

Parallel Split-Merge MCMC for the HDP

Generato da Doxygen 1.8.6

Mer 24 Feb 2016 13:09:30

Indice

1	Hierarchical Dirichlet Process Mixture Model parallelizzabile	1
1.1	Introduzione	1
2	Indice dei namespace	3
2.1	Lista dei namespace	3
3	Indice della gerarchia	5
3.1	Gerarchia delle classi	5
4	Indice dei tipi composti	7
4.1	Elenco dei tipi composti	7
5	Indice dei file	9
5.1	Elenco dei file	9
6	Documentazione dei namespace	11
6.1	Riferimenti per il namespace <code>CoeffSimilitudine</code>	11
6.1.1	Descrizione dettagliata	11
6.1.2	Documentazione delle funzioni	11
6.1.2.1	<code>Coeff</code>	11
6.1.2.2	<code>ScalarProduct</code>	12
6.1.2.3	<code>VectorNorm</code>	13
7	Documentazione delle classi	15
7.1	Riferimenti per la struct <code>BETA</code>	15
7.1.1	Descrizione dettagliata	15
7.2	Riferimenti per la struct <code>C</code>	15
7.2.1	Descrizione dettagliata	16
7.3	Riferimenti per la classe <code>CategoricalCluster</code>	16
7.3.1	Descrizione dettagliata	18
7.3.2	Documentazione dei costruttori e dei distruttori	18
7.3.2.1	<code>CategoricalCluster</code>	18
7.3.3	Documentazione delle funzioni membro	19
7.3.3.1	<code>IsEmpty</code>	19

7.3.3.2	ResetStatistics	19
7.3.3.3	ResetStatisticsLeft	19
7.3.3.4	ResetStatisticsRight	19
7.3.3.5	SetBeta	20
7.3.3.6	SetBetaLeft	20
7.3.3.7	SetBetaRight	20
7.3.3.8	SetGlobalTable	20
7.3.3.9	SetGlobalTableLeft	20
7.3.3.10	SetGlobalTableRight	20
7.3.3.11	SetStatistics	21
7.3.3.12	SetStatisticsLeft	21
7.3.3.13	SetStatisticsRight	21
7.3.3.14	SetTheta	21
7.3.3.15	SetThetaLeft	21
7.3.3.16	SetThetaRight	21
7.3.3.17	UpdateStatistics	22
7.3.3.18	ViewBeta	22
7.3.3.19	ViewBetaLeft	22
7.3.3.20	ViewBetaRight	22
7.3.3.21	ViewGlobalTable	22
7.3.3.22	ViewGlobalTableLeft	23
7.3.3.23	ViewGlobalTableRight	23
7.3.3.24	ViewStatistics	23
7.3.3.25	ViewStatisticsLeft	23
7.3.3.26	ViewStatisticsRight	23
7.3.3.27	ViewTheta	23
7.3.3.28	ViewThetald	24
7.3.3.29	ViewThetaLeft	24
7.3.3.30	ViewThetaLeftId	24
7.3.3.31	ViewThetaRight	24
7.3.3.32	ViewThetaRightId	24
7.4	Template per la classe CategoricalDocument< DIM >	25
7.4.1	Descrizione dettagliata	29
7.4.2	Documentazione dei costruttori e dei distruttori	29
7.4.2.1	CategoricalDocument	29
7.4.3	Documentazione delle funzioni membro	29
7.4.3.1	CheckLeftSubcluster	29
7.4.3.2	CheckRightSubcluster	29
7.4.3.3	InsertNewCluster	29
7.4.3.4	RemoveCluster	30

7.4.3.5	RemoveCluster	30
7.4.3.6	RemoveCluster	30
7.4.3.7	ResetDataCountSub	30
7.4.3.8	Sampling	30
7.4.3.9	SetAlpha	31
7.4.3.10	SetDataset	31
7.4.3.11	SetNj	31
7.4.3.12	SortData	31
7.4.3.13	UpdateAllLocalTableSub	31
7.4.3.14	UpdateAllPiSub	32
7.4.3.15	UpdateLocalTable	32
7.4.3.16	UpdateLocalTableSub_OneCluster	32
7.4.3.17	UpdatePi	32
7.4.3.18	UpdatePiSub	33
7.4.3.19	UpdateZeta	33
7.4.3.20	UpdateZeta	33
7.4.3.21	UpdateZeta	33
7.4.3.22	UpdateZeta_and_Sub	33
7.4.3.23	UpdateZetaSub	34
7.4.3.24	ViewCluster	34
7.4.3.25	ViewCounts4c	34
7.4.3.26	ViewData	34
7.4.3.27	ViewDataCount	35
7.4.3.28	ViewDataCountID	35
7.4.3.29	ViewDataCountLeft	35
7.4.3.30	ViewDataCountLeftID	35
7.4.3.31	ViewDataCountRight	35
7.4.3.32	ViewDataCountRightID	36
7.4.3.33	ViewIdCounts	36
7.4.3.34	ViewLabel	36
7.4.3.35	ViewNj	36
7.4.3.36	ViewNumTableID	36
7.4.3.37	ViewNumTableLeftID	37
7.4.3.38	ViewNumTableRightID	37
7.4.3.39	ViewPi	37
7.4.3.40	ViewPiID	37
7.4.3.41	ViewPiLeft	38
7.4.3.42	ViewPiLeftID	39
7.4.3.43	ViewPiRight	39
7.4.3.44	ViewPiRightID	39

7.5	Template per la classe <code>CategoricalModel< DIM ></code>	39
7.5.1	Descrizione dettagliata	41
7.5.2	Documentazione delle funzioni membro	42
7.5.2.1	<code>AddOneCluster</code>	42
7.5.2.2	<code>DefaultHyperparameter</code>	43
7.5.2.3	<code>LogDensity</code>	43
7.5.2.4	<code>Loglikelihood</code>	43
7.5.2.5	<code>LoglikelihoodLeft</code>	43
7.5.2.6	<code>LoglikelihoodRight</code>	44
7.5.2.7	<code>Marginalized_Loglikelihood</code>	44
7.5.2.8	<code>operator=</code>	44
7.5.2.9	<code>operator=</code>	44
7.5.2.10	<code>operator[]</code>	44
7.5.2.11	<code>PrintLambdaInfo</code>	45
7.5.2.12	<code>RemoveClusters</code>	45
7.5.2.13	<code>RemoveOneCluster</code>	45
7.5.2.14	<code>SetHyperparameter</code>	45
7.5.2.15	<code>SetInitialClusters</code>	45
7.5.2.16	<code>UpdateOneThetaCluster</code>	45
7.5.2.17	<code>UpdateOneThetaSubCluster</code>	46
7.5.2.18	<code>UpdateThetaCluster</code>	46
7.5.2.19	<code>UpdateThetaSubCluster</code>	46
7.5.2.20	<code>ViewBeta</code>	46
7.5.2.21	<code>ViewBetaLeft</code>	46
7.5.2.22	<code>ViewBetaRight</code>	47
7.5.2.23	<code>ViewK</code>	47
7.5.2.24	<code>ViewKey</code>	47
7.5.3	Documentazione dei membri dato	47
7.5.3.1	<code>OMP_NUM_THREADS</code>	47
7.6	Template per la classe <code>CategoricalPosteriorAnalysis< DIM ></code>	47
7.6.1	Descrizione dettagliata	49
7.6.2	Documentazione delle funzioni membro	50
7.6.2.1	<code>AGPosteriorAnalysis</code>	50
7.6.2.2	<code>KPosteriorAnalysis</code>	51
7.6.2.3	<code>LeastSquareClustering</code>	51
7.6.2.4	<code>LoadLabels</code>	51
7.6.2.5	<code>LPML</code>	51
7.6.2.6	<code>SetAllAlpha</code>	52
7.6.2.7	<code>SetAllGamma</code>	52
7.6.2.8	<code>SetAllK</code>	52

7.6.2.9	SetD	52
7.6.2.10	SetN	52
7.6.2.11	SetTheta	53
7.6.2.12	SetW	53
7.6.2.13	Setwd	53
7.6.2.14	TrackingClusters	53
7.6.2.15	VisualizeBeta	53
7.7	Riferimenti per la struct CLUSTER	54
7.8	Riferimenti per la struct DATACOUNT	54
7.9	Template per la classe GenericCluster< ClassType, DIM >	54
7.9.1	Descrizione dettagliata	56
7.9.2	Documentazione delle funzioni membro	56
7.9.2.1	IsEmpty	56
7.9.2.2	ResetStatistics	56
7.9.2.3	ResetStatisticsLeft	56
7.9.2.4	ResetStatisticsRight	57
7.9.2.5	SetBeta	57
7.9.2.6	SetBetaLeft	57
7.9.2.7	SetBetaRight	57
7.9.2.8	SetGlobalTable	57
7.9.2.9	SetGlobalTableLeft	58
7.9.2.10	SetGlobalTableRight	58
7.9.2.11	SetStatistics	58
7.9.2.12	SetStatisticsLeft	58
7.9.2.13	SetStatisticsRight	58
7.9.2.14	SetTheta	59
7.9.2.15	SetThetaLeft	60
7.9.2.16	SetThetaRight	60
7.9.2.17	UpdateStatistics	60
7.9.2.18	ViewBeta	60
7.9.2.19	ViewBetaLeft	60
7.9.2.20	ViewBetaRight	61
7.9.2.21	ViewGlobalTable	61
7.9.2.22	ViewGlobalTableLeft	61
7.9.2.23	ViewGlobalTableRight	61
7.9.2.24	ViewStatistics	61
7.9.2.25	ViewStatisticsLeft	61
7.9.2.26	ViewStatisticsRight	62
7.9.2.27	ViewTheta	62
7.9.2.28	ViewThetaLeft	62

7.9.2.29 ViewThetaRight	62
7.10 Template per la classe GenericDocument< Type, DIM >	62
7.10.1 Descrizione dettagliata	65
7.10.2 Documentazione delle funzioni membro	65
7.10.2.1 CheckLeftSubcluster	65
7.10.2.2 CheckRightSubcluster	65
7.10.2.3 InsertNewCluster	65
7.10.2.4 RemoveCluster	66
7.10.2.5 RemoveCluster	67
7.10.2.6 RemoveCluster	67
7.10.2.7 ResetDataCountSub	67
7.10.2.8 Sampling	67
7.10.2.9 SetAlpha	67
7.10.2.10 SetDataset	68
7.10.2.11 SetNj	68
7.10.2.12 SortData	68
7.10.2.13 UpdateAllLocalTableSub	68
7.10.2.14 UpdateAllPiSub	68
7.10.2.15 UpdateLocalTable	69
7.10.2.16 UpdateLocalTableSub_OneCluster	69
7.10.2.17 UpdatePi	69
7.10.2.18 UpdatePiSub	69
7.10.2.19 UpdateZeta	70
7.10.2.20 UpdateZeta	70
7.10.2.21 UpdateZeta	70
7.10.2.22 UpdateZeta_and_Sub	70
7.10.2.23 UpdateZetaSub	71
7.10.2.24 ViewCluster	72
7.10.2.25 ViewCounts4c	72
7.10.2.26 ViewData	72
7.10.2.27 ViewDataCount	72
7.10.2.28 ViewDataCountID	72
7.10.2.29 ViewDataCountLeft	73
7.10.2.30 ViewDataCountLeftID	73
7.10.2.31 ViewDataCountRight	73
7.10.2.32 ViewDataCountRightID	73
7.10.2.33 ViewIdCounts	74
7.10.2.34 ViewLabel	75
7.10.2.35 ViewNj	75
7.10.2.36 ViewNumTableID	75

7.10.2.37 ViewNumTableLeftID	75
7.10.2.38 ViewNumTableRightID	75
7.10.2.39 ViewPi	76
7.10.2.40 ViewPiID	76
7.10.2.41 ViewPiLeft	76
7.10.2.42 ViewPiLeftID	76
7.10.2.43 ViewPiRight	77
7.10.2.44 ViewPiRightID	78
7.11 Template per la classe GenericPosteriorAnalysis< Type, DIM >	78
7.11.1 Descrizione dettagliata	79
7.11.2 Documentazione delle funzioni membro	79
7.11.2.1 AGPosteriorAnalysis	79
7.11.2.2 KPosteriorAnalysis	79
7.11.2.3 LeastSquareClustering	80
7.11.2.4 LoadLabels	80
7.11.2.5 LPML	80
7.11.2.6 SetAllAlpha	80
7.11.2.7 SetAllGamma	80
7.11.2.8 SetAllK	81
7.11.2.9 SetD	81
7.11.2.10 SetN	81
7.11.2.11 SetTheta	81
7.11.2.12 SetW	81
7.11.2.13 Setwd	81
7.11.2.14 TrackingClusters	82
7.11.2.15 VisualizeBeta	82
7.12 Riferimenti per la struct greater_for_pair	82
7.12.1 Descrizione dettagliata	82
7.13 Template per la classe HDP_MCMC< MODEL, DOCUMENT, DIM >	82
7.13.1 Descrizione dettagliata	87
7.13.2 Documentazione delle funzioni membro	87
7.13.2.1 Algorithm	87
7.13.2.2 Check_Model	87
7.13.2.3 Gibbs_SubCluster	87
7.13.2.4 IsEmptySubcluster	88
7.13.2.5 SetAlphaFixed	88
7.13.2.6 SetAlphaPrior	88
7.13.2.7 SetDataset	88
7.13.2.8 SetGammaFixed	88
7.13.2.9 SetGammaPrior	88

7.13.2.10 SetK_init	89
7.13.2.11 SetLambdaInfo	89
7.13.2.12 SetSeed	89
7.13.2.13 Swap	89
7.13.2.14 UpdateBetaSub	89
7.13.2.15 ViewD	90
7.13.2.16 ViewK	90
7.13.2.17 ViewN	90
7.13.2.18 ViewW	90
7.13.3 Documentazione dei membri dato	90
7.13.3.1 Model	90
7.13.3.2 OMP_NUM_THREADS	91
7.14 Template per la classe ModelGeneric< Type, DIM >	91
7.14.1 Descrizione dettagliata	92
7.14.2 Documentazione delle funzioni membro	92
7.14.2.1 AddOneCluster	92
7.14.2.2 DefaultHyperparameter	92
7.14.2.3 LogDensity	92
7.14.2.4 Loglikelihood	93
7.14.2.5 LoglikelihoodLeft	93
7.14.2.6 LoglikelihoodRight	93
7.14.2.7 Marginalized_Loglikelihood	93
7.14.2.8 PrintLambdaInfo	94
7.14.2.9 RemoveClusters	94
7.14.2.10 RemoveOneCluster	94
7.14.2.11 SetHyperparameter	94
7.14.2.12 SetInitialClusters	94
7.14.2.13 UpdateOneThetaCluster	95
7.14.2.14 UpdateOneThetaSubCluster	95
7.14.2.15 UpdateThetaCluster	95
7.14.2.16 UpdateThetaSubCluster	95
7.14.2.17 ViewBeta	95
7.14.2.18 ViewBetaLeft	96
7.14.2.19 ViewBetaRight	96
7.14.2.20 ViewK	96
7.14.2.21 ViewKey	96
7.15 Riferimenti per la struct NJK	96
7.15.1 Descrizione dettagliata	97
7.16 Riferimenti per la struct NUMTABLE	97
7.16.1 Descrizione dettagliata	97

7.17	Riferimenti per la classe <code>omprng</code>	97
7.17.1	Descrizione dettagliata	98
7.17.2	Documentazione delle funzioni membro	99
7.17.2.1	<code>fixedSeed</code>	99
7.17.2.2	<code>rbernoulli</code>	99
7.17.2.3	<code>rbeta</code>	99
7.17.2.4	<code>rbinomial</code>	99
7.17.2.5	<code>rchisq</code>	100
7.17.2.6	<code>rdirichlet</code>	100
7.17.2.7	<code>rdiscrete</code>	100
7.17.2.8	<code>rexp</code>	101
7.17.2.9	<code>rgamma</code>	101
7.17.2.10	<code>rnorm</code>	101
7.17.2.11	<code>runif</code>	101
7.17.2.12	<code>runif</code>	102
7.17.2.13	<code>runifdiscrete</code>	103
7.17.2.14	<code>setNumThreads</code>	103
7.18	Riferimenti per la struct <code>PI</code>	103
7.18.1	Descrizione dettagliata	103
7.19	Riferimenti per la classe <code>RngStream</code>	104
7.19.1	Documentazione dei membri dato	104
7.19.1.1	<code>nextSeed</code>	104
7.20	Template per la classe <code>TypeCategorical< DIM ></code>	105
7.20.1	Descrizione dettagliata	105
8	Documentazione dei file	107
8.1	Riferimenti per il file <code>Cluster.hpp</code>	107
8.1.1	Descrizione dettagliata	107
8.2	Riferimenti per il file <code>Document.hpp</code>	107
8.2.1	Descrizione dettagliata	108
8.3	Riferimenti per il file <code>Functions.hpp</code>	108
8.3.1	Descrizione dettagliata	109
8.3.2	Documentazione delle funzioni	109
8.3.2.1	<code>Antoniak</code>	109
8.3.2.2	<code>ComputeLogStirlingNumbers</code>	109
8.3.2.3	<code>FindBestNumTable</code>	110
8.3.2.4	<code>Kahan_algorithm</code>	111
8.4	Riferimenti per il file <code>HDP_MCMC.hpp</code>	111
8.4.1	Descrizione dettagliata	111
8.5	Riferimenti per il file <code>Model.hpp</code>	112

8.5.1	Descrizione dettagliata	112
8.6	Riferimenti per il file omprng.hpp	112
8.6.1	Descrizione dettagliata	112
8.7	Riferimenti per il file PosteriorAnalysis.hpp	113
8.7.1	Descrizione dettagliata	113
8.8	Riferimenti per il file Struct.hpp	114
8.8.1	Descrizione dettagliata	114
8.9	Riferimenti per il file Type.hpp	114
8.9.1	Descrizione dettagliata	115

Capitolo 1

Hierarchical Dirichlet Process Mixture Model parallelizzabile

1.1 Introduzione

In questo codice e' implementato l'algoritmo proposto nel paper: J. Chang, J. W. Fisher III Parallel Sampling of HDPs using sub-clusters splits, NIPS, 2014.

Nel paper si fa riferimento solo al problema del topic-modeling, ma l'algoritmo e' estendibile a problemi di altro tipo, vedere il capitolo 7 di [Relazione_Parisi_Perego.pdf](#). In questo algoritmo si alternano passi di Gibbs sampler a passi di Metropolis-Hastings. Nei passi di Gibbs sampler si considerano soltanto i cluster non vuoti e si aggiornano le quantita' di interesse (β, π, θ) usando le full-conditional, equazioni dalla 3.3 alla 3.11; queste ultime non sono altro che alcuni step dell'algoritmo. Durante questi passi il numero di cluster rimane invariato. Con i passi di Metropolis-Hastings, invece, si propone l'unione di due cluster (mosse di merge) oppure la divisione di un cluster (mosse di split). Per ogni topic k si individuano due sub-topic, kl e kr che rispettivamente corrispondono al sub-topic sinistro e destro; i nuovi topic si propongono sulla base di tali sub-topic.

Per quanto riguarda l'inferenza sui parametri latenti e' stato trattato solo il caso di modelli coniugati.

