' ids.

Parameters

Key	- object where current clusters' ids are stored

Implemented in CategoricalModel< DIM >.

The documentation for this class was generated from the following file:

- Model.hpp

5.15 NJK Struct Reference

Number of elements of group j in cluster k.

`#include <Struct.hpp>`

Public Attributes

- unsigned int **a**
- unsigned int **b**
- unsigned int **c**
- unsigned int **k**
- pair< unsigned int, unsigned int > **a_sub**
- pair< unsigned int, unsigned int > **b_sub**
- pair< unsigned int, unsigned int > **c_sub**

5.15.1 Detailed Description

Number of elements of group j in cluster k.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

The documentation for this struct was generated from the following file:

- Struct.hpp

5.16 NUMTABLE Struct Reference

Tables.

`#include <Struct.hpp>`

Public Attributes

- unsigned int **a_Left**
- unsigned int **a_Right**
- unsigned int **a**
- unsigned int **ja_Left**
- unsigned int **ja_Right**
- unsigned int **ja**
- unsigned int **Tilde_sum**
- vector< unsigned int > **Tilde_b_c**
- vector< unsigned int > **Tilde_k**

5.16.1 Detailed Description

Tables.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

The documentation for this struct was generated from the following file:

- Struct.hpp

5.17 omprng Class Reference

Omprng library for sampling random numbers in OpenMp.

`#include <omprng.hpp>`

Public Member Functions

- omprng ()
  DEfault constructor.
- ~omprng ()
  Destructor.
- void fixedSeed (long unsigned int)
  Set the seed.
- void randomSeed ()
  Generate a random seed.
- void setNumThreads (int)
  Set the number of threads.
- double runif ()
  Samples from the Uniform distribution between 0 and 1.
- double runif (double, double)
  Samples from the Uniform distribution between two fixed values.
- double rnorm (double, double)
  Samples from the Gaussian distribution.
- double rexp (double)
  Samples from the Exponential distribution.
- double rgamma (double, double)
  Samples from the Gamma distribution.
- double rchisq (unsigned int)
  Samples from the Chi-squared distribution.
- double rbeta (double, double)
  Samples from the Beta distribution.
- unsigned int rdiscrete (std::vector< double > &)
  Samples a discrete random variable with support 0:(K-1)
- unsigned int runifdiscrete (unsigned int)
  Samples a discrete uniform random variable with support 0:(N-1)
- unsigned int rbernoulli (double p)
  Samples from the Bernoulii distribution X~Bernoulli(p) P(X=1) = p P(X=0) = 1-p X={0,1}.
- unsigned int rbinomial (unsigned int n, double p)
  Samples from the Binomial distribution X~Bin(n,p)
- void rdirichlet (const vector< double > &params, vector< double > &dir_sampled)
  Samples from the d-dimensional Dirichlet distribution (X_1,...,X_d)~Dir(a_1,...,a_d) a_i >0 i={1,..,d} sum_{X_i} = 1.

Private Attributes

- int nprocs

  Number of processors available.
- RngStream * myRng

  RngStream object. Implemented by Matthew Bognar.

5.17.1 Detailed Description

Omprng library for sampling random numbers in OpenMp.

The library was developed by Matthew Bognar Department of Statistics and Actuarial Science University of Iowa http://www.stat.uiowa.edu/~mbognar/omprng matthew-bognar@uiowa.edu

5.17.2 Member Function Documentation

5.17.2.1 void omprng::fixedSeed (long unsigned int *myInt*)

Set the seed.

Parameters

Seed	- seed

5.17.2.2 unsigned int omprng::rbernoulli (double *p*)

Samples from the Bernoulii distribution X~Bernoulli(p) P(X=1) = p P(X=0) = 1-p X={0,1}.

Parameters

p	- success probability

Returns

a sample from the Bernoulii distribution with given probability of success

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.17.2.3 double omprng::rbeta (double *alpha,* double *beta*)

Samples from the Beta distribution.