Capitolo 2

Indice dei namespace

2.1 Lista dei namespace

Questa è l'elenco dei namespace documentati, con una loro breve descrizione:

[CoeffSimilitudine](#)

Namespace per il calcolo del coefficiente di similitudine Contiene tre funzioni template utili al calcolo del coefficiente di similitudine tra vettori di un qualsiasi tipo 11

Capitolo 3

Indice della gerarchia

3.1 Gerarchia delle classi

Questo elenco di ereditarietà è ordinato approssimativamente, ma non completamente, in ordine alfabetico:

BETA	15
C	15
CLUSTER	54
DATACOUNT	54
GenericCluster< ClassType, DIM >	54
GenericCluster< TypeCategorical, 1 >	54
CategoricalCluster	16
GenericDocument< Type, DIM >	62
GenericDocument< TypeCategorical< DIM >, DIM >	62
CategoricalDocument< DIM >	25
GenericPosteriorAnalysis< Type, DIM >	78
GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >	78
CategoricalPosteriorAnalysis< DIM >	47
greater_for_pair	82
HDP_MCMC< MODEL, DOCUMENT, DIM >	82
ModelGeneric< Type, DIM >	91
ModelGeneric< TypeCategorical< DIM >, DIM >	91
CategoricalModel< DIM >	39
NJK	96
NUMTABLE	97
omprng	97
PI	103
RngStream	104
TypeCategorical< DIM >	105
TypeCategorical< 1 >	105

Capitolo 4

Indice dei tipi composti

4.1 Elenco dei tipi composti

Queste sono le classi, le struct, le union e le interfacce con una loro breve descrizione:

BETA		
	Pesi globali dei cluster	15
C		
	Statistiche	15
CategoricalCluster		
	Gestione informazioni cluster e sub-cluster per dati di verosimiglianza Categorical	16
CategoricalDocument< DIM >		
	Classe derivata per il modello Dirichlet-Categorical	25
CategoricalModel< DIM >		
	Modello Dirichlet-Categorical	39
CategoricalPosteriorAnalysis< DIM >		
	Modello Dirichlet-Categorical	47
CLUSTER		
	CLUSTER	54
DATACOUNT		
	DATACOUNT	54
GenericCluster< ClassType, DIM >		
	Modello Generico di Cluster	54
GenericDocument< Type, DIM >		
	Classe generica per i gruppi	62
GenericPosteriorAnalysis< Type, DIM >		
	Classe generica per l'analisi a posteriori Classe virtuale dove tutti i metodi sono null. Ogni classe che eredita deve definire tutti i metodi della classe base e, se necessario, può aggiungere altri metodi. Invoca gli script R per l'analisi delle catene MCMC. Calcola l'indice LPML, riconosce i topic e individua il miglior clustering secondo il criterio dei minimi quadrati	78
greater_for_pair		
	Criterio di confronto tra due elementi che sono coppie (unsigned int, double), da utilizzare nel sort di un vettore; il confronto è sul secondo valore nella coppia	82
HDP_MCMC< MODEL, DOCUMENT, DIM >		
	HDP_MCMC	82
ModelGeneric< Type, DIM >		
	Modello Generico di Model	91
NJK		
	Numero degli elementi del gruppo j che sono nel cluster k	96
NUMTABLE		
	Tavoli	97
omprng		
	Libreria Omprng per la generazione dei numeri casuali in OpenMp	97
PI		
	Pesi dei cluster in ogni gruppo	103
RngStream		
	RngStream	104

[TypeCategorical< DIM >](#)

Classe dei tipi per dati con verosimiglianza categorica [105](#)

Capitolo 5

Indice dei file

5.1 Elenco dei file

Questo è un elenco dei file documentati con una loro breve descrizione:

Cluster.hpp	Strutture dei dati per la gestione dei cluster in base al modello scelto. In queste classi si definisce solo come gestire i parametri latenti e gli altri parametri del cluster. Non sono presenti campionamenti, ma solo metodi che stampano e fissano i valori dei cluster e sub-cluster . . .	107
Document.hpp	In questo file sono presenti le classi che gestiscono i documenti o, piu' in generale, i gruppi di dati. La classe generica fornisce l'interfaccia comune, mentre le classi derivate e specializzate sono specifiche del modello	107
Functions.hpp		108
HDP_MCMC.hpp		111
Model.hpp	Classe che si occupa di gestire tutti i cluster, campionamenti che dipendono dal modello scelto e funzioni che dipendono dal modello	112
omprng.hpp	Generatore di numeri casuali in parallelo per OpenMP	112
PosteriorAnalysis.hpp		113
rngstream.hpp		??
Struct.hpp	Raccolta delle strutture che vengono utilizzate nei metodi di HDP_MCMC.hpp . Si è scelto di creare un file a parte con la dichiarazione delle strutture, perché il loro utilizzo è comune a più metodi	114
Type.hpp	Strutture dati per i tipi di modello e dei cluster	114

Capitolo 6

Documentazione dei namespace

6.1 Riferimenti per il namespace CoeffSimilitudine

Namespace per il calcolo del coefficiente di similitudine Contiene tre funzioni template utili al calcolo del coefficiente di similitudine tra vettori di un qualsiasi tipo.

Funzioni

- `template<typename T >`
`double VectorNorm (vector< T > &v)`
Calcola la norma di un vettore.
- `template<typename T >`
`T ScalarProduct (vector< T > &v1, vector< T > &v2)`
Calcola il prodotto scalare tra due vettori.
- `template<typename T >`
`double Coeff (vector< T > &v1, vector< T > &v2)`
Calcola il coefficiente di similitudine tra due vettori.

6.1.1 Descrizione dettagliata

Namespace per il calcolo del coefficiente di similitudine Contiene tre funzioni template utili al calcolo del coefficiente di similitudine tra vettori di un qualsiasi tipo.

6.1.2 Documentazione delle funzioni

6.1.2.1 `template<typename T > double CoeffSimilitudine::Coeff (vector< T > & v1, vector< T > & v2)`

Calcola il coefficiente di similitudine tra due vettori.

Parametri

<code>v1</code>	- primo vettore
<code>v2</code>	- secondo vettore

Restituisce

Coefficiente di similitudine

6.1.2.2 `template<typename T > T CoeffSimilitudine::ScalarProduct (vector< T > & v1, vector< T > & v2)`

Calcola il prodotto scalare tra due vettori.

Parametri

$v1$	- primo vettore
$v2$	- secondo vettore

Restituisce

Prodotto scalare tra i due vettori

6.1.2.3 `template<typename T > double CoeffSimilitudine::VectorNorm (vector< T > & v)`

Calcola la norma di un vettore.

Parametri

v	- vettore di cui calcolare la norma
-----	-------------------------------------

Restituisce

Norma del vettore

Capitolo 7

Documentazione delle classi

7.1 Riferimenti per la struct BETA

Pesi globali dei cluster.

```
#include <Struct.hpp>
```

Attributi pubblici

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

7.1.1 Descrizione dettagliata

Pesi globali dei cluster.

La documentazione per questa struct è stata generata a partire dal seguente file:

- [Struct.hpp](#)

7.2 Riferimenti per la struct C

Statistiche.

```
#include <Struct.hpp>
```

Attributi pubblici

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

7.2.1 Descrizione dettagliata

Statistiche.

La documentazione per questa struct è stata generata a partire dal seguente file:

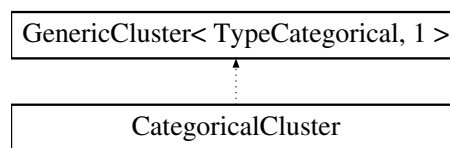
- [Struct.hpp](#)

7.3 Riferimenti per la classe CategoricalCluster

Gestione informazioni cluster e sub-cluster per dati di verosimiglianza Categorical.

```
#include <Cluster.hpp>
```

Diagramma delle classi per CategoricalCluster



Tipi pubblici

- using **THETA** = [TypeCategorical< 1 >::THETA](#)
Parametro latente, vettore di pesi degli elementi distinti nel cluster.
- using **Point** = [TypeCategorical< 1 >::Point](#)
Singlo dato, dati ripetuti.
- using **STAT** = [TypeCategorical< 1 >::STAT](#)
Statistiche per aggiornare l'iperparametro del parametro latente, numero di dati che sono contenuti nel cluster e sub-cluster.

Membri pubblici

- [CategoricalCluster](#) ()
Costruttore di default.
- [~CategoricalCluster](#) ()=default
Distruttore di default.
- [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)
Costruttore che richiede tutte le informazioni del cluster e subcluster.
- void [SetTheta](#) ([THETA](#) &_Theta)
Fissa il parametro latente del cluster.
- void [SetThetaLeft](#) ([THETA](#) &_ThetaLeft)
Fissa il parametro latente del sub-cluster sinistro.
- void [SetThetaRight](#) ([THETA](#) &_ThetaRight)
Fissa il parametro latente del sub-cluster destro.
- void [SetBeta](#) (double _Beta)
Fissa il peso del cluster.
- void [SetBetaLeft](#) (double _BetaLeft)
Fissa il peso del sub-cluster sinistro.

- void [SetBetaRight](#) (double _BetaRight)
Fissa il peso del sub-cluster destro.
- void [SetGlobalTable](#) (unsigned int _NrTable)
Fissa il numero di tavoli globali che caratterizza il cluster.
- void [SetGlobalTableLeft](#) (unsigned int _NrTableLeft)
Fissa il numero di tavoli globali che caratterizza il sub-cluster sinistro.
- void [SetGlobalTableRight](#) (unsigned int _NrTableRight)
Fissa il numero di tavoli globali che caratterizza il sub-cluster destro.
- void [SetStatistics](#) (STAT &_c)
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 [ViewThetald](#) (unsigned int _id) const
Estrae l'i-esimo elemento del parametro latente del cluster.
- double [ViewThetaLeftld](#) (unsigned int _id) const
Estrae l'i-esimo elemento del parametro latente del sub-cluster sinistro.
- double [ViewThetaRightld](#) (unsigned int _id) const
Estrae 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
Estra la statistica del parametro latente del cluster.
- void [ViewStatisticsLeft](#) (STAT &_cLeft) const
Estra la statistica del parametro latente del sub-cluster sinistro.
- void [ViewStatisticsRight](#) (STAT &_cRight) const
Estra la statistica del parametro latente del sub-cluster destro.
- void [ResetStatistics](#) (unsigned int W)
Azzera le statistiche nel cluster.
- void [ResetStatisticsLeft](#) (unsigned int W)
Azzera le statistiche nel sub-cluster sinistro.
- void [ResetStatisticsRight](#) (unsigned int W)
Azzera le statistiche nel sub-cluster destro.
- void [UpdateStatistics](#) (STAT &counts4cleft, STAT &counts4right)
Aggiorna le statistiche, ovvero gli iperparametri dei parametri latenti dei cluster e sub-cluster.
- bool [IsEmpty](#) () const
Controlla se il cluster non contiene elementi.

Attributi privati

- double [Beta](#)
Peso globale del cluster.
- double [BetaLeft](#)
Peso globale del sub-cluster sinistro.
- double [BetaRight](#)
Peso globale del sub-cluster destro.
- [THETA Theta](#)
Parametro latente del cluster: peso degli elementi distinti nel cluster.
- [THETA ThetaLeft](#)
Parametro latente del sub-cluster sinistro: peso degli elementi distinti nel sub-cluster sinistro.
- [THETA ThetaRight](#)
Parametro latente del sub-cluster destro: peso degli elementi distinti nel sub-cluster destro.
- [STAT c](#)
Statisitca per aggiornare il parametro latente del cluster: conteggi degli elementi finiti nel cluster.
- [STAT cLeft](#)
Statisitca per aggiornare il parametro latente del sub-cluster sinistro: conteggi degli elementi finiti nel sub-cluster sinistro.
- [STAT cRight](#)
Statisitca per aggiornare il parametro latente del sub-cluster destro: conteggi degli elementi finiti nel sub-cluster destro.
- unsigned int [NrTable](#)
Numero di tavoli nel cluster.
- unsigned int [NrTableLeft](#)
Numero di tavoli nel sub-cluster sinistro.
- unsigned int [NrTableRight](#)
Numero di tavoli nel sub-cluster destro.

Altri membri ereditati

7.3.1 Descrizione dettagliata

Gestione informazioni cluster e sub-cluster per dati di verosimiglianza Categorical.

Questa classe si occupa di memorizzare ed estrarre informazioni riguardanti il peso globale del cluster e sub-cluster, parametri latenti dei cluster e sub-cluster, informazioni inerenti agli aggiornamenti degli iperparametri dei parametri latenti. Nel caso di verosimiglianza caregorica. i dati sono ripetuti, i parametri latenti sono i pesi degli elementi distinti e le statistiche sono i conteggi degli elementi distinti nel cluster. I parametri latenti non sono altro che i parametri della mistura.

Data

Febbraio 2016

7.3.2 Documentazione dei costruttori e dei distruttori

7.3.2.1 `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)`

Costruttore che richiede tutte le informazioni del cluster e subcluster.

Parametri

<code>_Beta</code>	- Peso globale del cluster
<code>_BetaLeft</code>	- Peso globale del sub-cluster sinistro
<code>_BetaRight</code>	- Peso globale del sub-cluster destro
<code>_Theta</code>	- Parametro latente del cluster: peso degli elementi distinti nel cluster
<code>_ThetaLeft</code>	- Parametro latente del sub-cluster sinistro: peso degli elementi distinti nel sub-cluster sinistro
<code>_ThetaRight</code>	- Parametro latente del sub-cluster destro: peso degli elementi distinti nel sub-cluster destro
<code>_c</code>	- Statistica per aggiornare il parametro latente del cluster: conteggi degli elementi finiti nel cluster
<code>_cLeft</code>	- Statistica per aggiornare il parametro latente del sub-cluster sinistro: conteggi degli elementi finiti nel sub-cluster sinistro
<code>_cRight</code>	- Statistica per aggiornare il parametro latente del sub-cluster destro: conteggi degli elementi finiti nel sub-cluster destro
<code>-NrTable</code>	- Numero di tavoli nel cluster
<code>-NrTableLeft</code>	- Numero di tavoli nel sub-cluster sinistro
<code>-NrTableRight</code>	- Numero di tavoli nel sub-cluster destro

7.3.3 Documentazione delle funzioni membro

7.3.3.1 `bool CategoricalCluster::IsEmpty () const [virtual]`

Controlla se il cluster non contiene elementi.

Restituisce

TRUE se il cluster e' vuoto FALSE se non lo e'

Implementa [GenericCluster< TypeCategorical, 1 >](#).

7.3.3.2 `void CategoricalCluster::ResetStatistics (unsigned int W) [virtual]`

Azzera le statistiche nel cluster.

Parametri

<code>W</code>	- dimensione della statistica con cui aggiornare gli iperparametri del parametro latente del cluster, numero di elementi distinti
----------------	---

Implementa [GenericCluster< TypeCategorical, 1 >](#).

7.3.3.3 `void CategoricalCluster::ResetStatisticsLeft (unsigned int W) [virtual]`

Azzera le statistiche nel sub-cluster sinistro.

Parametri

<code>W</code>	- dimensione della statistica con cui aggiornare gli iperparametri del parametro latente del sub-cluster sinistro, numero di elementi distinti
----------------	--

Implementa [GenericCluster< TypeCategorical, 1 >](#).

7.3.3.4 `void CategoricalCluster::ResetStatisticsRight (unsigned int W) [virtual]`

Azzera le statistiche nel sub-cluster destro.

Parametri

<i>W</i>	- dimensione della statistica con cui aggiornare gli iperparametri del parametro latente del sub-cluster destro, numero di elementi distinti
----------	--

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Fissa il peso del cluster.

Parametri

<i>_Beta</i>	- peso in ingresso
--------------	--------------------

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Fissa il peso del sub-cluster sinistro.

Parametri

<i>_BetaLeft</i>	- peso in ingresso
------------------	--------------------

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Fissa il peso del sub-cluster destro.

Parametri

<i>_BetaLeft</i>	- peso in ingresso
------------------	--------------------

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Fissa il numero di tavoli globali che caratterizza il cluster.

Parametri

<i>NrTable</i>	- numero di tavoli che caratterizza il cluster
----------------	--

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Fissa il numero di tavoli globali che caratterizza il sub-cluster sinistro.

Parametri

<i>NrTableLeft</i>	- numero di tavoli che caratterizza il sub-cluster sinistro
--------------------	---

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Fissa il numero di tavoli globali che caratterizza il sub-cluster destro.

Parametri

<i>NrTableLeft</i>	- numero di tavoli che caratterizza il sub-cluster destro
--------------------	---

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

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

Parametri

<i>_c</i>	- statistiche del cluster, conteggi degli elementi finiti nel cluster
-----------	---

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

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

Parametri

<i>_cLeft</i>	- statistiche del sub-cluster sinistro, conteggi degli elementi finiti nel sub-cluster sinistro
---------------	---

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

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

Parametri

<i>_cLeft</i>	- statistiche del sub-cluster destro, conteggi degli elementi finiti nel sub-cluster destro
---------------	---

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

Fissa il parametro latente del cluster.

Parametri

<i>_Theta</i>	- parametro latente in ingresso di tipo THETA
---------------	---

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

Fissa il parametro latente del sub-cluster sinistro.

Parametri

<i>_ThetaLeft</i>	- parametro latente in ingresso di tipo THETA
-------------------	---

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

Fissa il parametro latente del sub-cluster destro.

Parametri

<code>_ThetaRight</code>	- parametro latente in ingresso di tipo THETA
--------------------------	---

7.3.3.17 void CategoricalCluster::UpdateStatistics (STAT & counts4cleft, STAT & counts4right)

Aggiorna le statistiche, ovvero gli iperparametri dei parametri latenti dei cluster e sub-cluster.

Parametri

<code>counts4cleft</code>	- statistiche per aggiornare gli iperparametri dei parametri latenti del sub-cluster sinistro
<code>counts4cright</code>	- statistiche per aggiornare gli iperparametri dei parametri latenti del sub-cluster destro

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

Estra il peso globale del cluster.

Restituisce

Peso del cluster

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Estra il peso globale del sub-cluster sinistro.

Restituisce

Peso del sub-cluster sinistro

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Estra il peso globale del sub-cluster destro.

Restituisce

Peso del sub-cluster destro

Implementa [GenericCluster< TypeCategorical, 1 >](#).

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

Estra il numero globale dei tavoli nel cluster.

Restituisce

Numero di tavoli nel cluster

Implementa [GenericCluster< TypeCategorical, 1 >](#).

7.3.3.22 `unsigned int CategoricalCluster::ViewGlobalTableLeft () const [virtual]`

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

Restituisce

Numero di tavoli nel sub-cluster sinistro

Implementa [GenericCluster< TypeCategorical, 1 >](#).

7.3.3.23 `unsigned int CategoricalCluster::ViewGlobalTableRight () const [virtual]`

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

Restituisce

Numero di tavoli nel sub-cluster destrp

Implementa [GenericCluster< TypeCategorical, 1 >](#).

7.3.3.24 `void CategoricalCluster::ViewStatistics (STAT & _c) const`

Estra la statistica del parametro latente del cluster.

Parametri

<code>_c</code>	- oggetto di tipo STAT in cui viene memorizzato la statistica del parametro latente del cluster
-----------------	---

7.3.3.25 `void CategoricalCluster::ViewStatisticsLeft (STAT & _cLeft) const`

Estra la statistica del parametro latente del sub-cluster sinistro.

Parametri

<code>_cLeft</code>	- oggetto di tipo STAT in cui viene memorizzato la statistica del parametro latente del sub-cluster sinistro
---------------------	--

7.3.3.26 `void CategoricalCluster::ViewStatisticsRight (STAT & _cRight) const`

Estra la statistica del parametro latente del sub-cluster destro.

Parametri

<code>_cLeft</code>	- oggetto di tipo STAT in cui viene memorizzato la statistica del parametro latente del sub-cluster destro
---------------------	--

7.3.3.27 `void CategoricalCluster::ViewTheta (THETA & _Theta) const`

Estrae il parametro latente del cluster.

Parametri

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

7.3.3.28 `double CategoricalCluster::ViewThetaId (unsigned int _id) const`

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

Parametri

<i>Elemento</i>	da estrarre
-----------------	-------------

Restituisce

peso dell'elemento nella posizione indicata del cluster

7.3.3.29 `void CategoricalCluster::ViewThetaLeft (THETA & _ThetaLeft) const`

Estrae il parametro latente del sub-cluster sinistro.

Parametri

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

7.3.3.30 `double CategoricalCluster::ViewThetaLeftId (unsigned int _id) const`

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

Parametri

<i>Elemento</i>	da estrarre
-----------------	-------------

Restituisce

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

7.3.3.31 `void CategoricalCluster::ViewThetaRight (THETA & _ThetaRight) const`

Estrae il parametro latente del sub-cluster destro.

Parametri

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

7.3.3.32 `double CategoricalCluster::ViewThetaRightId (unsigned int _id) const`

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

Parametri

<i>Elemento</i>	da estrarre
-----------------	-------------

Restituisce

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

La documentazione per questa classe è stata generata a partire dal seguente file:

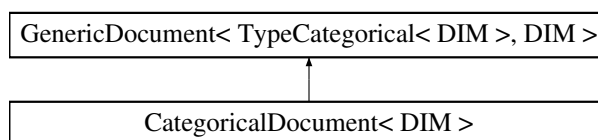
- [Cluster.hpp](#)

7.4 Template per la classe CategoricalDocument< DIM >

Classe derivata per il modello Dirichlet-Categorical.

```
#include <Document.hpp>
```

Diagramma delle classi per CategoricalDocument< DIM >

**Tipi pubblici**

- using **STAT** = [TypeCategorical< 1 >::STAT](#)
Statistiche per aggiornare gli iperparametri della distribuzione del parametro latente.
- using **THETA** = [TypeCategorical< 1 >::THETA](#)
Parametro latente, vettore di pesi delle parole distinte nel cluster.
- using **POINT** = [TypeCategorical< 1 >::Point](#)
Singolo dato.
- using **ClusterID** = unsigned int
Identificativo del cluster.

Membri pubblici

- [CategoricalDocument](#) (double _alpha)
Costruttore alternativo.
- [CategoricalDocument](#) ()=default
Costruttore di default.
- [~CategoricalDocument](#) ()
Distruttore.
- [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)
Aggiorna i pesi dei topic specifici del documento; si veda equazione (3.5) in Relazione_Parisi_Perego.pdf.
- void **UpdatePiSub** (const double _BetaLeft, const double _BetaRight, const **ClusterID** k, **omprng** &Gen)
Aggiorna i pesi, specifici del documento, dei subtopic del topic k; ; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.
- void **UpdateAllPiSub** (const vector< double > _BetaLeft, const vector< double > _BetaRight, **omprng** &Gen)
Aggiorna i pesi, specifici del documento, di tutti i subtopic, ; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.
- void **UpdateLocalTable** (const vector< long double > &_stirling, const vector< double > &_Beta, **omprng** &Gen)
Aggiorna i tavoli; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.
- void **UpdateLocalTableSub_OneCluster** (const vector< long double > &_stirling, const double _BetaLeft, const double _BetaRight, const **ClusterID** k, **omprng** &Gen)
Aggiorna i tavoli dei subtopic del topic k; si veda equazione (3.12) in Relazione_Parisi_Perego.pdf.
- void **UpdateAllLocalTableSub** (const vector< long double > &_stirling, const vector< double > &_BetaLeft, const vector< double > &_BetaRight, **omprng** &Gen)
Aggiorna i tavoli dei subtopic di tutti i topic; si veda equazione (3.12) in Relazione_Parisi_Perego.pdf.
- void **UpdateZeta** (const **THETA** &_Thetald, const **POINT** _VettId, **omprng** &Gen)
Aggiorna l'etichetta per il topic di una parola, campionata con il metodo Sampling; si veda equazione (3.7) in Relazione_Parisi_Perego.pdf.
- void **UpdateZeta_and_Sub** (const **THETA** &_Thetald, const **THETA** &_ThetaldLeft, const **THETA** &_ThetaldRight, const unsigned int _VettId, **omprng** &Gen)
Aggiorna l'etichetta per il topic e per il subtopic di una parola, campionate con il metodo Sampling si vedano equazioni (3.7) - (3.11) in Relazione_Parisi_Perego.pdf.
- void **UpdateZetaSub** (const **THETA** &_ThetaldLeft, const **THETA** &_ThetaldRight, const **POINT** id, const unsigned int nidjk, const **ClusterID** k, **omprng** &Gen)
Distribuisce la parola id nei subtopic del topic k, dopo aver campionato l'etichetta del subtopic con il metodo Sampling si veda equazione (3.11) in Relazione_Parisi_Perego.pdf.
- void **UpdateZeta** (const **ClusterID** _k)
Metodo che serve nelle mosse di M-H per rimuovere il topic k.
- void **UpdateZeta** (const **ClusterID** _k1, const **ClusterID** _k2)
Metodo che serve nelle mosse di M-H per rimuovere due topic.
- unsigned int **ViewNj** () const
Visualizza il numero di parole nel documento j.
- void **ViewData** (vector< **POINT** > &_VettId) const
- void **ViewCounts4c** (**ClusterID** _k, **STAT** &_counts4cleft, **STAT** &_counts4cright)
Visualizza i conteggi necessari per aggiornare i parametri latenti dei subtopic.
- unsigned int **ViewNumTableID** (const **ClusterID** _k) const
Estrae il numero di tavoli in uno specifico topic.
- unsigned int **ViewNumTableLeftID** (const **ClusterID** _k) const
Estrae il numero di tavoli del subtopic sinistro di uno specifico topic.
- unsigned int **ViewNumTableRightID** (const **ClusterID** _k) const
Estrae il numero di tavoli del subtopic destro di uno specifico topic.
- unsigned int **ViewDataCountID** (const **ClusterID** _k) const
Estrae il numero di parole nel topic k.
- unsigned int **ViewDataCountLeftID** (const **ClusterID** _k) const
Estrae il numero di parole nel subtopic sinistro del topic k.
- unsigned int **ViewDataCountRightID** (const **ClusterID** _k) const
Estrae il numero di parole nel subtopic destro del topic k.
- void **ResetDataCountSub** (const **ClusterID** k)
Azzera il conteggio delle parole nel topic k.
- void **ViewDataCount** (vector< unsigned int > &_WordCount) const
Estrae il vettore del numero di parole in ogni topic.

- void **ViewDataCountLeft** (vector< unsigned int > &_WordCountLeft) const
Estrae il vettore del numero di parole in ogni subtopic sinistro.
- void **ViewDataCountRight** (vector< unsigned int > &_WordCountRight) const
Estrae il vettore del numero di parole in ogni subtopic destro.
- void **ViewIdCounts** (vector< pair< **POINT**, unsigned int >> &_nidjk, const **ClusterID** _k)
Estrae identificativi e conteggi delle parole nel topic k.
- void **ViewCluster** (const **ClusterID** _k, pair< unordered_map< **POINT**, unsigned int >, unordered_map< **POINT**, unsigned int >> &_Cluster)
Estrae il topic k.
- double **ViewPiID** (const **ClusterID** _k) const
Estrae il peso, specifico del documento, del topic k.
- double **ViewPiLeftID** (const **ClusterID** _k) const
Estrae il peso, specifico del documento, del subtopic sinistro del topic k.
- double **ViewPiRightID** (const **ClusterID** _k) const
Estrae il peso, specifico del documento, del subtopic destro del topic k.
- void **ViewPi** (vector< double > &_pi) const
Estrae il vettore di pesi dei topic specifici dei documenti.
- void **ViewPiLeft** (vector< double > &_pi) const
Estrae il vettore di pesi dei subtopic sinistri specifici dei documenti.
- void **ViewPiRight** (vector< double > &_pi) const
Estrae il vettore di pesi dei subtopic destri specifici dei documenti.
- void **SetAlpha** (const double _alpha)
Imposta il parametro di concentrazione del processo di Dirichlet che governa il documento.
- void **SetNj** (const unsigned int _Nj)
Imposta il numero di parole nel documento j.
- void **SetPi** (vector< double > &_Pi)
Imposta il vettore di pesi dei topic specifici dei documenti.
- 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)
Inserisce un nuovo topic.
- void **RemoveCluster** (const vector< **ClusterID** > &_k)
Rimuove i topic con identificativo presente nel vettore in ingresso.
- void **RemoveCluster** (const **ClusterID** _k)
Rimuove un topic.
- void **RemoveCluster** (**ClusterID** _k1, **ClusterID** _k2)
Rimuove due topic.
- unsigned int **CheckLeftSubcluster** (const **ClusterID** _k)
Verifica se un topic ha il subtopic sinistro vuoto.
- unsigned int **CheckRightSubcluster** (const **ClusterID** _k)
Verifica se un topic ha il subtopic destro vuoto.
- void **ViewLabel** (vector< pair< **POINT**, **ClusterID** >> &Data)
Estrae le etichette associate alle parole.
- void **SetDataset** (std::istream &SSTR)
Acquisisce i dati.
- unsigned int **SortData** (unsigned int _K, **omprng** &Gen)
Smista le parole nel contenitore Zeta.

Membri privati

- void [UpdateDataCount](#) ()
Aggiorna i conteggi delle parole nei topic.
- void [Sampling](#) (std::vector< unsigned int > &_temp_counts, std::vector< double > &_Weights, unsigned int _nidj, [omprng](#) &Gen)
Campionamento da distribuzione categorica per l'etichetta del topic o del subtopic.

Attributi privati

- unordered_map< [ClusterID](#), pair< unordered_map< [POINT](#), unsigned int >, unordered_map< [POINT](#), unsigned int > > > [Zeta](#)
Contenitore dei dati. Per ogni cluster ho il subcluster sinistro e destro: nella mappa la chiave é il dato e il valore mappato é il numero di volte che il dato compare in quel documento, in quel cluster, in quel subcluster.
- unordered_map< [POINT](#), unsigned int > [Vocabulary](#)
Vocabolario delle parole distinte del documento La chiave nella mappa e' il dato, il valore mappato é il numero di volte che il dato compare nel documento.
- double [alpha](#)
lperparametro del processo di Dirichlet che governa il documento.
- unsigned int [Nj](#)
Numero di parole contenute nel documento.
- vector< double > [Pi](#)
Vettore di pesi del cluster specifici del documento.
- vector< double > [PiLeft](#)
Vettore di pesi del subcluster sinistro specifici del documento.
- vector< double > [PiRight](#)
Vettore di pesi del subcluster destro specifici del documento.
- vector< unsigned int > [WordCount](#)
Vettore di conteggi di dimensione K: l'elemento in posizione k indica il numero di dati del documento j nel cluster k.
- vector< unsigned int > [WordCountLeft](#)
Vettore di conteggi di dimensione K: l'elemento in posizione k indica il numero di dati del documento j nel subcluster sinistro del cluster k.
- vector< unsigned int > [WordCountRight](#)
Vettore di conteggi di dimensione K: l'elemento in posizione k indica il numero di dati del documento j nel subcluster destro del cluster k.
- vector< unsigned int > [LocalTable](#)
Vettore dei tavoli di dimensione K : l'elemento in posizione k indica il numero di tavoli nel ristorante j che servono il piatto k.
- vector< unsigned int > [LocalTableLeft](#)
Vettore dei tavoli di dimensione K : l'elemento in posizione k indica il numero di tavoli nel ristorante j che servono il piatto k left.
- vector< unsigned int > [LocalTableRight](#)
Vettore dei tavoli di dimensione K : l'elemento in posizione k indica il numero di tavoli nel ristorante j che servono il piatto k right.

7.4.1 Descrizione dettagliata

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

Classe derivata per il modello Dirichlet-Categorical.

Classe che rappresenta un documento per il problema del topic modeling. Gestisce le parole e si occupa del campionamento delle etichette per il topic a cui assegnare ogni parola. Gestisce i parametri del modello specifici del documento: $\alpha, \pi_j, \bar{\pi}_{jl}, \bar{\pi}_{jr}, m_j, \bar{m}_{jl}, \bar{m}_{jr}$; si occupa del campionamento di queste quantità. Tiene traccia dei conteggi delle parole nei topic

Data

Febbraio 2016

7.4.2 Documentazione dei costruttori e dei distruttori

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

Costruttore alternativo.

Parametri

<i>_alpha</i>	- iperparametro del processo di Dirichlet che governa il documento
---------------	--

7.4.3 Documentazione delle funzioni membro

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

Verifica se un topic ha il subtopic sinistro vuoto.

Parametri

<i>_k</i>	- id del topic di cui controllare il subtopic
-----------	---

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

Verifica se un topic ha il subtopic destro vuoto.

Parametri

<i>_k</i>	- id del topic di cui controllare il subtopic
-----------	---

7.4.3.3 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*)

Inserisce un nuovo topic.

Parametri

<i>NewCluster</i>	- il nuovo topic
<i>_Pi</i>	- peso del nuovo topic specifico del documento
<i>_PiLeft</i>	- peso del subtopic sinistro nuovo topic, specifico del documento
<i>_PiRight</i>	- peso del subtopic destro del nuovo topic, specifico del documento
<i>_WordCount</i>	- numero di parole nel nuovo topic
<i>_WordCountLeft</i>	- numero di parole nel subtopic sinistro del nuovo topic
<i>_WordCount- Right</i>	- numero di parole nel subtopic destro del nuovo topic
<i>_LocalTable</i>	- numero di tavoli che servono il nuovo piatto nel ristorante j
<i>_LocalTableLeft</i>	- numero di tavoli che servono il nuovo piatto left nel ristorante j
<i>_LocalTable</i>	- numero di tavoli che servono il nuovo piatto right nel ristorante j

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

Rimuove i topic con identificativo presente nel vettore in ingresso.

Parametri

<i>_k</i>	- vettore con gli id dei topic da eliminare
-----------	---

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

7.4.3.5 `template<unsigned int DIM> void CategoricalDocument< DIM >::RemoveCluster (const ClusterID _k)`

Rimuove un topic.

Parametri

<i>_k</i>	- id del topic da eliminare
-----------	-----------------------------

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

Rimuove due topic.

Parametri

<i>_k1</i>	- id del topic da eliminare
<i>_k2</i>	- id del topic da eliminare

7.4.3.7 `template<unsigned int DIM> void CategoricalDocument< DIM >::ResetDataCountSub (const ClusterID k)`

Azzera il conteggio delle parole nel topic k.

Parametri

<i>k</i>	- id del topic
----------	----------------

7.4.3.8 `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]`

Campionamento da distribuzione categorica per l'etichetta del topic o del subtopic.

Parametri

<code>_temp_counts</code>	- vettore che contiene i conteggi della parola id nei topic
<code>_Weights</code>	- pesi con cui campionare le etichette
<code>_nidj</code>	- numero di volte che la parola id compare nel documento j
<code>Gen</code>	- generatore di numeri casuali

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Imposta il parametro di concentrazione del processo di Dirichlet che governa il documento.

Parametri

<code>_alpha</code>	- parametro di concentrazione del processo di Dirichlet che governa il documento
---------------------	--

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

7.4.3.10 `template<unsigned int DIM> void CategoricalDocument< DIM >::SetDataset (std::istream & SSTR)`
[virtual]

Acquisisce i dati.

Parametri

<code>SSTR</code>	- contiene id della parola e numero di volte che parola compare nel documento
-------------------	---

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Imposta il numero di parole nel documento j.

Parametri

<code>_Nj</code>	- numero di parole nel documento j
------------------	------------------------------------

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Smista le parole nel contenitore Zeta.

Parametri

<code>_K</code>	- numero iniziale di topic
<code>Gen</code>	- generatore di numeri casuali

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Aggiorna i tavoli dei subtopic di tutti i topic; si veda equazione (3.12) in *Relazione_Parisi_Perego.pdf*.

Parametri

<code>_stirling</code>	- numeri di stirling
<code>_BetaLeft</code>	- pesi globali dei subtopic sinistri
<code>_BetaRight</code>	- pesi globali dei subtopic destri
<code>Gen</code>	- generatore di numeri casuali in parallelo

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Aggiorna i pesi, specifici del documento, di tutti i subtopic, ; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_BetaLeft</code>	- pesi globali dei subtopic sinistri
<code>_BetaRight</code>	- pesi globali dei subtopic destri
<code>Gen</code>	- generatore di numeri casuali in parallelo

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Aggiorna i tavoli; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_stirling</code>	- numeri di stirling
<code>_Beta</code>	- pesi globali dei topic
<code>Gen</code>	- generatore di numeri casuali in parallelo

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

7.4.3.16 `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)`

Aggiorna i tavoli dei subtopic del topic k; si veda equazione (3.12) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_stirling</code>	- numeri di stirling
<code>_BetaLeft</code>	- peso globale del subtopic sinistro del topic k
<code>_BetaRight</code>	- peso globale del subtopic destro del topic k
<code>k</code>	- id del topic
<code>Gen</code>	- generatore di numeri casuali in parallelo

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

Aggiorna i pesi dei topic specifici del documento; si veda equazione (3.5) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_AllBeta</code>	- pesi globali dei topic
<code>Gen</code>	- generatore di numeri casuali in parallelo

Implementa `GenericDocument< TypeCategorical< DIM >, DIM >`.

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

Aggiorna i pesi, specifici del documento, dei subtopic del topic k; ; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_BetaLeft</code>	- peso globale del subtopic sinistro del topic k
<code>_BetaRight</code>	- peso globale del subtopic destro del topic k
<code>k</code>	- id del topic
<code>Gen</code>	- generatore di numeri casuali in parallelo

7.4.3.19 `template<unsigned int DIM = 1> void CategoricalDocument< DIM >::UpdateZeta (const THETA & _Thetald, const POINT _Vettld, omprng & Gen)`

Aggiorna l'etichetta per il topic di una parola, campionata con il metodo Sampling; si veda equazione (3.7) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_Thetald</code>	- vettore dei pesi della parola id in tutti i topic
<code>_Vettld</code>	- id della parola
<code>Gen</code>	- generatore di numeri casuali in parallelo

7.4.3.20 `template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateZeta (const ClusterID _k)`

Metodo che serve nelle mosse di M-H per rimuovere il topic k.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

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

Metodo che serve nelle mosse di M-H per rimuovere due topic.

Parametri

<code>_k1</code>	- id del topic
<code>_k2</code>	- id del topic

7.4.3.22 `template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateZeta_and_Sub (const THETA & _Thetald, const THETA & _ThetaldLeft, const THETA & _ThetaldRight, const unsigned int _Vettld, omprng & Gen)`

Aggiorna l'etichetta per il topic e per il subtopic di una parola, campionate con il metodo Sampling si vedano equazioni (3.7) - (3.11) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_Thetald</code>	- vettore dei pesi della parola id in tutti i topic
<code>_ThetaldLeft</code>	- vettore dei pesi della parola id in tutti i subtopic sinistri
<code>_ThetaldRight</code>	- vettore dei pesi della parola id in tutti i subtopic destri
<code>_Vetld</code>	- id della parola
<code>Gen</code>	- generatore di numeri casuali in parallelo

7.4.3.23 `template<unsigned int DIM> void CategoricalDocument< DIM >::UpdateZetaSub (const THETA & _ThetaldLeft, const THETA & _ThetaldRight, const POINT id, const unsigned int nidjk, const ClusterID k, omprng & Gen)`

Distribuisce la parola id nei subtopic del topic k, dopo aver campionato l'etichetta del subtopic con il metodo Sampling si veda equazione (3.11) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_ThetaldLeft</code>	- vettore dei pesi della parola id in tutti i subtopic sinistri
<code>_ThetaldRight</code>	- vettore dei pesi della parola id in tutti i subtopic destri
<code>id</code>	- id della parola
<code>nidjk</code>	- numero di volte che la parola id nel documento j e' capitata nel topic k
<code>k</code>	- id del topic
<code>Gen</code>	- generatore di numeri casuali in parallelo

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

Estrae il topic k.