X~beta(alpha,beta)

f(x) = gamma(alpha+beta)/(gamma(alpha)*gamma(beta)) * x^(alpha-1) * (1-x)^(beta-1)

0 < x < 1, alpha > 0, beta > 0

E(X) = alpha/(alpha+beta), Var(X) = alpha*beta / ((alpha+beta+1)*(alpha+beta)^2)

Parameters

alpha	- first shape parameter
beta	- second shape parameter

Returns

x - a sample from the Beta distribution with given shape parameters

5.17.2.4 unsigned int omprng::rbinomial (unsigned int *n,* double *p*)

Samples from the Binomial distribution X∼Bin(n,p)

P(X=k) = n!/(k!(n-k)!) p^k (1-p)^(n-k) k={0,1,...,n}

Parameters

n	- number of trials
p	- success probability

Returns

a sample from the Binomial distribution with given number of trials and probability of success

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.17.2.5 double omprng::rchisq (unsigned *int*)

Samples from the Chi-squared distribution.

X∼chisq(df)

f(x) = 1/(gamma(df/2)∗2^(df/2)) ∗ x^(df/2-1) ∗ exp(-x/2)

x > 0, df = 1,2,3,...

E(X) = df, Var(X) = 2∗df

Parameters

df	- degrees of freedom (dof)

Returns

a sample from the Chi-squared distribution with given dof

5.17.2.6 void omprng::rdirichlet (const vector< double > & *params,* vector< double > & *dir_sampled*)

Samples from the d-dimensional Dirichlet distribution (X_1,...,X_d)∼Dir(a_1,...,a_d) a_i >0 i={1,..,d} sum_{X_i} = 1.

Parameters

params	- vector of parameters
dir_sampled	- a sample from the d-dimensional Dirichlet distribution with given parameters

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

5.17.2.7 unsigned int omprng::rdiscrete (std::vector< double > & *weights*)

Samples a discrete random variable with support 0:(K-1)

K = size(inputvector)

Parameters

Logp	- logarithm of weights associated to atoms

Returns

a sample of a discrete random variable with given support ad weights

5.17.2.8 double omprng::rexp (double *theta*)

Samples from the Exponential distribution.

X$\sim$exp(theta)

f(x) = 1 /theta $*$ exp(-x/theta)

x $>$ 0, theta $>$ 0

E(X) = theta, Var(X) = theta$\hat{}$2

Parameters

theta	- scale parameter

Returns

a sample from the Exponential distribution with given scale parameter

5.17.2.9 double omprng::rgamma (double *alpha,* double *beta*)

Samples from the Gamma distribution.

X$\sim$gamma(alpha,beta)

f(x) = 1/(gamma(alpha)$*$beta$\hat{}$alpha) $*$ x$\hat{}$(alpha-1) $*$ exp(-x/beta)

x $>$ 0, alpha $>$ 0, beta $>$ 0

E(X) = alpha$*$beta, Var(X) = alpha$*$beta$\hat{}$2

Parameters

alpha	- shape parameter
beta	- scale parameter

Returns

a sample from the Gamma distribution with given shape and scale parameters

5.17.2.10 double omprng::rnorm (double , double)

Samples from the Gaussian distribution.

Parameters

mu	- mean
sigma	- standard deviation

Returns

a sample from the Gaussian distribution with given mean and variance

5.17.2.11 double omprng::runif ( )

Samples from the Uniform distribution between 0 and 1.

Returns

a sample from the Uniform distribution between 0 and 1

5.17.2.12 double omprng::runif (double *a*, double *b*)

Samples from the Uniform distribution between two fixed values.

Parameters

a	- lower bound
b	- upper bound

Returns

a sample from the Uniform distribution between two fixed values

5.17.2.13 unsigned int omprng::runifdiscrete (unsigned int *N*)

Samples a discrete uniform random variable with support 0:(N-1)

Parameters

N	- number of classes

Returns

x - a sample of a discrete uniform random variable with support 0:(N-1)

5.17.2.14 void omprng::setNumThreads (int *nt*)

Set the number of threads.

Parameters

NumThread	- number of threads

The documentation for this class was generated from the following files:

- omprng.hpp
- omprng.cpp

5.18 PI Struct Reference

Group specific clusters' weights.