Parametri

<code>_k</code>	- id del topic da estrarre
<code>_Cluster</code>	- struttura in cui viene estratto il topic

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

Visualizza i conteggi necessari per aggiornare i parametri latenti dei subtopic.

Parametri

<code>_k</code>	- id del topic
<code>_counts4cleft</code>	- conteggi per parametro latente del subtopic sinistro
<code>_counts4cright</code>	- conteggi per parametro latente del subtopic destro

7.4.3.26 `template<unsigned int DIM> void CategoricalDocument< DIM >::ViewData (vector< POINT > & _Vetld) const`

Estrae gli identificativi delle parole nel documento j

Parametri

<code>_VettId</code>	- viene riempito con gli id delle parole nel documento j
----------------------	--

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

Estrae il vettore del numero di parole in ogni topic.

Parametri

<code>_WordCount</code>	- vettore del numero di parole in ogni topic
-------------------------	--

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

7.4.3.28 `template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewDataCountID (const ClusterID _k) const`

Estrae il numero di parole nel topic k.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Numero di parole nel topic k

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

Estrae il vettore del numero di parole in ogni subtopic sinistro.

Parametri

<code>_WordCountLeft</code>	- vettore del numero di parole in ogni subtopic sinistro
-----------------------------	--

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

7.4.3.30 `template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewDataCountLeftID (const ClusterID _k) const`

Estrae il numero di parole nel subtopic sinistro del topic k.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Numero di parole nel subtopic sinistro del topic k

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

Estrae il vettore del numero di parole in ogni subtopic destro.

Parametri

<code>_WordCount-Right</code>	- vettore del numero di parole in ogni subtopic destro
-------------------------------	--

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

7.4.3.32 `template<unsigned int DIM> unsigned int CategoricalDocument< DIM >::ViewDataCountRightID (const ClusterID _k) const`

Estrae il numero di parole nel subtopic destro del topic k.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Numero di parole nel subtopic destro del topic k

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

Estrae identificativi e conteggi delle parole nel topic k.

Parametri

<code>_nidjk</code>	- struttura che contiene identificativi e conteggi delle parole nel topic k
<code>_k</code>	- id del topic

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

Estrae le etichette associate alle parole.

Parametri

<code>Data</code>	- struttura in cui estrarre le etichette
-------------------	--

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

Visualizza il numero di parole nel documento j.

Restituisce

Numero di parole nel documento j

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Estrae il numero di tavoli in uno specifico topic.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Numero di tavoli nel ristorante j che servono il piatto k

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

Estrae il numero di tavoli del subtopic sinistro di uno specifico topic.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Numero di tavoli nel ristorante j che servono il piatto k left

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

Estrae il numero di tavoli del subtopic destro di uno specifico topic.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Numero di tavoli nel ristorante j che servono il piatto k destro

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

Estrae il vettore di pesi dei topic specifici dei documenti.

Parametri

<code>_pi</code>	- vettore di pesi dei topic specifici dei documenti
------------------	---

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Estrae il peso, specifico del documento, del topic k.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Peso specifico del documento del topic k

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

Estrae il vettore di pesi dei subtopic sinistri specifici dei documenti.

Parametri

<code>_pi</code>	- vettore di pesi dei subtopic sinistri specifici dei documenti
------------------	---

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Estrae il peso, specifico del documento, del subtopic sinistro del topic k.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Peso specifico del documento del subtopic sinistro del topic k

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

Estrae il vettore di pesi dei subtopic destri specifici dei documenti.

Parametri

<code>_pi</code>	- vettore di pesi dei subtopic destri specifici dei documenti
------------------	---

Implementa [GenericDocument< TypeCategorical< DIM >, DIM >](#).

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

Estrae il peso, specifico del documento, del subtopic destro del topic k.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Peso specifico del documento del subtopic destro del topic k

La documentazione per questa classe è stata generata a partire dal seguente file:

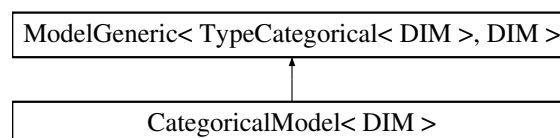
- [Document.hpp](#)

7.5 Template per la classe CategoricalModel< DIM >

Modello Dirichlet-Categorical.

```
#include <Model.hpp>
```

Diagramma delle classi per CategoricalModel< DIM >



Tipi pubblici

- using **THETA** = **TypeCategorical**< 1 >::**THETA**
Parametro latente, vettore di pesi degli elementi distinti nel cluster.
- using **Point** = **TypeCategorical**< 1 >::**Point**
Singlo dato, dati ripetuti.
- using **STAT** = **TypeCategorical**< 1 >::**STAT**
Statistiche per aggiornare l'iperparametro del parametro latente, numero di dati che sono contenuti nel cluster e sub-cluster.
- using **HYP** = **TypeCategorical**< 1 >::**HYP**
Iperparametro del parametro latente.

Membri pubblici

- **CategoricalModel** ()
Costruttore di default.
- **CategoricalModel** (const **CategoricalModel** &mod)=default
Copy constructor.
- **CategoricalModel** (const **CategoricalModel** &&mod)
Move constructor.
- **~CategoricalModel** ()=default
Distruttore di default.
- **CategoricalCluster** & **operator[]** (unsigned int _K)
Access operator per gli l-value.
- **CategoricalModel** & **operator=** (const **CategoricalModel** &mod)
Assignement operator.
- **CategoricalModel** & **operator=** (**CategoricalModel** &&mod)
Move assignement operator.
- void **SetHyperparameter** (const **HYP** &_Lambda)
Fissa gli iperparametri dei parametri latenti.
- void **DefaultHyperparameter** (size_t W)
Fissa gli iperparametri dei parametri latenti con i valori di default.
- void **SetInitialClusters** (unsigned int _K)
Fissa i cluster iniziali, assegnando i pesi globali iniziali.
- double **Marginalized_Loglikelihood** (const unsigned int _K)
Calcola la verosimiglianza logaritmica marginale del cluster, una volta specificato l'id del clusters.
- double **Loglikelihood** (const **POINT** X, const unsigned int _K)
Calcola la loglikelihood in un punto X una volta precisato l'id del cluster.
- double **LoglikelihoodLeft** (const **POINT** X, const unsigned int _K)
Calcola la loglikelihood in un punto X una volta precisato l'id del sub-cluster sinistro.
- double **LoglikelihoodRight** (const **POINT** X, const unsigned int _K)
Calcola la loglikelihood in un punto X una volta precisato l'id del sub-cluster destro.
- long double **LogDensity** (const unsigned int _K)
Calcola la densita' del parametro latente che caratterizza il cluster specificato.
- void **UpdateThetaCluster** (**omprng** &Gen)
Aggiornamento dei parametro latenti di tutti i clusters.
- void **UpdateThetaSubCluster** (**omprng** &Gen)
Aggiornamento dei parametro latenti del sub-cluster sinistro e destro di tutti i clusters.
- void **UpdateOneThetaCluster** (const unsigned int _K, **omprng** &Gen)
Aggiornamento dei parametro latenti di un cluster, una volta precisato il suo id.
- void **UpdateOneThetaSubCluster** (const unsigned int _K, **omprng** &Gen)

Aggiornamento dei parametro latenti del sub-cluster sinistro e destro di un cluster, una volta precisato il suo id.

- void [AddOneCluster](#) (const unsigned int _k)
Aggiunge un cluster vuoto ai clusters presenti ed aggiorna il parametro K, che identifica il numero di cluster attuali. I parametri del cluster devono essere specificati in un secondo momento.
- void [RemoveOneCluster](#) (const unsigned int _K)
Rimuove un cluster una volta specificato l'id, ed aggiorna il parametro K, che identifica il numero di cluster attuali.
- void [RemoveClusters](#) (const vector< unsigned int > &_K)
Rimuove piu' clusters una volta che sono specificati gli ID, ed aggiorna il parametro K, che identifica il numero di cluster attuali.
- unsigned int [ViewK](#) () const
Da informazione sul numero di clusters correnti.
- void [ViewKey](#) (vector< unsigned int > &Key) const
Visualizza gli ID dei clusters attualmente presenti.
- void [ViewBeta](#) (vector< double > &_AllBeta)
Mostra i pesi globali dei cluster attualmente presenti.
- void [ViewBetaLeft](#) (vector< double > &_AllBetaLeft)
Mostra i pesi globali del sub-cluster sinistro dei clusters attualmente presenti.
- void [ViewBetaRight](#) (vector< double > &_AllBetaRight)
Mostra i pesi globali del sub-cluster destro dei clusters attualmente presenti.
- void [PrintTheta](#) ()
Stampa su file i valori dei parametri latenti dei cluster attualmente presenti, i pesi degli elementi distinti in ogni clusters.
- void [PrintLambdaInfo](#) () const

Attributi privati

- [HYP Lambda](#)
Vettore degli iperparametri.
- unsigned int [K](#)
numero corrente di cluster
- unordered_map< unsigned int,
[CategoricalCluster](#) > [Clusters](#)
Insieme degli oggetti di tipo CategoricalCluster, individuati in base al suo Id.
- unsigned int [OMP_NUM_THREADS](#)

7.5.1 Descrizione dettagliata

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

Modello Dirichlet-Categorical.

Verosimiglianza: Categorical. Prior sui parametri latenti: Dirichlet

Questa classe e' impiegata per campionare i parametri latenti da una Dirichlet, per il calcolo delle verosimiglianze e delle marginali per un modello Categorical. Gestisce anche l'aggiunta e la rimozione dei clusters, la stampa su file dei parametri latenti.

Data

Febbraio 2016

7.5.2 Documentazione delle funzioni membro

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

Aggiunge un cluster vuoto ai clusters presenti ed aggiorna il parametro K, che identifica il numero di cluster attuali. I parametri del cluster devono essere specificati in un secondo momento.

Parametri

<code>_K</code>	- Id del nuovo cluster che si aggiunge.
-----------------	---

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Fissa gli iperparametri dei parametri latenti con i valori di default.

Parametri

<code>W</code>	- dimensione degli iperparametri
----------------	----------------------------------

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Calcola la densita' del parametro latente che caratterizza il cluster specificato.

Parametri

<code>_K</code>	- Id del cluster
-----------------	------------------

Restituisce

Densita' logaritmica del cluster `_K`

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Calcola la loglikelihood in un punto X una volta precisato l'id del cluster.

Parametri

<code>X</code>	- punto di valutazione
<code>_K</code>	- id del cluster

Restituisce

verosimiglianza logaritmica valutante nel punto X del cluster `_K`

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

Calcola la loglikelihood in un punto X una volta precisato l'id del sub-cluster sinistro.

Parametri

<code>X</code>	- punto di valutazione
----------------	------------------------

<code>_K</code>	- id del sub-cluster sinistro
-----------------	-------------------------------

Restituisce

verosmiglianza logaritmica valutante nel punto X del sub-cluster sinistro del cluster `_K`

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

Calcola la loglikelihood in un punto X una volta precisato l'id del sub-cluster destro.

Parametri

<code>X</code>	- punto di valutazione
<code>_K</code>	- id del sub-cluster destro

Restituisce

verosmiglianza logaritmica valutante nel punto X del sub-cluster destro del cluster `_K`

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

Calcola la verosmiglianza logaritmica marginale del cluster, una volta specificato l'id del clusters.

Parametri

<code>_K</code>	- id del cluster di cui si vuole calcolare la verosmiglianza logaritmica marginale
-----------------	--

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

7.5.2.8 `template<unsigned int DIM> CategoricalModel< DIM > & CategoricalModel< DIM >::operator= (const CategoricalModel< DIM > & mod)`

Assignment operator.

Parametri

<code>mod</code>	- oggetto di tipo CategoricalModel
------------------	--

7.5.2.9 `template<unsigned int DIM> CategoricalModel< DIM > & CategoricalModel< DIM >::operator= (CategoricalModel< DIM > && mod)`

Move assignment operator.

Parametri

<code>mod</code>	- oggetto di tipo CategoricalModel
------------------	--

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

Access operator per gli l-value.

Parametri

<code>_K</code>	- a quale cluster si vuole accedere
-----------------	-------------------------------------

7.5.2.11 `template<unsigned int DIM> void CategoricalModel< DIM >::PrintLambdaInfo () const [virtual]`

Stampa a video le informazioni degli iperparametri dei parametri latenti dei cluster

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Rimuove piu' clusters una volta che sono specificati gli ID, ed aggiorna il parametro K, che identifica il numero di cluster attuali.

Parametri

<code>_K</code>	- Vettore degli Id dei clusters che si vogliono rimuovere
-----------------	---

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Rimuove un cluster una volta specificato l'id, ed aggiorna il parametro K, che identifica il numero di cluster attuali.

Parametri

<code>_K</code>	- Id del cluster che si vuole rimuovere
-----------------	---

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Fissa gli iperparametri dei parametri latenti.

Parametri

<code>_Lambda</code>	- Iperparametri in ingresso
----------------------	-----------------------------

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

Fissa i cluster iniziali, assegnando i pesi globali iniziali.

Parametri

<code>_K</code>	- Numero di cluster iniziali
-----------------	------------------------------

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

7.5.2.16 `template<unsigned int DIM> void CategoricalModel< DIM >::UpdateOneThetaCluster (const unsigned int _K, omprng & Gen) [virtual]`

Aggiornamento dei parametro latenti di un cluster, una volta precisato il suo id.

Equazione (3.6) della relazione Relazione_Parisi_Perego.pdf

Parametri

<code>_K</code>	- id del cluster di cui si vuole aggiornare parametri latenti
<code>Gen</code>	- Generatore dei numeri casuali

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

7.5.2.17 `template<unsigned int DIM> void CategoricalModel< DIM >::UpdateOneThetaSubCluster (const unsigned int _K, omprng & Gen) [virtual]`

Aggiornamento dei parametro latenti del sub-cluster sinistro e destro di un cluster, una volta precisato il sui id.

Equazione (3.10) della relazione Relazione_Parisi_Perego.pdf

Parametri

<code>_K</code>	- id del cluster di cui si vuole aggiornare i parametri latenti nei sui sub-clusters
<code>Gen</code>	- Generatore dei numeri casuali

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

7.5.2.18 `template<unsigned int DIM> void CategoricalModel< DIM >::UpdateThetaCluster (omprng & Gen) [virtual]`

Aggiornamento dei parametro latenti di tutti i clusters.

Equazione (3.6) della relazione Relazione_Parisi_Perego.pdf

Parametri

<code>Gen</code>	- Generatore dei numeri casuali
------------------	---------------------------------

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

7.5.2.19 `template<unsigned int DIM> void CategoricalModel< DIM >::UpdateThetaSubCluster (omprng & Gen) [virtual]`

Aggiornamento dei parametro latenti del sub-cluster sinistro e destro di tutti i clusters.

Equazione (3.10) della relazione Relazione_Parisi_Perego.pdf

Parametri

<code>Gen</code>	- Generatore dei numeri casuali
------------------	---------------------------------

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Mostra i pesi globali dei cluster attualmente presenti.

Parametri

<code>AllBeta</code>	- vettore riempito con i pesi globali dei clusters
----------------------	--

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Mostra i pesi globali del sub-cluster sinistro dei clusters attualmente presenti.

Parametri

<i>AllBetaLeft</i>	- vettore riempito con i pesi globali del sub-cluster sinistro dei clusters
--------------------	---

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Mostra i pesi globali del sub-cluster destro dei clusters attualmente presenti.

Parametri

<i>AllBetaRight</i>	- vettore riempito con i pesi globali del sub-cluster destro dei clusters
---------------------	---

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Da informazione sul numero di clusters correnti.

Restituisce

il numero corrente di clusters

Implementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

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

Visualizza gli ID dei clusters attualmente presenti.

Parametri

<i>Key</i>	- oggetto dove salvare gli Id dei clusters
------------	--

Reimplementa [ModelGeneric< TypeCategorical< DIM >, DIM >](#).

7.5.3 Documentazione dei membri dato

7.5.3.1 `template<unsigned int DIM = 1> unsigned int CategoricalModel< DIM >::OMP_NUM_THREADS [private]`

Numero di threads

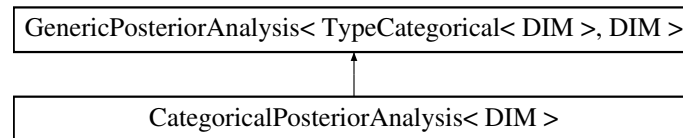
La documentazione per questa classe è stata generata a partire dal seguente file:

- [Model.hpp](#)

7.6 Template per la classe CategoricalPosteriorAnalysis< DIM >

```
#include <PosteriorAnalysis.hpp>
```

Diagramma delle classi per CategoricalPosteriorAnalysis< DIM >



Tipi pubblici

- using **THETA** = **TypeCategorical**< 1 >::**THETA**
Parametro latente, vettore di pesi delle parole distinte nel cluster.
- using **ClusterId** = unsigned int
Identificativo del cluster.
- using **GroupId** = unsigned int
Identificativo del documento.
- using **DataId** = unsigned int
Identificativo del dato.
- using **Check** = unsigned int
Variabile di controllo.

Membri pubblici

- **CategoricalPosteriorAnalysis** ()=default
Costruttore di default.
- **~CategoricalPosteriorAnalysis** ()=default
Distruttore di default.
- unsigned int **SetAllK** ()
Imposta la catena AllK.
- unsigned int **SetAllAlpha** ()
Imposta la catena Alpha.
- unsigned int **SetAllGamma** ()
Imposta la catena Gamma.
- void **SetVocabulary** ()
Imposta il vocabolario.
- void **VisualizeBeta** (RInside &R, unsigned long BestClustering)
Chiama lo script R che visualizza le traiettorie della distribuzione dei topic nel corpus.
- void **KPosteriorAnalysis** (RInside &R, const unsigned int Burnin, const unsigned int Thinning)
Chiama lo script R per l'analisi della catena dei K.
- 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)
Chiama lo script R per l'analisi delle catene Alpha e Gamma.
- void **Setwd** (const std::string &_wd)
Imposta la working directory di R.
- void **SetW** (const unsigned int _W)
Imposta il numero di parole distinte.
- void **SetD** (const unsigned int _D)
Imposta il numero di documenti.
- void **SetN** (const unsigned int _N)
Imposta il numero totale di parole.
- void **SetTheta** (const unsigned long MaxIt)

- Imposta la ricerca dei cluster, acquisendo da file i e i delle ultime iterazioni.*
- void [LoadLabels](#) (const unsigned long NrClusterings)
Acquisisce le etichette assegnate alle parole nelle ultime iterazioni.
- unsigned long [LeastSquareClustering](#) ()
Individua il miglior clustering secondo il criterio dei minimi quadrati; si veda equazione (4.3) in Relazione_Parisi_Perego.pdf.
- void [LPML](#) (unsigned long MaxIt)
Calcola e stampa a video l'indice LPML; si vedano equazioni (4.1)- (4.2) in Relazione_Parisi_Perego.pdf.
- void [TrackingClusters](#) (RInside &R)
Effettua il riconoscimento dei topic.