`#include <Struct.hpp>`

Public Attributes

- vector< double > **b_c**
- double **a**
- vector< double > **Tilde_b_c**
- double **Left**
- double **Right**

5.18.1 Detailed Description

Group specific clusters' weights.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

The documentation for this struct was generated from the following file:

- Struct.hpp

5.19 RngStream Class Reference

Public Member Functions

- **RngStream** (const char *name="")
- void **ResetStartStream** ()
- void **ResetStartSubstream** ()
- void **ResetNextSubstream** ()
- void **SetAntithetic** (bool a)
- void **IncreasedPrecis** (bool incp)
- bool **SetSeed** (const unsigned long seed[6])
- void **AdvanceState** (long e, long c)
- void **GetState** (unsigned long seed[6]) const
- void **WriteState** () const
- void **WriteStateFull** () const
- double **RandU01** ()
- int **RandInt** (int i, int j)

Static Public Member Functions

- static bool **SetPackageSeed** (const unsigned long seed[6])

Private Member Functions

- double **U01** ()
- double **U01d** ()

Private Attributes

- double **Cg** [6]
- double **Bg** [6]
- double **Ig** [6]
- bool **anti**
- bool **incPrec**
- std::string **name**

Static Private Attributes

- static double **nextSeed** [6]

5.19.1 Member Data Documentation

5.19.1.1 double RngStream::nextSeed [static], [private]

Initial value:

```
=
{
   12345.0, 12345.0, 12345.0, 12345.0, 12345.0, 12345.0
}
```

The documentation for this class was generated from the following files:

- rngstream.hpp
- rngstream.cpp

5.20 TypeCategorical< DIM > Class Template Reference

This class defines data's types, with them it is possible to represent the categorcal likelihood.

`#include <Type.hpp>`

Public Types

- using THETA = vector< double >

  Vector of latent parameter of cluster or sub-cluster.
- using Point = unsigned int

  Type of single observation: Xji.
- using HYP = vector< double >

  Vector of latent parameter's hyperparameter of cluster or sub-cluster.
- using STAT = vector< unsigned int >

  Statistics' vector. In this container there are the update of latent parameters' hyperparameters of clusters or sub-clusters.

5.20.1 Detailed Description

template<unsigned int DIM = 1>class TypeCategorical< DIM >

This class defines data's types, with them it is possible to represent the categorcal likelihood.

Authors

{ Debora Parisi and Perego Stefania }

Date

February 2016

The documentation for this class was generated from the following file:

- Type.hpp

Chapter 6

File Documentation

6.1 Cluster.hpp File Reference

Data's structures which manage the cluster. This structure depend on the model. These classes define how to manage all parameter which are involve in the definition of cluster. In these classes there aren't any methods which sample some variables that describe the cluster. There are only methods that read, write and keep in memory information of cluster and his sub-clusters.

```
#include <vector>
#include <iostream>
#include "Type.hpp"
```

Classes

- class GenericCluster< ClassType, DIM >

  Generic Model of Cluster.

- class CategoricalCluster

  Management of cluster's and subclusters' informations with Categorical Likelihood.

6.1.1 Detailed Description

Data's structures which manage the cluster. This structure depend on the model. These classes define how to manage all parameter which are involve in the definition of cluster. In these classes there aren't any methods which sample some variables that describe the cluster. There are only methods that read, write and keep in memory information of cluster and his sub-clusters.

Date

February 2016

6.2 Document.hpp File Reference

This file contains classes which manage the documents or, in general, groups of data. The generic class provides the common interface, whereas the derived and specialized classes are specific to the model.

```
#include <unordered_map>
#include <utility>
#include <vector>
#include "Functions.hpp"
#include "Type.hpp"
#include <iostream>
#include <algorithm>
#include <iomanip>
#include <tuple>
#include <fstream>
```

Classes

- class GenericDocument< Type, DIM >

  Generic class for groups of data.

- class CategoricalDocument< DIM >

  Derived class for topic modeling, where data are categorical and the base measure is the Dirichlet distribution. This class represents a document in the topic modeling problem. It manages the words and is in charge of sampling the topic labels. It samples the model's parameters specific to the document: $\alpha, \pi_j, (\bar{\pi})_{jl}, (\bar{\pi})_{jr}, m_j, \bar{m}_{jl}, \bar{m}_{jl}$. *It keep track of words' counts in topics.*

6.2.1 Detailed Description

This file contains classes which manage the documents or, in general, groups of data. The generic class provides the common interface, whereas the derived and specialized classes are specific to the model.

Date

February 2016

6.3 Functions.hpp File Reference

```
#include <vector>
#include "omprng.hpp"
#include <math.h>
#include <random>
#include <algorithm>
#include <iostream>
#include <cmath>
#include <utility>
#include <iomanip>
#include <omp.h>
#include <limits>
```

Classes

- struct greater_for_pair

  Sorting operator for <unsigned int,double> pairs. The order is based on the second element in the pair.

Functions

- long double logsumexp (long double x1, long double x2)

  Allows to compute Stirling numbers.
- void ComputeLogStirlingNumbers (unsigned int N, vector< long double > &logstirling)

  Computes unsigned Stirling numbers of the first kind $|s(n,m)|$, for $n = 0, \ldots, N$, and stores them in the input vector. By definition: s(0,0) = s(1,1) = 1, s(n,0) = 0 for n > 0, s(n,m) = 0 for m > n, s(n,m) = s(n-1,m-1) + (n-1)∗s(n-1,m). To achieve greater precision, computes the numbers' logarithm.
- template<typename T >
  T Kahan_algorithm (vector< T > &numbers)

  Kahan_algorithm: computes the sum of the numbers contained in the input vector and reduces the round-off error due to the machine.
- unsigned int Antoniak (double alpha, double beta, unsigned int njk, const std::vector< long double > &LogStirling, omprng &Gen)

  Antoniak: samples tables in each group.
- unsigned int FindBestNumTable (double alpha, unsigned int K, unsigned int njk, vector< long double > &StirlingNumber)

  Computes $\tilde{m_{jb}}$, which approximates the tables in document j serving dish b.

6.3.1 Detailed Description

This file gathers useful functions for the algorithm, that are not model specific.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

6.3.2 Function Documentation

6.3.2.1 unsigned int Antoniak (double *alpha*, double *beta*, unsigned int *njk*, const std::vector< long double > & *LogStirling*, omprng & *Gen*)

Antoniak: samples tables in each group.

Parameters

Alpha	- concentration parameter of the Dirichlet process governing a group
Beta	- global weight for cluster k
njk	- number of element of group j in cluster k
LogStirling	- vector of the logarithm of the Stirling numbers
Gen	- parallel random number generator

Returns

number of tables in group j serving dish k

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

6.3.2.2 void ComputeLogStirlingNumbers (unsigned int *N,* vector< long double > & *logstirling*)

Computes unsigned Stirling numbers of the first kind |s(n,m)|, for $n = 0, \ldots, N$, and stores them in the input vector. By definition: s(0,0) = s(1,1) = 1, s(n,0) = 0 for n > 0, s(n,m) = 0 for m > n, s(n,m) = s(n-1,m-1) + (n-1)*s(n-1,m). To achieve greater precision, computes the numbers' logarithm.

Parameters

N	- max value for n
logstirling	- vector that stores the logarithm of Stirling numbers

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

6.3.2.3 unsigned int FindBestNumTable (double *alpha,* unsigned int *K,* unsigned int *njk,* vector< long double > & *StirlingNumber*)

Computes $\tilde{m_{jb}}$, which approximates the tables in document j serving dish b.

Parameters

alpha	- concentration parameter of the Dirichlet process governing a group
K	- current number of clusters
njk	- number of data of groupp j in cluster k
StirlingNumber	- vector of the logarithm of the Stirling numbers

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

6.3.2.4 template<typename T > T Kahan_algorithm (vector< T > & *numbers*)

Kahan_algorithm: computes the sum of the numbers contained in the input vector and reduces the round-off error due to the machine.

Parameters

numbers	- numbers to be summed

Returns

sum

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

6.3.2.5 long double logsumexp (long double *x1*, long double *x2*)

Allows to compute Stirling numbers.

Parameters

x1	first element
x2	second element

6.4 HDP_MCMC.hpp File Reference

```
#include "Model.hpp"
#include "Document.hpp"
#include "Struct.hpp"
#include "omprng.hpp"
#include "Functions.hpp"
#include <random>
#include <tuple>
#include <fstream>
#include <sstream>
#include <string>
#include <iostream>
#include <algorithm>
#include <cmath>
#include <iomanip>
#include <cstdlib>
#include <omp.h>
```

Classes

- class HDP_MCMC< MODEL, DOCUMENT, DIM >

  HDP_MCMC.