Attributi privati

- unordered_map< unsigned int, vector< tuple< [GroupId](#), [Check](#), double, vector< double > > > > [Theta](#)
Struttura in cui memorizzare le informazioni sui topic riconosciuti. La chiave della mappa e' l'iterazione, il valore mappato e' il blocco di topic individuati a quella iterazione. Per ogni topic nel blocco si ha: l'etichetta assegnata al topic, il numero di parole in comune con i topic di altri blocchi aventi la stessa etichetta, il peso globale del topic, il vettore di pesi delle parole che rappresenta il topic.
- vector< vector< double > > [AllBeta](#)
Memorizza i vettori delle ultime iterazioni.
- unordered_map< [DataId](#), std::string > [Vocabulary](#)
Vocabolario del corpus.
- vector< double > [Alpha](#)
Contiene i valori assunti da ad ogni iterazione, nel caso di prior su .
- vector< double > [Gamma](#)
Contiene i valori assunti da ad ogni iterazione, nel caso di prior su .
- vector< unsigned int > [AllK](#)
Contiene il numero di cluster dedotti dall'algorithm ad ogni iterazione.
- std::string [wd](#)
Working directory per R.
- unsigned int [W](#)
Numero di parole distinte.
- unsigned int [N](#)
Numero di parole totali.
- unsigned int [D](#)
Numero di documenti.
- vector< vector< unsigned int > > [Labels](#)
Contiene le etichette assegnate alle parole ad ogni iterazione.
- deque< double > [Pairwise_probabilities](#)
Matrice necessaria per individuare il least square clustering L'elemento in posizione (i,j) e' una stima Monte Carlo della probabilita' che la parola i sia nello stesso topic della parola j.

7.6.1 Descrizione dettagliata

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

Classe per l'analisi a posteriori specifica del modello Dirichle-Categorical Legge e memorizza i risultati delle simulazioni in opportune strutture.

7.6.2 Documentazione delle funzioni membro

7.6.2.1 `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]`

Chiama lo script R per l'analisi delle catene Alpha e Gamma.

Parametri

<i>R</i>	- Istanza di R
<i>AlphaBurnin</i>	- numero di valori iniziali della catena Alpha da scartare
<i>AlphaThinning</i>	- tiene un valore della catena Alpha ogni AlphaThinning valori
<i>GammaBurnin</i>	- numero di valori iniziali della catena Gamma da scartare
<i>GammaThinning</i>	- tiene un valore della catena Gamma ogni GammaThinning valori
<i>AlphaTry</i>	- yes se si vuole ripetere l'analisi della catena Alpha, no altrimenti
<i>GammaTry</i>	- yes se si vuole ripetere l'analisi della catena Gamma, no altrimenti

Implementa [GenericPosteriorAnalysis](#)< [TypeCategorical](#)< DIM >, DIM >.

7.6.2.2 `template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::KPosteriorAnalysis (Rinside & R, const unsigned int Burnin, const unsigned int Thinning) [virtual]`

Chiama lo script R per l'analisi della catena dei K.

Parametri

<i>R</i>	- Istanza di R
<i>Burnin</i>	- numero di valori iniziali della catena da scartare
<i>Thinning</i>	- tiene un valore della catena ogni thinning valori

Implementa [GenericPosteriorAnalysis](#)< [TypeCategorical](#)< DIM >, DIM >.

7.6.2.3 `template<unsigned int DIM> unsigned long CategoricalPosteriorAnalysis< DIM >::LeastSquareClustering () [virtual]`

Individua il miglior clustering secondo il criterio dei minimi quadrati; si veda equazione (4.3) in Relazione_Parisi_Perego.pdf.

Restituisce

Iterazione a cui e' stato individuato il miglior clustering

Implementa [GenericPosteriorAnalysis](#)< [TypeCategorical](#)< DIM >, DIM >.

7.6.2.4 `template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::LoadLabels (const unsigned long NrClusterings) [virtual]`

Acquisisce le etichette assegnate alle parole nelle ultime iterazioni.

Parametri

<i>NrClusterings</i>	- numero di iterazioni da monitorare
----------------------	--------------------------------------

Implementa [GenericPosteriorAnalysis](#)< [TypeCategorical](#)< DIM >, DIM >.

7.6.2.5 `template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::LPML (unsigned long MaxIt) [virtual]`

Calcola e stampa a video l'indice LPML; si vedano equazioni (4.1)- (4.2) in Relazione_Parisi_Perego.pdf.

Parametri

<i>MaxIt</i>	- numero di iterazioni da considerare nel calcolo di LPML
--------------	---

Implementa [GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >](#).

7.6.2.6 `template<unsigned int DIM> unsigned int CategoricalPosteriorAnalysis< DIM >::SetAllAlpha ()`
[virtual]

Imposta la catena Alpha.

Restituisce

Lunghezza della catena

Implementa [GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >](#).

7.6.2.7 `template<unsigned int DIM> unsigned int CategoricalPosteriorAnalysis< DIM >::SetAllGamma ()`
[virtual]

Imposta la catena Gamma.

Restituisce

Lunghezza della catena

Implementa [GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >](#).

7.6.2.8 `template<unsigned int DIM> unsigned int CategoricalPosteriorAnalysis< DIM >::SetAllK ()` [virtual]

Imposta la catena AllK.

Restituisce

Lunghezza della catena

Implementa [GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >](#).

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

Imposta il numero di documenti.

Parametri

<i>_D</i>	- numero di documenti
-----------	-----------------------

Implementa [GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >](#).

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

Imposta il numero totale di parole.

Parametri

<code>_N</code>	- numero totale di parole
-----------------	---------------------------

Implementa `GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >`.

```
7.6.2.11 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::SetTheta ( const unsigned long MaxIt
) [virtual]
```

Imposta la ricerca dei cluster, acquisendo da file i e i delle ultime iterazioni.

Parametri

<code>MaxIt</code>	- numero di iterazioni dell'algoritmo
--------------------	---------------------------------------

Implementa `GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >`.

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

Imposta il numero di parole distinte.

Parametri

<code>_W</code>	- numero di parole distinte
-----------------	-----------------------------

Implementa `GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >`.

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

Imposta la working directory di R.

Parametri

<code>_wd</code>	- working directory di R
------------------	--------------------------

Implementa `GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >`.

```
7.6.2.14 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::TrackingClusters ( RInside & R )
[virtual]
```

Effettua il riconoscimento dei topic.

Parametri

<code>R</code>	- Istanza di R
----------------	----------------

Implementa `GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >`.

```
7.6.2.15 template<unsigned int DIM> void CategoricalPosteriorAnalysis< DIM >::VisualizeBeta ( RInside & R,
unsigned long BestClustering ) [virtual]
```

Chiama lo script R che visualizza le traiettorie della distribuzione dei topic nel corpus.

Parametri

<i>R</i>	- Istanza di R
<i>BestClustering</i>	- iterazione a cui e' stato individuato il least square clustering

Implementa [GenericPosteriorAnalysis< TypeCategorical< DIM >, DIM >](#).

La documentazione per questa classe è stata generata a partire dal seguente file:

- [PosteriorAnalysis.hpp](#)

7.7 Riferimenti per la struct CLUSTER

Attributi pubblici

- `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`

La documentazione per questa struct è stata generata a partire dal seguente file:

- [Struct.hpp](#)

7.8 Riferimenti per la struct DATACOUNT

Attributi pubblici

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

La documentazione per questa struct è stata generata a partire dal seguente file:

- [Struct.hpp](#)

7.9 Template per la classe GenericCluster< ClassType, DIM >

Modello Generico di Cluster.

```
#include <Cluster.hpp>
```

Membri pubblici

- virtual void **SetTheta** (typename ClassType< DIM >::THETA &)=0
Fissa il parametro latente del cluster.
- virtual void **SetThetaLeft** (typename ClassType< DIM >::THETA &)=0
Fissa il parametro latente del sub-cluster sinistro.
- virtual void **SetThetaRight** (typename ClassType< DIM >::THETA &)=0
Fissa il parametro latente del sub-cluster destro.
- virtual void **SetBeta** (double)=0
Fissa il peso del cluster.
- virtual void **SetBetaLeft** (double)=0
Fissa il peso del sub-cluster sinistro.
- virtual void **SetBetaRight** (double)=0
Fissa il peso del sub-cluster destro.
- virtual void **SetGlobalTable** (unsigned int)=0
Fissa il numero di tavoli globali che caratterizza il cluster.
- virtual void **SetGlobalTableLeft** (unsigned int)=0
Fissa il numero di tavoli globali che caratterizza il sub-cluster sinistro.
- virtual void **SetGlobalTableRight** (unsigned int)=0
Fissa il numero di tavoli globali che caratterizza il sub-cluster destro.
- virtual void **SetStatistics** (typename ClassType< DIM >::STAT &)=0
Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del cluster.
- virtual void **SetStatisticsLeft** (typename ClassType< DIM >::STAT &)=0
Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del sub-cluster sinistro.
- virtual void **SetStatisticsRight** (typename ClassType< DIM >::STAT &)=0
Fissa le statistiche, ovvero gli iperparametri dei parametri latenti del sub-cluster destro.
- virtual void **ViewTheta** (typename ClassType< DIM >::THETA &) const =0
Estrae il parametro latente del cluster.
- virtual void **ViewThetaLeft** (typename ClassType< DIM >::THETA &) const =0
Estrae il parametro latente del sub-cluster sinistro.
- virtual void **ViewThetaRight** (typename ClassType< DIM >::THETA &) const =0
Estrae il parametro latente del sub-cluster destro.
- virtual double **ViewBeta** () const =0
Estra il peso globale del cluster.
- virtual double **ViewBetaLeft** () const =0
Estra il peso globale del sub-cluster sinistro.
- virtual double **ViewBetaRight** () const =0
Estra il peso globale del sub-cluster destro.
- virtual void **ViewStatistics** (typename ClassType< DIM >::STAT &) const =0
Estra la statistica del parametro latente del cluster.
- virtual void **ViewStatisticsLeft** (typename ClassType< DIM >::STAT &) const =0
Estra la statistica del parametro latente del sub-cluster sinistro.
- virtual void **ViewStatisticsRight** (typename ClassType< DIM >::STAT &) const =0
Estra la statistica del parametro latente del sub-cluster destro.
- virtual unsigned int **ViewGlobalTable** () const =0
Estra il numero globale dei tavoli nel cluster.
- virtual unsigned int **ViewGlobalTableLeft** () const =0
Estra il numero globale dei tavoli nel sub-cluster sinistro.
- virtual unsigned int **ViewGlobalTableRight** () const =0
Estra il numero globale dei tavoli nel sub-cluster destro.
- virtual void **ResetStatistics** (unsigned int)=0

- Azzera le statistiche nel cluster.*
- virtual void [ResetStatisticsLeft](#) (unsigned int)=0
Azzera le statistiche nel sub-cluster sinistro.
- virtual void [ResetStatisticsRight](#) (unsigned int)=0
Azzera le statistiche nel sub-cluster destro.
- virtual void [UpdateStatistics](#) (typename ClassType< DIM >::STAT &, typename ClassType< DIM >::STAT &)=0
Aggiorna le statistiche, ovvero gli iperparametri dei parametri latenti dei cluster e sub-cluster.
- virtual bool [IsEmpty](#) () const =0
Controlla se il cluster non contiene elementi.

7.9.1 Descrizione dettagliata

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

Modello Generico di Cluster.

Classe astratta dove tutti i metodi virtuali sono null. Le classi che ereditano da Cluster Generic servono per estrarre, memorizzare ed impostare i dati associati ai cluster ed ai sub-cluster. Pertanto si tratta di classi di appoggio, non fanno nessun tipo di campionamento

Data

Febbraio 2016

7.9.2 Documentazione delle funzioni membro

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

Controlla se il cluster non contiene elementi.

Restituisce

TRUE se il cluster e' vuoto FALSE se non lo e'

Implementato in [CategoricalCluster](#).

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

Azzera le statistiche nel cluster.

Parametri

<i>W</i>	- dimensione della statistica con cui aggiornare gli iperparametri del parametro latente del cluster
----------	--

Implementato in [CategoricalCluster](#).

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

Azzera le statistiche nel sub-cluster sinistro.

Parametri

<code>W</code>	- dimensione della statistica con cui aggiornare gli iperparametri del parametro latente del sub-cluster sinistro
----------------	---

Implementato in [CategoricalCluster](#).

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

Azzera le statistiche nel sub-cluster destro.

Parametri

<code>W</code>	- dimensione della statistica con cui aggiornare gli iperparametri del parametro latente del sub-cluster destro
----------------	---

Implementato in [CategoricalCluster](#).

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

Fissa il peso del cluster.

Parametri

<code>_Beta</code>	- peso in ingresso
--------------------	--------------------

Implementato in [CategoricalCluster](#).

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

Fissa il peso del sub-cluster sinistro.

Parametri

<code>_BetaLeft</code>	- peso in ingresso
------------------------	--------------------

Implementato in [CategoricalCluster](#).

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

Fissa il peso del sub-cluster destro.

Parametri

<code>_BetaRight</code>	- peso in ingresso
-------------------------	--------------------

Implementato in [CategoricalCluster](#).

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

Fissa il numero di tavoli globali che caratterizza il cluster.

Parametri

<i>NrTable</i>	- numero di tavoli che caratterizza il cluster
----------------	--

Implementato in [CategoricalCluster](#).

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

Fissa il numero di tavoli globali che caratterizza il sub-cluster sinistro.

Parametri

<i>NrTableLeft</i>	- numero di tavoli che caratterizza il sub-cluster sinistro
--------------------	---

Implementato in [CategoricalCluster](#).

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

Fissa il numero di tavoli globali che caratterizza il sub-cluster destro.

Parametri

<i>NrTableRight</i>	- numero di tavoli che caratterizza il sub-cluster destro
---------------------	---

Implementato in [CategoricalCluster](#).

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

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

Parametri

<i>_c</i>	- statistiche del cluster
-----------	---------------------------

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

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

Parametri

<i>_cLeft</i>	- statistiche del sub-cluster sinistro
---------------	--

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

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

Parametri

<i>_cRight</i>	- statistiche del sub-cluster destro
----------------	--------------------------------------

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

Fissa il parametro latente del cluster.

Parametri

<code>_Theta</code>	- parametro latente in ingresso di tipo THETA
---------------------	---

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

Fissa il parametro latente del sub-cluster sinistro.

Parametri

<code>_ThetaLeft</code>	- parametro latente in ingresso di tipo THETA
-------------------------	---

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

Fissa il parametro latente del sub-cluster destro.

Parametri

<code>_ThetaRight</code>	- parametro latente in ingresso di tipo THETA
--------------------------	---

7.9.2.17 `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]`

Aggiorna le statistiche, ovvero gli iperparametri dei parametri latenti dei cluster e sub-cluster.

Parametri

<code>counts4cleft</code>	- statistiche per aggiornare gli iperparametri dei parametri latenti del sub-cluster sinistro
<code>counts4cright</code>	- statistiche per aggiornare gli iperparametri dei parametri latenti del sub-cluster destro

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

Estra il peso globale del cluster.

Restituisce

Peso del cluster

Implementato in [CategoricalCluster](#).

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

Estra il peso globale del sub-cluster sinistro.

Restituisce

Peso del sub-cluster sinistro

Implementato in [CategoricalCluster](#).

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

Estra il peso globale del sub-cluster destro.

Restituisce

Peso del del sub-cluster destro

Implementato in [CategoricalCluster](#).

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

Estra il numero globale dei tavoli nel cluster.

Restituisce

Numero di tavoli nel cluster

Implementato in [CategoricalCluster](#).

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

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

Restituisce

Numero di tavoli nel sub-cluster sinistro

Implementato in [CategoricalCluster](#).

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

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

Restituisce

Numero di tavoli nel sub-cluster destr

Implementato in [CategoricalCluster](#).

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

Estra la statistica del parametro latente del cluster.

Parametri

<code>_c</code>	- oggetto di tipo STAT in cui viene memorizzato la statistica del parametro latente del cluster
-----------------	---

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

Estra la statistica del parametro latente del sub-cluster sinistro.

Parametri

<code>_cLeft</code>	- oggetto di tipo STAT in cui viene memorizzato la statistica del parametro latente del sub-cluster sinistro
---------------------	--

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

Estra la statistica del parametro latente del sub-cluster destro.

Parametri

<code>_cRight</code>	- oggetto di tipo STAT in cui viene memorizzato la statistica del parametro latente del sub-cluster destro
----------------------	--

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

Estrae il parametro latente del cluster.

Parametri

<code>_Theta</code>	- oggetto di tipo THETA in cui viene memorizzato il parametro latente del cluster
---------------------	---

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

Estrae il parametro latente del sub-cluster sinistro.

Parametri

<code>_ThetaLeft</code>	- oggetto di tipo THETA in cui viene memorizzato il parametro latente del sub-cluster sinistro
-------------------------	--

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

Estrae il parametro latente del sub-cluster destro.

Parametri

<code>_ThetaRight</code>	- oggetto di tipo THETA in cui viene memorizzato il parametro latente del sub-cluster destro
--------------------------	--

La documentazione per questa classe è stata generata a partire dal seguente file:

- [Cluster.hpp](#)

7.10 Template per la classe `GenericDocument< Type, DIM >`

Classe generica per i gruppi.

```
#include <Document.hpp>
```

Membri pubblici

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

- Aggiorna i pesi dei cluster specifici del gruppo; si veda equazione (3.5) in Relazione_Parisi_Perego.pdf.*
- virtual void **UpdatePiSub** (const double, const double, const unsigned int, **omprng** &)=0
Aggiorna i pesi, specifici del gruppo, dei subcluster del cluster k; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.
- virtual void **UpdateAllPiSub** (const vector< double >, const vector< double >, **omprng** &)=0
Aggiorna i pesi, specifici del gruppo, di tutti i subcluster; ; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.
- virtual void **UpdateLocalTable** (const vector< long double > &, const vector< double > &, **omprng** &)=0
Aggiorna i tavoli; si veda equazione (3.3) in Relazione_Parisi_Perego.pdf.
- virtual void **UpdateLocalTableSub_OneCluster** (const vector< long double > &, const double __, const double, const unsigned int, **omprng** &)=0
Aggiorna i tavoli dei subcluster del cluster k; si veda equazione (3.12) in Relazione_Parisi_Perego.pdf.
- virtual void **UpdateAllLocalTableSub** (const vector< long double > &, const vector< double > &, const vector< double > &, **omprng** &)=0
Aggiorna i tavoli dei subcluster di tutti i cluster; si veda equazione (3.12) in Relazione_Parisi_Perego.pdf.
- virtual void **UpdateZeta** (const typename Type::THETA &, const unsigned int, **omprng** &)=0
Aggiorna l'etichetta per il cluster di un dato, campionata con il metodo Sampling; si veda equazione (3.7) in Relazione_Parisi_Perego.pdf.
- virtual void **UpdateZeta_and_Sub** (const typename Type::THETA &, const typename Type::THETA &, const typename Type::THETA &, const unsigned int, **omprng** &)=0
Aggiorna l'etichetta per il cluster e per il subcluster di un dato, campionate con il metodo Sampling; si vedano equazioni (3.7) - (3.11) in Relazione_Parisi_Perego.pdf.
- virtual void **UpdateZetaSub** (const typename Type::THETA &, const typename Type::THETA &, const typename Type::Point, const unsigned int, const unsigned int, **omprng** &)=0
Distribuisce il dato nei subcluster del cluster k, dopo aver campionato l'etichetta del subcluster con il metodo Sampling; si veda equazione (3.11) in Relazione_Parisi_Perego.pdf.
- virtual void **UpdateZeta** (const unsigned int)=0
Metodo che serve nelle mosse di M-H per rimuovere il cluster k.
- virtual void **UpdateZeta** (const unsigned int, const unsigned int)=0
Metodo che serve nelle mosse di M-H per rimuovere due cluster.
- virtual unsigned int **ViewNj** () const =0
Visualizza il numero di dati nel gruppo j.
- virtual void **ViewData** (vector< typename Type::Point > &) const =0
- virtual void **ViewCounts4c** (const unsigned int, typename Type::STAT &, typename Type::STAT &)=0
Visualizza i conteggi necessari per aggiornare i parametri latenti dei subcluster.
- virtual unsigned int **ViewNumTableID** (const unsigned int) const =0
Estrae il numero di tavoli in uno specifico cluster.
- virtual unsigned int **ViewNumTableLeftID** (const unsigned int) const =0
Estrae il numero di tavoli del subcluster sinistro di uno specifico cluster.
- virtual unsigned int **ViewNumTableRightID** (const unsigned int) const =0
Estrae il numero di tavoli del subcluster destro di uno specifico cluster.
- virtual unsigned int **ViewDataCountID** (const unsigned int) const =0
Estrae il numero di dati nel cluster k.
- virtual unsigned int **ViewDataCountLeftID** (const unsigned int) const =0
Estrae il numero di dati nel subcluster sinistro del cluster k.
- virtual unsigned int **ViewDataCountRightID** (const unsigned int) const =0
Estrae il numero di dati nel subcluster destro del cluster k.
- virtual void **ResetDataCountSub** (const unsigned int)=0
Azzera il conteggio dei dati nel cluster k.
- virtual void **ViewDataCount** (vector< unsigned int > &) const =0
Estrae il vettore del numero di dati in ogni cluster.
- virtual void **ViewDataCountLeft** (vector< unsigned int > &) const =0
Estrae il vettore del numero di dati in ogni subcluster sinistro.

- virtual void [ViewDataCountRight](#) (vector< unsigned int > &) const =0
Estrae il vettore del numero di dati in ogni subcluster destro.
- virtual void [ViewIdCounts](#) (vector< pair< typename Type::Point, unsigned int >> &, const unsigned int)=0
Estrae identificativi e conteggi dei dati nel cluster k.
- virtual void [ViewCluster](#) (const unsigned int, pair< unordered_map< typename Type::Point, unsigned int >, unordered_map< typename Type::Point, unsigned int >> &)=0
Estrae il cluster k.
- virtual double [ViewPiID](#) (const unsigned int) const =0
Estrae il peso, specifico del gruppo, del cluster k.
- virtual double [ViewPiLeftID](#) (const unsigned int) const =0
Estrae il peso, specifico del gruppo, del subcluster sinistro del cluster k.
- virtual double [ViewPiRightID](#) (const unsigned int) const =0
Estrae il peso, specifico del gruppo, del subcluster destro del cluster k.
- virtual void [ViewPi](#) (vector< double > &) const =0
Estrae il vettore di pesi dei cluster specifici dei gruppi.
- virtual void [ViewPiLeft](#) (vector< double > &) const =0
Estrae il vettore di pesi dei subcluster sinistri specifici dei gruppi.
- virtual void [ViewPiRight](#) (vector< double > &) const =0
Estrae il vettore di pesi dei subcluster destri specifici dei gruppi.
- virtual void [SetAlpha](#) (const double)=0
Imposta il parametro di concentrazione del processo di Dirichlet che governa il gruppo.
- virtual void [SetNj](#) (const unsigned int)=0
Imposta il numero di dati nel gruppo j.
- virtual void [SetPi](#) (vector< double > &)=0
Imposta il vettore di pesi dei cluster specifici dei gruppi.
- 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
Inserisce un nuovo cluster.
- virtual void [RemoveCluster](#) (const vector< unsigned int > &)=0
Rimuove i cluster con identificativo presente nel vettore in ingresso.
- virtual void [RemoveCluster](#) (const unsigned int)=0
Rimuove un cluster.
- virtual void [RemoveCluster](#) (unsigned int, unsigned int)=0
Rimuove due cluster.
- virtual unsigned int [CheckLeftSubcluster](#) (const unsigned int)=0
Verifica se un cluster ha il subcluster sinistro vuoto.
- virtual unsigned int [CheckRightSubcluster](#) (const unsigned int)=0
Verifica se un cluster ha il subcluster destro vuoto.
- virtual void [ViewLabel](#) (vector< pair< typename Type::Point, unsigned int >> &)=0
Estrae le etichette associate ai dati.
- virtual void [SetDataset](#) (std::istream &)=0
Acquisisce i dati.
- virtual unsigned int [SortData](#) (unsigned int, [omprng](#) &)=0
Smista i dati nel contenitore Zeta.

Membri privati

- virtual void [UpdateDataCount](#) ()=0
Aggiorna i conteggi dei dati nei cluster.
- virtual void [Sampling](#) (std::vector< unsigned int > &, std::vector< double > &, unsigned int, [omprng](#) &)=0
Campionamento da distribuzione categorica per l'etichetta del cluster o del subcluster.

7.10.1 Descrizione dettagliata

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

Classe generica per i gruppi.

Classe astratta dove tutti i metodi virtuali sono null. Contiene i metodi che devono essere obbligatoriamente definiti in tutte le classi derivate.

Data

Febbraio 2016

7.10.2 Documentazione delle funzioni membro

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

Verifica se un cluster ha il subcluster sinistro vuoto.

Parametri

<code>_k</code>	- id del cluster di cui controllare il subcluster
-----------------	---

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

Verifica se un cluster ha il subcluster destro vuoto.

Parametri

<code>_k</code>	- id del cluster di cui controllare il subcluster
-----------------	---

7.10.2.3 `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, const unsigned int, const unsigned int, const unsigned int) [pure virtual]`

Inserisce un nuovo cluster.

Parametri

<code>NewCluster</code>	- il nuovo cluster
<code>_Pi</code>	- peso del nuovo cluster specifico del gruppo
<code>_PiLeft</code>	- peso del subcluster sinistro nuovo cluster, specifico del gruppo
<code>_PiRight</code>	- peso del subcluster destro del nuovo cluster, specifico del gruppo
<code>_WordCount</code>	- numero di dati nel nuovo cluster
<code>_WordCountLeft</code>	- numero di dati nel subcluster sinistro del nuovo cluster
<code>_WordCount-Right</code>	- numero di dati nel subcluster destro del nuovo cluster
<code>_LocalTable</code>	- numero di tavoli che servono il nuovo piatto nel ristorante j
<code>_LocalTableLeft</code>	- numero di tavoli che servono il nuovo piatto left nel ristorante j
<code>_LocalTable</code>	- numero di tavoli che servono il nuovo piatto right nel ristorante j

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

Rimuove i cluster con identificativo presente nel vettore in ingresso.

Parametri

<code>_k</code>	- vettore con gli id dei cluster da eliminare
-----------------	---

Implementato in [CategoricalDocument< DIM >](#).

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

Rimuove un cluster.

Parametri

<code>_k</code>	- id del cluster da eliminare
-----------------	-------------------------------

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

Rimuove due cluster.

Parametri

<code>_k1</code>	- id del cluster da eliminare
<code>_k2</code>	- id del cluster da eliminare

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

Azzera il conteggio dei dati nel cluster k.

Parametri

<code>k</code>	- id del cluster
----------------	------------------

7.10.2.8 `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]`

Campionamento da distribuzione categorica per l'etichetta del cluster o del subcluster.

Parametri

<code>_temp_counts</code>	- vettore che contiene i conteggi del dato id nei cluster
<code>_Weights</code>	- pesi con cui campionare le/la etichette/a
<code>_nidj</code>	- numero di volte che il dato id compare nel gruppo j
<code>Gen</code>	- generatore di numeri casuali

Implementato in [CategoricalDocument< DIM >](#).

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

Imposta il parametro di concentrazione del processo di Dirichlet che governa il gruppo.

Parametri

<code>_alpha</code>	- parametro di concentrazione del processo di Dirichlet che governa il gruppo
---------------------	---

Implementato in [CategoricalDocument< DIM >](#).

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

Acquisisce i dati.

Parametri

<code>SSTR</code>	- contiene id del dato e numero di volte che il dato compare nel gruppo
-------------------	---

Implementato in [CategoricalDocument< DIM >](#).

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

Imposta il numero di dati nel gruppo j.

Parametri

<code>_Nj</code>	- numero di dati nel gruppo j
------------------	-------------------------------

Implementato in [CategoricalDocument< DIM >](#).

7.10.2.12 `template<typename Type, unsigned int DIM> virtual unsigned int GenericDocument< Type, DIM >::SortData (unsigned int, omprng &) [pure virtual]`

Smista i dati nel contenitore Zeta.

Parametri

<code>_K</code>	- numero iniziale di cluster
<code>Gen</code>	- generatore di numeri casuali

Implementato in [CategoricalDocument< DIM >](#).

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

Aggiorna i tavoli dei subcluster di tutti i cluster; si veda equazione (3.12) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_stirling</code>	- numeri di stirling
<code>_BetaLeft</code>	- pesi globali dei subcluster sinistri
<code>_BetaRight</code>	- pesi globali dei subcluster destri
<code>Gen</code>	- generatore di numeri casuali in parallelo

Implementato in [CategoricalDocument< DIM >](#).

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

Aggiorna i pesi, specifici del gruppo, di tutti i subcluster; ; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.

Parametri

<i>_BetaLeft</i>	- pesi globali dei subcluster sinistri
<i>_BetaRight</i>	- pesi globali dei subcluster destri
<i>Gen</i>	- generatore di numeri casuali in parallelo

Implementato in [CategoricalDocument](#)< DIM >.

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

Aggiorna i tavoli; si veda equazione (3.3) in Relazione_Parisi_Perego.pdf.

Parametri

<i>_stirling</i>	- numeri di stirling
<i>_Beta</i>	- pesi globali dei cluster
<i>Gen</i>	- generatore di numeri casuali in parallelo

Implementato in [CategoricalDocument](#)< DIM >.

7.10.2.16 `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]`

Aggiorna i tavoli dei subcluster del cluster k; si veda equazione (3.12) in Relazione_Parisi_Perego.pdf.

Parametri

<i>_stirling</i>	- numeri di stirling
<i>_BetaLeft</i>	- peso globale del subcluster sinistro del cluster k
<i>_BetaRight</i>	- peso globale del subcluster destro del cluster k
<i>k</i>	- id del cluster
<i>Gen</i>	- generatore di numeri casuali in parallelo

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

Aggiorna i pesi dei cluster specifici del gruppo; si veda equazione (3.5) in Relazione_Parisi_Perego.pdf.

Parametri

<i>_AllBeta</i>	- pesi globali dei cluster
<i>Gen</i>	- generatore di numeri casuali in parallelo

Implementato in [CategoricalDocument](#)< DIM >.

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

Aggiorna i pesi, specifici del gruppo, dei subcluster del cluster k; si veda equazione (3.9) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_BetaLeft</code>	- peso globale del subcluster sinistro del cluster k
<code>_BetaRight</code>	- peso globale dei subcluster destro del cluster k
<code>k</code>	- id del cluster
<code>Gen</code>	- generatore di numeri casuali in parallelo

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

Aggiorna l'etichetta per il cluster di un dato, campionata con il metodo Sampling; si veda equazione (3.7) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_Thetald</code>	- vettore dei pesi del dato in tutti i cluster
<code>_VettId</code>	- id del dato
<code>Gen</code>	- generatore di numeri casuali in parallelo

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

Metodo che serve nelle mosse di M-H per rimuovere il cluster k.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

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

Metodo che serve nelle mosse di M-H per rimuovere due cluster.

Parametri

<code>_k1</code>	- id del cluster
<code>_k2</code>	- id del cluster

7.10.2.22 `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]`

Aggiorna l'etichetta per il cluster e per il subcluster di un dato, campionate con il metodo Sampling; si vedano equazioni (3.7) - (3.11) in Relazione_Parisi_Perego.pdf.

Parametri

<code>_Thetald</code>	- vettore dei pesi del dato in tutti i cluster
<code>_ThetaldLeft</code>	- vettore dei pesi del dato in tutti i subcluster sinistri
<code>_ThetaldRight</code>	- vettore dei pesi del dato in tutti i subcluster destri
<code>_VettId</code>	- id del dato
<code>Gen</code>	- generatore di numeri casuali in parallelo

```
7.10.2.23  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]
```

Distribuisce il dato nei subcluster del cluster k , dopo aver campionato l'etichetta del subcluster con il metodo Sampling; si veda equazione (3.11) in *Relazione_Parisi_Perego.pdf*.

Parametri

<code>_ThetaldLeft</code>	- vettore dei pesi del dato id in tutti i subcluster sinistri
<code>_ThetaldRight</code>	- vettore dei pesi del dato id in tutti i subcluster destri
<code>id</code>	- id del dato
<code>nidjk</code>	- numero di volte che il dato id nel gruppo j e' capitata nel cluster k
<code>k</code>	- id del cluster
<code>Gen</code>	- generatore di numeri casuali in parallelo

7.10.2.24 `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]`

Estrae il cluster k.

Parametri

<code>_k</code>	- id del cluster da estrarre
<code>_Cluster</code>	- struttura in cui viene estratto il cluster

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

Visualizza i conteggi necessari per aggiornare i parametri latenti dei subcluster.

Parametri

<code>_k</code>	- id del cluster
<code>_counts4cleft</code>	- conteggi per parametro latente del subcluster sinistro
<code>_counts4cright</code>	- conteggi per parametro latente del subcluster destro

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

Estrae gli identificativi dei dati nel gruppi j

Parametri

<code>_VettId</code>	- viene riempito con gli id dei dati nel gruppo j
----------------------	---

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

Estrae il vettore del numero di dati in ogni cluster.

Parametri

<code>_WordCount</code>	- vettore del numero di dati in ogni cluster
-------------------------	--

Implementato in [CategoricalDocument< DIM >](#).

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

Estrae il numero di dati nel cluster k.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

Restituisce

Numero di dati nel cluster k

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

Estrae il vettore del numero di dati in ogni subcluster sinistro.

Parametri

<code>_WordCountLeft</code>	- vettore del numero di dati in ogni subcluster sinistro
-----------------------------	--

Implementato in [CategoricalDocument< DIM >](#).

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

Estrae il numero di dati nel subcluster sinistro del cluster k.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

Restituisce

Numero di dati nel subcluster sinistro del cluster k

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

Estrae il vettore del numero di dati in ogni subcluster destro.

Parametri

<code>_WordCount-Right</code>	- vettore del numero di dati in ogni subcluster destro
-------------------------------	--

Implementato in [CategoricalDocument< DIM >](#).

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

Estrae il numero di dati nel subcluster destro del cluster k.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

Restituisce

Numero di dati nel subcluster destro del cluster k

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

Estrae identificativi e conteggi dei dati nel cluster k.

Parametri

<code>_nidjk</code>	- struttura che contiene identificativi e conteggi dei dati nel cluster k
<code>_k</code>	- id del cluster

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

Estrae le etichette associate ai dati.

Parametri

<code>Data</code>	- struttura in cui estrarre le etichette
-------------------	--

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

Visualizza il numero di dati nel gruppo j.

Restituisce

Numero di dati nel gruppo j

Implementato in [CategoricalDocument](#)< **DIM** >.

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

Estrae il numero di tavoli in uno specifico cluster.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

Restituisce

Numero di tavoli nel ristorante j che servono il piatto k

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

Estrae il numero di tavoli del subcluster sinistro di uno specifico cluster.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

Restituisce

Numero di tavoli nel ristorante j che servono il piatto k left

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

Estrae il numero di tavoli del subcluster destro di uno specifico cluster.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

Restituisce

Numero di tavoli nel ristorante j che servono il piatto k right

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

Estrae il vettore di pesi dei cluster specifici dei gruppi.

Parametri

<code>_pi</code>	- vettore di pesi dei cluster specifici dei gruppi
------------------	--

Implementato in [CategoricalDocument< DIM >](#).

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

Estrae il peso, specifico del gruppo, del cluster k.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

Restituisce

Peso specifico del gruppo del cluster k

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

Estrae il vettore di pesi dei subcluster sinistri specifici dei gruppi.

Parametri

<code>_pi</code>	- vettore di pesi dei subcluster sinistri specifici dei gruppi
------------------	--

Implementato in [CategoricalDocument< DIM >](#).

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

Estrae il peso, specifico del gruppo, del subcluster sinistro del cluster k.

Parametri

<code>_k</code>	- id del topic
-----------------	----------------

Restituisce

Peso specifico del gruppo del subcluster sinistro del cluster k

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

Estrae il vettore di pesi dei subcluster destri specifici dei gruppi.

Parametri

<code>_pi</code>	- vettore di pesi dei subcluster destri specifici dei gruppi
------------------	--

Implementato in [CategoricalDocument< DIM >](#).

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

Estrae il peso, specifico del gruppo, del subcluster destro del cluster k.

Parametri

<code>_k</code>	- id del cluster
-----------------	------------------

Restituisce

Peso specifico del documento del subcluster destro del cluster k

La documentazione per questa classe è stata generata a partire dal seguente file:

- [Document.hpp](#)

7.11 Template per la classe GenericPosteriorAnalysis< Type, DIM >

Classe generica per l'analisi a posteriori. Classe virtuale dove tutti i metodi sono null. Ogni classe che eredita deve definire tutti i metodi della classe base e, se necessario, può aggiungere altri metodi. Invoca gli script R per l'analisi delle catene MCMC. Calcola l'indice LPML, riconosce i topic e individua il miglior clustering secondo il criterio dei minimi quadrati.

```
#include <PosteriorAnalysis.hpp>
```

Membri pubblici

- virtual unsigned int [SetAllK](#) ()=0
Imposta la catena AllK.
- virtual unsigned int [SetAllAlpha](#) ()=0
Imposta la catena Alpha.
- virtual unsigned int [SetAllGamma](#) ()=0
Imposta la catena Gamma.
- virtual void [VisualizeBeta](#) (RInside &, unsigned long)=0
Chiama lo script R che visualizza le traiettorie della distribuzione dei cluster nel corpus.
- virtual void [KPosteriorAnalysis](#) (RInside &, const unsigned int, const unsigned int)=0
Chiama lo script R per l'analisi della catena dei K.
- virtual void [AGPosteriorAnalysis](#) (RInside &, const unsigned int, const unsigned int, const unsigned int, const unsigned int, const char, const char)=0
Chiama lo script R per l'analisi delle catene Alpha e Gamma.
- virtual void [Setwd](#) (const std::string &)=0
- virtual void [SetW](#) (const unsigned int)=0
- virtual void [SetD](#) (const unsigned int)=0
Imposta il numero di gruppi.
- virtual void [SetN](#) (const unsigned int)=0
Imposta il numero totale di dati.
- virtual void [SetTheta](#) (const unsigned long)=0

Imposta la ricerca dei cluster, acquisendo da file i e i delle ultime iterazioni.

- virtual void [LoadLabels](#) (const unsigned long)=0

Acquisisce le etichette assegnate ai dati nelle ultime iterazioni.

- virtual unsigned long [LeastSquareClustering](#) ()=0

Individua il miglior clustering secondo il criterio dei minimi quadrati; si veda equazione (4.3) in Relazione_Parisi_Perego.pdf.

- virtual void [LPML](#) (unsigned long)=0

Calcola e stampa a video l'indice LPML; si vedano equazioni (4.1)- (4.2) in Relazione_Parisi_Perego.pdf.

- virtual void [TrackingClusters](#) (RInside &)=0

Effettua il riconoscimento dei cluster.