6.4.1 Detailed Description

HDP_MCMC class to implement the execution of algorithm

Date

Febbraio 2016

6.5 Model.hpp File Reference

```
#include "Cluster.hpp"
#include "Functions.hpp"
#include "omprng.hpp"
#include <vector>
#include <unordered_map>
#include <utility>
#include <cmath>
#include <iostream>
#include <sstream>
#include <fstream>
```

Classes

- class ModelGeneric< Type, DIM >

  Interface for the Model class Abstract class where all methods are virtual. Classes that inherit from ModelGeneric sample latent parameters and manages related hyperparameters, which are specific to the chosen model. Manages model specific functions, such as likelihood, marginals and other densities. Removes and adds clusters.

- class CategoricalModel< DIM >

  Specialized class for topic modeling, where data are categorical and H is the Dirichlet distribution. This class is used to sample latent parameters from the Dirichlet distribution, to compute likelihood and marginal distribution for categorical data. It also removes and adds topics and print values of latent parameters on file.

6.5.1 Detailed Description

Class that manages all clusters, sampling equations and functions that depend on the chosen model.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

6.6 omprng.hpp File Reference

Parallel random number generator for OpenMP.

```
#include <omp.h>
#include <iostream>
#include "rngstream.hpp"
#include "sys/time.h"
#include <cmath>
#include <vector>
```

Classes

- class omprng

  Omprng library for sampling random numbers in OpenMp.

6.6.1 Detailed Description

Parallel random number generator for OpenMP.

6.7 PosteriorAnalysis.hpp File Reference

```
#include <RInside.h>
#include <Rcpp.h>
#include <string>
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <queue>
#include <utility>
#include <unordered_map>
#include <algorithm>
#include <math.h>
#include <tuple>
#include <iomanip>
#include "Type.hpp"
```

Classes

- struct greater_for_pair

  Sorting operator for <unsigned int,double> pairs. The order is based on the second element in the pair.
- class GenericPosteriorAnalysis< Type, DIM >

  Generic class for the posterior analysis. Virtual class where all methods are null. Each inherited class must define all methods in the base class and, if necessary, add other methods. Calls R scripts. Computes LPML index, identifies the topics and detect the best clustering of data according to the least square criteria.
- class CategoricalPosteriorAnalysis< DIM >

  Class for the posterior analysis when data are categorical and H is the Dirichlet distribution. Reads and stores the results in suitable structures.

Functions

- template<typename T >
  double **CoeffSimilitudine::VectorNorm** (vector< T > &v)
- template<typename T >
  T **CoeffSimilitudine::ScalarProdoct** (vector< T > &v1, vector< T > &v2)
- template<typename T >
  double **CoeffSimilitudine::Coeff** (vector< T > &v1, vector< T > &v2)

6.7.1 Detailed Description

Contains classes for the posterior analysis of the algorithm's results. For each model there is a derived and specialized class. The generic class provides the common interface to all model specific classes.

Date

February 2016

6.8 Struct.hpp File Reference

Gathers structures used in the methods of HDP_MCMC class. We chose to create a separate file for the structures' definition because they are common to different methods.

```
#include <vector>
#include <utility>
#include <unordered_map>
#include "Type.hpp"
```

Classes

- struct BETA

  Clusters' global weights.
- struct NJK

  Number of elements of group j in cluster k.
- struct PI

  Group specific clusters' weights.
- struct NUMTABLE

  Tables.
- struct C

  Statistics.
- struct CLUSTER

  Structure for a cluster.
- struct DATACOUNT

  Structure for data counts.

Typedefs

- using **POINT** = TypeCategorical< 1 >::Point
- using **STAT** = TypeCategorical< 1 >::STAT

6.8.1 Detailed Description

Gathers structures used in the methods of HDP_MCMC class. We chose to create a separate file for the structures' definition because they are common to different methods.

Authors

{Debora Parisi and Stefania Perego}

Date

February 2016

6.9 Type.hpp File Reference

Data's structures which define the model's type and clusters' type.

```
#include <vector>
#include <iostream>
```

Classes

- class TypeCategorical< DIM >

  This class defines data's types, with them it is possible to represent the categorcal likelihood.

6.9.1 Detailed Description

Data's structures which define the model's type and clusters' type.