7.11.1 Descrizione dettagliata

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

Classe generica per l'analisi a posteriori Classe virtuale dove tutti i metodi sono null. Ogni classe che eredita deve definire tutti i metodi della classe base e, se necessario, può aggiungere altri metodi. Invoca gli script R per l'analisi delle catene MCMC. Calcola l'indice LPML, riconosce i topic e individua il miglior clustering secondo il criterio dei minimi quadrati.

7.11.2 Documentazione delle funzioni membro

7.11.2.1 `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]`

Chiama lo script R per l'analisi delle catene Alpha e Gamma.

Parametri

<i>R</i>	- Istanza di R
<i>AlphaBurnin</i>	- numero di valori iniziali della catena Alpha da scartare
<i>AlphaThinning</i>	- tiene un valore della catena Alpha ogni AlphaThinning valori
<i>GammaBurnin</i>	- numero di valori iniziali della catena Gamma da scartare
<i>GammaThinning</i>	- tiene un valore della catena Gamma ogni GammaThinning valori
<i>AlphaTry</i>	- yes se si vuole ripetere l'analisi della catena Alpha, no altrimenti
<i>GammaTry</i>	- yes se si vuole ripetere l'analisi della catena Gamma, no altrimenti

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

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

Chiama lo script R per l'analisi della catena dei K.

Parametri

<i>R</i>	- Istanza di R
<i>Burnin</i>	- numero di valori iniziali della catena catena da scartare
<i>Thinning</i>	- tiene un valore della catena ogni thinning valori

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.3 `template<typename Type, unsigned int DIM> virtual unsigned long GenericPosteriorAnalysis< Type, DIM >::LeastSquareClustering () [pure virtual]`

Individua il miglior clustering secondo il criterio dei minimi quadrati; si veda equazione (4.3) in Relazione_Parisi_Perego.pdf.

Restituisce

Iterazione a cui e' stato individuato il miglior clustering

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.4 `template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::LoadLabels (const unsigned long) [pure virtual]`

Acquisisce le etichette assegnate ai dati nelle ultime iterazioni.

Parametri

<i>NrClusterings</i>	- numero di iterazioni da monitorare
----------------------	--------------------------------------

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.5 `template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::LPML (unsigned long) [pure virtual]`

Calcola e stampa a video l'indice LPML; si vedano equazioni (4.1)- (4.2) in Relazione_Parisi_Perego.pdf.

Parametri

<i>MaxIt</i>	- numero di iterazioni da considerare nel calcolo di LPML
--------------	---

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.6 `template<typename Type, unsigned int DIM> virtual unsigned int GenericPosteriorAnalysis< Type, DIM >::SetAllAlpha () [pure virtual]`

Imposta la catena Alpha.

Restituisce

Lunghezza della catena

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.7 `template<typename Type, unsigned int DIM> virtual unsigned int GenericPosteriorAnalysis< Type, DIM >::SetAllGamma () [pure virtual]`

Imposta la catena Gamma.

Restituisce

Lunghezza della catena

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.8 `template<typename Type, unsigned int DIM> virtual unsigned int GenericPosteriorAnalysis< Type, DIM >::SetAllK () [pure virtual]`

Imposta la catena AllK.

Restituisce

Lunghezza della catena

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

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

Imposta il numero di gruppi.

Parametri

<code>_D</code>	- numero di gruppi
-----------------	--------------------

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

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

Imposta il numero totale di dati.

Parametri

<code>_N</code>	- numero totale di dati
-----------------	-------------------------

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.11 `template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::SetTheta (const unsigned long) [pure virtual]`

Imposta la ricerca dei cluster, acquisendo da file i e i delle ultime iterazioni.

Parametri

<code>MaxIt</code>	- numero di iterazioni dell'algoritmo
--------------------	---------------------------------------

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

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

Imposta la dimensione dell'iperparametro della distribuzione del parametro latente

Parametri

<code>_W</code>	- dimensione dell'iperparametro della distribuzione del parametro latente
-----------------	---

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

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

Imposta la working directory di R

Parametri

<code>_wd</code>	- working directory di R
------------------	--------------------------

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.14 `template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::TrackingClusters (RInside &) [pure virtual]`

Effettua il riconoscimento dei cluster.

Parametri

<code>R</code>	- Istanza di R
----------------	----------------

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

7.11.2.15 `template<typename Type, unsigned int DIM> virtual void GenericPosteriorAnalysis< Type, DIM >::VisualizeBeta (RInside & , unsigned long) [pure virtual]`

Chiama lo script R che visualizza le traiettorie della distribuzione dei cluster nel corpus.

Parametri

<code>R</code>	- Istanza di R
<code>BestClustering</code>	- iterazione a cui e' stato individuato il least square clustering

Implementato in [CategoricalPosteriorAnalysis< DIM >](#).

La documentazione per questa classe è stata generata a partire dal seguente file:

- [PosteriorAnalysis.hpp](#)

7.12 Riferimenti per la struct `greater_for_pair`

Criterio di confronto tra due elementi che sono coppie (unsigned int, double), da utilizzare nel sort di un vettore; il confronto è sul secondo valore nella coppia.

```
#include <Functions.hpp>
```

Membri pubblici

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

7.12.1 Descrizione dettagliata

Criterio di confronto tra due elementi che sono coppie (unsigned int, double), da utilizzare nel sort di un vettore; il confronto è sul secondo valore nella coppia.

La documentazione per questa struct è stata generata a partire dal seguente file:

- [Functions.hpp](#)

7.13 Template per la classe `HDP_MCMC< MODEL, DOCUMENT, DIM >`

[HDP_MCMC](#).

```
#include <HDP_MCMC.hpp>
```

Membri pubblici

- **HDP_MCMC** ()
Costruttore di default della classe in cui tutti gli elementi vengono inizializzati con il loro costruttore di default. Ai valori scalari viene assegnato il valore nullo. I Flags sono inizializzati tutti con FALSE. Gli oggetti inizializzati ma vuoti il numero di threads viene inizializzato invece con il valore passato da terminale.
- **~HDP_MCMC** ()=default
Distruttore di default.
- void **SetK_init** (unsigned int _K)
Imposta il numero di cluster iniziale che desidera l'utente.
- void **SetDataset** (const std::string &Dataset, const std::string &MainVariable)
Acquisisce il dataset da file e le dimensioni del dataset. Chiama dei metodi di Corpus che si occupano di creare le strutture che gestiscono il dataset.
- void **SetAlphaFixed** (double _Alpha)
Imposta Alpha fisso, il cui valore è deciso dall'utente.
- void **SetAlphaPrior** (double _AA, double _AB)
Imposta una prior su Alpha.
- void **SetGammaFixed** (double _Gamma)
Imposta Gamma fisso, il cui valore è deciso dall'utente.
- void **SetGammaPrior** (double _GA, double _GB)
Imposta una prior su Gamma.
- void **SetLambdaInfo** (HYP Lambda)
Imposta il valore su Lambda.
- void **SetSeed** (const unsigned long Seed)
Fissa il seed, unico per l'intera esecuzione dell'algoritmo.
- void **Check_Model** (unsigned long burnin)
Impone il controllo dell'algoritmo con LPML.
- unsigned int **ViewW** ()
Estra la dimensione del parametro latente.
- unsigned int **ViewK** ()
Estra il numero di cluster attuale.
- unsigned int **ViewD** ()
Estra il numero totale di gruppi stanziato.
- unsigned int **ViewN** ()
Estra la dimensione del dataset.
- void **Algorithm** (unsigned int Iterations, unsigned int Iterations_Sub)
Algoritmo globale vedi Algorithm 3 Relazione_Parisi_Perego.pdf.

Tipi privati

- 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

Membri privati

- `template<class T >`
`void Swap (T &Old, T &New)`
Scambia le vecchie proposte inserendole nelle strutture definitive con quelle nuove. Quando il metodo si chiude, le nuove proposte, che non sono state accettate nei passi di M-H, sono distrutte. Nelle mosse globali, le modifiche per le nuove proposte per i cluster e per le etichette vengono fatte direttamente nelle strutture definitive. Nel caso queste non sono accettate e' necessario reinserire nelle strutture definitive la situazione precedente alle proposte. E' un metodo template perche' e' usato sia per la struttura che racchiude i gruppi, Corpus, sia per la struttura che gestisce i clusters, Model.
- `void Summary ()`
Stampa a video un sunto delle impostazioni dell'algoritmo scelte.
- `void SetClusters ()`
Inizializza la struttura che gestisce i Clusters con il numero di cluster iniziale scelto.
- `void UpdateClusterCounts ()`
Aggiorna gli iperparametri del parametro latente, in base a come sono distribuiti i dati nei vari clusters.
- `void UpdateTable ()`
aggiorna gli m_{jk} in ogni gruppo con l'equazione 3.3 in Relazione_Parisi_Perego.pdf. Dopo di che $m_{.k} = \sum_j m_{jk}$ in modo da aggiornare i tavoli nei clusters
- `void UpdateSubTable ()`
aggiorna i tavoli nei sub.cluster di ogni clusters, m_{jkl} , m_{jkr} in ogni gruppo con l'equazione 3.12 in Relazione_Parisi_Perego.pdf. Dopo di che $m_{.kh} = \sum_j m_{jkh}$ in modo da aggiornare i tavoli in dei sub-clusters di ogni cluster
- `void UpdateDocWeights ()`
Aggiorna in ogni gruppo i pesi dei clusters che compaiono nel gruppo.
- `void UpdateDocWeights_Sub ()`
Aggiorna in ogni gruppo i pesi dei sub-clusters di ogni cluster che compare nel gruppo.
- `void UpdateAssignment_Cluster_and_Subcluster ()`
assegna le nuove etichette del cluster e del sub-cluster ad ogni dato
- `void UpdateAssignment_Cluster ()`
assegna le nuove etichette del cluster, senza assegnare il sub-cluster, ad ogni dato. Viene utilizzato nelle mosse di Merge/Split globale per fare le nuove proposte per le nuove etichette
- `void UpdateBeta ()`
Aggiorna i pesi globali di ogni cluster, compreso quello vuoto; Equazione di campionamento 3.4 di Relazione_Parisi_Perego.pdf.
- `void UpdateAllBetaSub ()`
Aggiorna i pesi globali dei sub-cluster di ogni cluster, compresi i sub-cluster del cluster vuoto; Equazione di campionamento 3.8 di Relazione_Parisi_Perego.pdf.
- `void UpdateBetaSub (const ClusterID k)`
Aggiorna i pesi globali dei sub-cluster del cluster identificato dal suo ID. Utilizzato nel metodo UpdateAllbetaSub() e nei passi di Gibbs-sampler per le proposte dei sub-topic dopo aver accettato le mosse di M-H. Equazione di campionamento 3.8 di Relazione_Parisi_Perego.pdf.
- `void EmptyCluster ()`
Controlla quali cluster sono vuoti e li elimina.
- `bool IsEmptySubcluster (const ClusterID _k)`
Controlla se uno dei due sub-cluster del cluster identificato con il suo ID, e' vuoto.
- `void computeLogL ()`
Calcola la matrice 8.12 con l'equazione 8.13, vedi Relazione_Parisi_Perego.pdf.
- `long double logq ()`
Calcola la quantita' 3.25 della Relazione_Parisi_Perego.pdf.
- `void Gibbs_SubCluster (const vector< ClusterID > &ProposedClusters)`
Passi di Gibbs Sampler per campionare le nuove proposte per i sub-cluster dei nuovi cluster. Vengono utilizzate le equazioni da 3.8 a 3.12.
- `void LocalSplit ()`

Mosse di split locale. Si propone la divisione di un cluster alla volta, in due cluster. Questa proposta e' la stessa per tutti i gruppi. Se si accetta la nuova prposta si campionano si sub-cluster dei due nuovi cluster in ogni gruppo.

- void [LocalMerge](#) ()

Mosse di merge locale. Si formano delle coppie casuali dei cluster attualmente presenti, cui proporre l'unione per formare un nuovo cluster. Se si accetta la nuova proposta si campionano i sub-cluster del nuovo cluster in ogni gruppo.

- void [GlobalMerge](#) ()

Mosse di merge globale. Si formano delle una coppia casuale dei cluster attualmente presenti, cui proporre l'unione per formare un nuovo cluster. Si campionano le nuove quantita' per tutti i cluster, anche per quelli che erano gia' presenti. Se si accetta la nuova proposta si campionano i sub-cluster di tutti i clusters.

- void [GlobalSplit](#) ()

Mosse di split globale. Si sceglie casualmente un cluster non vuoto di cui proporre lo split e formare due nuovi clusters. Si campionano le nuove quantita' per tutti i cluster, anche per quelli che erano gia' presenti. Se si accetta la nuova proposta si campionano i sub-cluster di tutti i clusters.

- void [AlphaPrior](#) ()

Campionamento della nuova Alpha, da usare nell'iterazione successiva dell'algoritmo.

- void [GammaPrior](#) ()

Campionamento della nuova Gamma, da usare nell'iterazione successiva dell'algoritmo.

- void [UpdateK](#) ()

Aggiornamento del K corrente a fronte di aggiunta o eliminazione di clusters.

- void [UpdateAllK](#) ()

Aggiornamento dello storico dei K. Fatto alla fine di ogni iterazione.

- void [SaveAllK](#) ()

Salva su file lo storico dei K alla fine dell'algoritmo, per fare le analisi a posteriori.

- void [SaveAllAlpha](#) ()

Salva su file lo storico degli Alpha, alla fine dell'algoritmo, per fare le analisi a posteriori.

- void [SaveAllGamma](#) ()

Salva su file lo storico dei Gamma, alla fine dell'algoritmo, per fare le analisi a posteriori.

- void [SaveLastBeta](#) ()

Salva su file i pesi globali dei clusters delle ultime 100 iterazioni.

- void [SaveLastTheta](#) ()

Salva su file i parametri latenti di ogni cluster che si e' manifestato nelle ultime 100 iterazioni.

- void [LPML](#) ()

Salva su file le quantita' necessarie per il calcolo dell' LPML, scartando le prime burnin iterazioni.

- void [SaveLabels](#) ()

Salva su file le etichette dei dati, ad ogni iterazione.

Attributi privati

- Corpus [corpus](#)

Oggetto che gestisce i dati.

- std::tuple< bool, bool, bool,
bool, bool, bool, bool, bool,
bool > [Flags](#)

Flags che fornisce le informazioni necessarie per eseguire l'algoritmo con le caratteristiche richieste dall'utente. Il primo elemento e' true se il dataset e' stato caricato. Il secondo elemento e' true se sono state inserite informazioni su Alpha Il terzo elemento e' true se si vuole eseguire l'algoritmo con le prior su Alpha Il quarto elemento e' false se si vuole eseguire l'algoritmo con Alpha fisso Il quinto elemento e' true se sono stati inserite informazioni su Gamma Il quinto elemento e' true se si vuole eseguire l'algoritmo con le prior su Gamma Il quinto elemento e' false se si vuole eseguire l'algoritmo con Gamma fisso Il sesto elemento e' true se si sono inserite informazioni su Lambda Il settimo elemento e' true se si e' inserito un numero iniziale di cluster L'ottavo elemento e' true se si e' scelto un seme Il non elemento e' true se si vuole monitorare il modello con LPML.

- unsigned int [D](#)

- Numero totale di gruppi.*

 - unsigned int [N](#)
- Numero totale di dati.*

 - unsigned int [W](#)
- Dimensione dell'iperparametro della distribuzione del parametro latente.*

 - unsigned long [It](#)
- Iterazione corrente.*

 - unsigned long [MaxIt](#)
- Numeror massimo iterazioni (criterio stop)*

 - unsigned long [MaxIt_SubCluster](#)
- Numero di iterazioni per i sub-cluster (criterio di stop)*

 - double [Gamma](#)
- Parametro di concentrazione del processo di dirchlet che governa i clusters (all'iteriazione It)*

 - vector< double > [AllGamma](#)
- Storico dei Gamma in caso dell'uso della prior.*

 - double [GA](#)
- Parametro di forma della prior su Gamma.*

 - double [GB](#)
- Parametro di rate della prior su Gamma.*

 - double [Alpha](#)
- Parametro di concentrazione del processo di dirchlet che governa i cluster nel gruppo (all'iteriazione It)*

 - vector< double > [AllAlpha](#)
- Strico degli Alpha nel caso dell'uso della prior.*

 - double [AA](#)
- Parametro di forma della prior su Alpha.*

 - double [AB](#)
- Parametro di rate della prior su Alpha.*

 - unsigned int [K](#)
- Numero corrente di cluster.*

 - vector< unsigned int > [AllK](#)
- Storico dei K.*

 - vector< long double > [LogStirlingNumbers](#)
- Stirling Number in scala logaritmica.*

 - MODEL< DIM > [Model](#)
- Modello per la distribuzione iniziale.*

 - double [Beta_empty](#)
- peso globale del cluster "vuoto" all'iterazione It. aggrega i pesi dei cluster (al momento vuoti), che potrebbero manifestarsi nelle iterazioni successive alla It-esima.*

 - vector< double > [logL](#)
- Matrice necessaria per il calcolo dell' Hasting Ratio calcolata con le equazioni 8.12 e 8.13 di Relazione_Parisi_Perego.pdf.*

 - unsigned int [m](#)
- Numero totale di tavoli, da usare per campionare la Gamma, viene aggiornato dopo aver campionato dall'equazione 3.3 di Relazione_Parisi_Perego.pdf.*

 - vector< double > [CPO](#)
- Oggetto che memorizza $\sum_{It=1}^{MaxIt} f_{ij}(y_{ij}|\theta_{z_{ij}}^g)$.*

 - unsigned long [burnin_CPO](#)
- Iterazione dal quale iniziare a calcolare il CPO.*

 - [omprng](#) [Gen](#)
- Generatore di numeri casuali in parallelo.*

 - unsigned int [OMP_NUM_THREADS](#)

7.13.1 Descrizione dettagliata

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

HDP_MCMC.

Implementazione algoritmo globale. Gibbs Sampler per i clusters e i sub-cluster, Metropolis-Hastings per le mosse di Merge/Split locale e globale. Sono implementate le equazioni di campionamento dei pesi globali e dei tavoli. Si occupa di aggiornare i conteggi dei cluster in Model, controllando la situazione dei gruppi. Per dettagli sul funzionamento dell'algoritmo globale consultare documento [Relazione_Parisi_Perego.pdf](#) capitolo 3. Questa classe si appoggia sulle classi [Cluster.hpp](#) per la gestione dei cluster e la definizione di verosimiglianza, [Model.hpp](#) per l'eventuale campionamento dalle distribuzioni di interesse, l'inferenza sui parametri latenti e la definizione di prior e invece si appoggia a [Document.hpp](#) per la gestione dei dati e per il campionamento delle etichette. La particolare classe Model e' l'istanza del parametro template MODEL. In [Model.hpp](#) e' implementato CategoricalModel (prior Dirichlet).

Data

Febbraio 2016

7.13.2 Documentazione delle funzioni membro

```
7.13.2.1 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 )
```

Algoritmo globale vedi Algorithm 3 [Relazione_Parisi_Perego.pdf](#).

Parametri

<i>Iterations</i>	- numero massimo di iterazioni (criterio di stop)
<i>Iterations_Sub</i>	- numero massimo di iterazione per il gibbs sampler che campiona i sub-topic (criterio di stop)

```
7.13.2.2 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 )
```

Impone il controllo dell'algoritmo con LPML.

Parametri

<i>burnin</i>	- indica da quale iterazioni iniziare a calcolarlo
---------------	--

```
7.13.2.3 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]
```

Passi di Gibbs Sampler per campionare le nuove proposte per i sub-cluster dei nuovi cluster. Vengono utilizzate le equazioni da 3.8 a 3.12.

Parametri

<i>Proposed-Clusters</i>	- Etichette dei nuovi cluster di cui bisogna fare la proposta per i rispettivi sub-cluster
--------------------------	--

7.13.2.4 `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]`

Controlla se uno dei due sub-cluster del cluster identificato con il suo ID, e' vuoto.

Restituisce

TRUE se uno dei sub-cluster e' vuoto.

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

Imposta Alpha fisso, il cui valore e' deciso dall'utente.

Parametri

<code>_Alpha</code>	- Valore fisso assegnato ad alpha dall'utente
---------------------	---

7.13.2.6 `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)`

Imposta una prior su Alpha.

Parametri

<code>_AA</code>	- parametro di forma passato dall'utente
<code>_AB</code>	- parametro di rate passato dall'utente

7.13.2.7 `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)`

Acquisisce il dataset da file e le dimensioni del dataset Chiama dei metodi di Corpus che si occupano di creare le strutture che gestiscono il dataset.

Parametri

<code>Dataset</code>	- Nome del file che contiene i dati
<code>MainVariable</code>	- Nome del file che contiene le informazioni sulle dimensioni

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

Imposta Gamma fisso, il cui valore e' deciso dall'utente.

Parametri

<code>_Gamma</code>	- Valore fisso assegnato a Gamma dall'utente
---------------------	--

7.13.2.9 `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)`

Imposta una prior su Gamma.

Parametri

<code>_GA</code>	- parametro di forma passato dall'utente
<code>_GB</code>	- parametro di rate passato dall'utente

7.13.2.10 `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)`

Imposta il numero di cluster iniziale che desidera l'utente.

Parametri

<code>_K</code>	- Cluster iniziali
-----------------	--------------------

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

Imposta il valore su Lambda.

Parametri

<code><i>Lambda</i></code>	- Parametro passato dall'utente
----------------------------	---------------------------------

7.13.2.12 `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)`

Fissa il seed, unico per l'intera esecuzione dell'algoritmo.

Parametri

<code><i>Seed</i></code>	- seed
--------------------------	--------

7.13.2.13 `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]`

Scambia le vecchie proposte inserendole nelle strutture definitive con quelle nuove. Quando il metodo si chiude, le nuove proposte, che non sono state accettate nei passi di M-H, sono distrutte. Nelle mosse globali, le modifiche per le nuove proposte per i cluster e per le etichette vengono fatte direttamente nelle strutture definitive. Nel caso queste non sono accettate e' necessario reinserire nelle strutture definitive la situazione precedente alle proposte. E' un metodo template perche' e' usato sia per la struttura che racchiude i gruppi, Corpus, sia per la struttura che gestisce i clusters, Model.

Parametri

<code><i>Old</i></code>	- Situazione antecedente al cambiamento
<code><i>New</i></code>	- Nuove proposte, non accettate

7.13.2.14 `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]`

Aggiorna i pesi globali dei sub-cluster del cluster identificato dal suo ID. Utilizzato nel metodo UpdateAllBetaSub() e nei passi di Gibbs-sampler per le proposte dei sub-topic dopo aver accettato le mosse di M-H. Equazione di campionamento 3.8 di Relazione_Parisi_Perego.pdf.

Parametri

<i>k</i>	- id del cluster di cui si vuole aggiornare i pesi globali dei suoi sub-cluster
----------	---

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

Estra il numero totale di gruppi stanziato.

Restituisce

Numero totale di gruppi

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

Estra il numero di cluster attuale.

Restituisce

Numero di cluster

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

Estra la dimensione del dataset.

Restituisce

dimensione del dataset

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

Estra la dimensione del parametro latente.

Restituisce

dimensione parametro latente

7.13.3 Documentazione dei membri dato

7.13.3.1 `template<template< unsigned int > class MODEL, template< unsigned int > class DOCUMENT, unsigned int DIM =
1> MODEL<DIM> HDP_MCMC< MODEL, DOCUMENT, DIM >::Model [private]`

Modello per la distribuzione iniziale.

Tiene conto dell'opportuna verosimiglianza per avere un modello bayesiano coniugato per quanto riguarda l'inferenza sui parametri latenti.

```
7.13.3.2 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]
```

Numero di threads

La documentazione per questa classe è stata generata a partire dal seguente file:

- [HDP_MCMC.hpp](#)

7.14 Template per la classe ModelGeneric< Type, DIM >

Modello Generico di Model.

```
#include <Model.hpp>
```

Membri pubblici

- virtual unsigned int [ViewK](#) () const =0
Da informazione sul numero di clusters correnti.
- virtual void [ViewKey](#) (vector< unsigned int > &) const
Visualizza gli ID dei clusters attualmente presenti.
- virtual void [SetHyperparameter](#) (const typename Type::HYP &)=0
Fissa gli iperparametri dei parametri latenti.
- virtual void [DefaultHyperparameter](#) (size_t)=0
Fissa gli iperparametri dei parametri latenti con i valori di default.
- virtual void [SetInitialClusters](#) (unsigned int)=0
Fissa i cluster iniziali, assegnando i pesi iniziali.
- virtual double [Marginalized_Loglikelihood](#) (const unsigned int)=0
Calcola la verosimiglianza logaritmica marginale del cluster, una volta specificato l'id del clusters.
- virtual double [Loglikelihood](#) (const typename Type::Point, const unsigned int)=0
Calcola la loglikelihood in un punto X una volta precisato l'id del cluster.
- virtual double [LoglikelihoodLeft](#) (const typename Type::Point, const unsigned int)=0
Calcola la loglikelihood in un punto X una volta precisato l'id del sub-cluster sinistro.
- virtual double [LoglikelihoodRight](#) (const typename Type::Point, const unsigned int)=0
Calcola la loglikelihood in un punto X una volta precisato l'id del sub-cluster destro.
- virtual long double [LogDensity](#) (const unsigned int)=0
Calcola la densita' del parametro latente che caratterizza il cluster specificato.
- virtual void [UpdateThetaCluster](#) (omprng &)=0
Aggiornamento dei parametro latenti di tutti i clusters.
- virtual void [UpdateThetaSubCluster](#) (omprng &)=0
Aggiornamento dei parametro latenti del sub-cluster sinistro e destro di tutti i clusters.
- virtual void [UpdateOneThetaCluster](#) (const unsigned int, omprng &)=0
Aggiornamento dei parametro latenti di un cluster, una volta precisato il suo id.
- virtual void [UpdateOneThetaSubCluster](#) (const unsigned int, omprng &)=0
Aggiornamento dei parametro latenti del sub-cluster sinistro e destro di un cluster, una volta precisato il sui id.
- virtual void [AddOneCluster](#) (const unsigned int)=0
Aggiunge un cluster vuoto ai clusters presenti ed aggiorna il parametro K, che identifica il numero di cluster attuali. I parametri del cluster devono essere specificati in un secondo momento.
- virtual void [RemoveOneCluster](#) (const unsigned int)=0
Rimuove un cluster una volta specificato l'id, ed aggiorna il parametro K, che identifica il numero di cluster attuali.
- virtual void [RemoveClusters](#) (const vector< unsigned int > &)=0

Rimuove piu' clusters una volta che sono specificati gli ID, ed aggiorna il parametro K, che identifica il numero di cluster attuali.

- virtual void [ViewBeta](#) (vector< double > &)=0
Mostra i pesi globali dei cluster attualmente presenti.
- virtual void [ViewBetaLeft](#) (vector< double > &)=0
Mostra i pesi globali del sub-cluster sinistro dei clusters attualmente presenti.
- virtual void [ViewBetaRight](#) (vector< double > &)=0
Mostra i pesi globali del sub-cluster destro dei clusters attualmente presenti.
- virtual void [PrintTheta](#) ()=0
Stampa su file i valori dei parametri latenti dei cluster attualmente presenti.
- virtual void [PrintLambdaInfo](#) () const =0

7.14.1 Descrizione dettagliata

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

Modello Generico di Model.

Classe astratta dove tutti i metodi virtuali sono null. Le classi che ereditano da [ModelGeneric](#) servono per campionare i parametri latenti e gestire i relativi iperparametri, che dipendono dal modello scelto ed utilizzare. Gestisce funzioni variabili in base al modello, come il calcolo delle versimiglianze, delle marginali e delle densità. Gestione di tutti i cluster, aggiunge e rimuove cluster.

Data

Febbraio 2016

7.14.2 Documentazione delle funzioni membro

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

Aggiunge un cluster vuoto ai clusters presenti ed aggiorna il parametro K, che identifica il numero di cluster attuali. I parametri del cluster devono essere specificati in un secondo momento.

Parametri

<code>_K</code>	- Id del nuovo cluster che si aggiunge.
-----------------	---

Implementato in [CategoricalModel< DIM >](#).

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

Fissa gli iperparametri dei parametri latenti con i valori di default.

Parametri

<code>W</code>	- dimensione degli iperparametri
----------------	----------------------------------

Implementato in [CategoricalModel< DIM >](#).

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

Calcola la densità del parametro latente che caratterizza il cluster specificato.

Parametri

<code>_K</code>	- Id del cluster
-----------------	------------------

Restituisce

Desita' logaritmica del cluster `_K`

Implementato in [CategoricalModel< DIM >](#).

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

Calcola la loglikelihood in un punto X una volta precisato l'id del cluster.

Parametri

<code>X</code>	- punto di valutazione
<code>_K</code>	- id del cluster

Restituisce

verosmiglianza logaritmica valutante nel punto X del cluster `_K`

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

Calcola la loglikelihood in un punto X una volta precisato l'id del sub-cluster sinistro.

Parametri

<code>X</code>	- punto di valutazione
<code>_K</code>	- id del sub-cluster sinistro

Restituisce

verosmiglianza logaritmica valutante nel punto X del sub-cluster sinistro del cluster `_K`

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

Calcola la loglikelihood in un punto X una volta precisato l'id del sub-cluster destro.

Parametri

<code>X</code>	- punto di valutazione
<code>_K</code>	- id del sub-cluster destro

Restituisce

verosmiglianza logaritmica valutante nel punto X del sub-cluster destro del cluster `_K`

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

Calcola la verosmiglianza logaritmica marginale del cluster, una volta specificato l'id del clusters.

Parametri

<code>_K</code>	- id del cluster di cui si vuole calcolare la verosimiglianza logaritmica marginale
-----------------	---

Implementato in [CategoricalModel< DIM >](#).

7.14.2.8 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::PrintLambdaInfo ()`
`const [pure virtual]`

Stampa a video le informazioni degli iperparametri dei parametri latenti dei cluster

Implementato in [CategoricalModel< DIM >](#).

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

Rimuove piu' clusters una volta che sono specificati gli ID, ed aggiorna il parametro K, che identifica il numero di cluster attuali.

Parametri

<code>_K</code>	- Vettore degli Id dei clusters che si vogliono rimuovere
-----------------	---

Implementato in [CategoricalModel< DIM >](#).

7.14.2.10 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::RemoveOneCluster (`
`const unsigned int) [pure virtual]`

Rimuove un cluster una volta specificato l'id, ed aggiorna il parametro K, che identifica il numero di cluster attuali.

Parametri

<code>_K</code>	- Id del cluster che si vuole rimuovere
-----------------	---

Implementato in [CategoricalModel< DIM >](#).

7.14.2.11 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::SetHyperparameter (`
`const typename Type::HYP &) [pure virtual]`

Fissa gli iperparametri dei parametri latenti.

Parametri

<code>_Lambda</code>	- oggetto di tipo HYP, iperparametri in ingresso del parametro latente
----------------------	--

7.14.2.12 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::SetInitialClusters (`
`unsigned int) [pure virtual]`

Fissa i cluster iniziali, assegnando i pesi iniziali.

Parametri

<code>_K</code>	- Numero di cluster iniziali
-----------------	------------------------------

Implementato in [CategoricalModel< DIM >](#).

7.14.2.13 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::UpdateOneThetaCluster (const unsigned int, omprng &) [pure virtual]`

Aggiornamento dei parametro latenti di un cluster, una volta precisato il suo id.

Equazione (3.6) della relazione Relazione_Parisi_Perego.pdf

Parametri

<code>_K</code>	- id del cluster di cui si vuole aggiornare parametri latenti
<code>Gen</code>	- Generatore dei numeri casuali

Implementato in [CategoricalModel< DIM >](#).

7.14.2.14 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::UpdateOneThetaSubCluster (const unsigned int, omprng &) [pure virtual]`

Aggiornamento dei parametro latenti del sub-cluster sinistro e destro di un cluster, una volta precisato il suo id.

Equazione (3.10) della relazione Relazione_Parisi_Perego.pdf

Parametri

<code>_K</code>	- id del cluster di cui si vuole aggiornare i parametri latenti nei suoi sub-clusters
<code>Gen</code>	- Generatore dei numeri casuali

Implementato in [CategoricalModel< DIM >](#).

7.14.2.15 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::UpdateThetaCluster (omprng &) [pure virtual]`

Aggiornamento dei parametro latenti di tutti i clusters.

Equazione (3.6) della relazione Relazione_Parisi_Perego.pdf

Parametri

<code>Gen</code>	- Generatore dei numeri casuali
------------------	---------------------------------

Implementato in [CategoricalModel< DIM >](#).

7.14.2.16 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::UpdateThetaSubCluster (omprng &) [pure virtual]`

Aggiornamento dei parametro latenti del sub-cluster sinistro e destro di tutti i clusters.

Equazione (3.10) della relazione Relazione_Parisi_Perego.pdf

Parametri

<code>Gen</code>	- Generatore dei numeri casuali
------------------	---------------------------------

Implementato in [CategoricalModel< DIM >](#).

7.14.2.17 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::ViewBeta (vector< double > &) [pure virtual]`

Mostra i pesi globali dei cluster attualmente presenti.

Parametri

<i>AllBeta</i>	- vettore riempito con i pesi globali dei clusters
----------------	--

Implementato in [CategoricalModel< DIM >](#).

7.14.2.18 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::ViewBetaLeft (vector< double > &) [pure virtual]`

Mostra i pesi globali del sub-cluster sinistro dei clusters attualmente presenti.

Parametri

<i>AllBetaLeft</i>	- riempito con i pesi globali del sub-cluster sinistro dei clusters
--------------------	---

Implementato in [CategoricalModel< DIM >](#).

7.14.2.19 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::ViewBetaRight (vector< double > &) [pure virtual]`

Mostra i pesi globali del sub-cluster destro dei clusters attualmente presenti.

Parametri

<i>AllBetaRight</i>	- riempito con i pesi globali del sub-cluster destro dei clusters
---------------------	---

Implementato in [CategoricalModel< DIM >](#).

7.14.2.20 `template<typename Type, unsigned int DIM = 1> virtual unsigned int ModelGeneric< Type, DIM >::ViewK () const [pure virtual]`

Da informazione sul numero di clusters correnti.

Restituisce

il numero corrente di clusters

Implementato in [CategoricalModel< DIM >](#).

7.14.2.21 `template<typename Type, unsigned int DIM = 1> virtual void ModelGeneric< Type, DIM >::ViewKey (vector< unsigned int > &) const [virtual]`

Visualizza gli ID dei clusters attualmente presenti.

Parametri

<i>Key</i>	- oggetto dove salvare gli Id dei clusters
------------	--

Reimplementata in [CategoricalModel< DIM >](#).

La documentazione per questa classe è stata generata a partire dal seguente file:

- [Model.hpp](#)

7.15 Riferimenti per la struct NJK

Numero degli elementi del gruppo j che sono nel cluster k.

```
#include <Struct.hpp>
```

Attributi pubblici

- 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**

7.15.1 Descrizione dettagliata

Numero degli elementi del gruppo j che sono nel cluster k.

La documentazione per questa struct è stata generata a partire dal seguente file:

- [Struct.hpp](#)

7.16 Riferimenti per la struct NUMTABLE

Tavoli.

```
#include <Struct.hpp>
```

Attributi pubblici

- 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**

7.16.1 Descrizione dettagliata

Tavoli.

La documentazione per questa struct è stata generata a partire dal seguente file:

- [Struct.hpp](#)

7.17 Riferimenti per la classe omprng

Libreria Omprng per la generazione dei numeri casuali in OpenMp.

```
#include <omprng.hpp>
```

Membri pubblici

- `omprng ()`
Costruttore di default.
- `~omprng ()`
Distruttore.
- `void fixedSeed (long unsigned int)`
Fissa seed.
- `void randomSeed ()`
Genera casualmente un seed.
- `void setNumThreads (int)`
Imposta il numero di thread.
- `double runif ()`
Generazione da Uniforme tra 0 ed 1.
- `double runif (double, double)`
Generazione da Uniforme tra due valori fissati.
- `double rnorm (double, double)`
Generazione da Gaussiana.
- `double rexp (double)`
Generazione da Esponenziale.
- `double rgamma (double, double)`
Generazione da Gamma.
- `double rchisq (unsigned int)`
Generazione da ChiQuadro.
- `double rbeta (double, double)`
Generazione da Beta.
- `unsigned int rdiscrete (std::vector< double > &)`
Campionamento da una variabile aleatoria discreta con supporto 0:(K-1)
- `unsigned int runifdiscrete (unsigned int)`
Campionamento da una variabile aleatoria discreta uniforme con supporto 0:(N-1)
- `unsigned int rbernoulli (double p)`
Generazione da una Bernoulli $X \sim \text{Bernoulli}(p)$ $P(X=1) = p$ $P(X=0) = 1-p$ $X=\{0,1\}$.
- `unsigned int rbinomial (unsigned int n, double p)`
Generazione da una Binomiale $X \sim \text{Bin}(n, p)$.
- `void rdirichlet (const vector< double > ¶ms, vector< double > &dir_sampled)`
Generazione da una Dirichlet di dimensine d $(X_1, \dots, X_d) \sim \text{Dir}(a_1, \dots, a_d)$ $a_i > 0$ $i=\{1, \dots, d\}$ $\sum_{i=1}^d X_i = 1$.

Attributi privati

- `int nprocs`
Numero di processori a disposizione.
- `RngStream * myRng`
Oggetto [RngStream](#). Implementazione dovuta a Matthew Bognar.

7.17.1 Descrizione dettagliata

Libreria Omprng per la generazione dei numeri casuali in OpenMp.

Si deve lo sviluppo a Matthew Bognar Department of Statistics and Actuarial Science University of Iowa <http://www.stat.uiowa.edu/~mbognar/omprng> matthew-bognar@uiowa.edu

Data

Luglio 2014

7.17.2 Documentazione delle funzioni membro

7.17.2.1 `void omprng::fixedSeed (long unsigned int myInt)`

Fissa seed.

Parametri

<i>Seed</i>	- seme casuale
-------------	----------------

7.17.2.2 `unsigned int omprng::rbernoulli (double p)`

Generazione da una Bernoulli $X \sim \text{Bernoulli}(p)$ $P(X=1) = p$ $P(X=0) = 1-p$ $X=\{0,1\}$.

Parametri

<i>p</i>	- probabilita' del successo
----------	-----------------------------

Restituisce

realizzazione di una Bernoulli con supporto $\{0,1\}$

Data

Febbraio 2016

7.17.2.3 `double omprng::rbeta (double alpha, double beta)`

Generazione da Beta.

$X \sim \text{beta}(\alpha, \beta)$

$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}$

$0 < x < 1$, $\alpha > 0$, $\beta > 0$

$E(X) = \alpha/(\alpha+\beta)$, $\text{Var}(X) = \alpha\beta / ((\alpha+\beta+1)(\alpha+\beta)^2)$

Parametri

<i>alpha</i>	- primo parametro di forma
<i>beta</i>	- secondo parametro di forma

Restituisce

x - realizzazione di una Beta con df parametri di forma *alpha* e *beta*

7.17.2.4 `unsigned int omprng::rbinomial (unsigned int n, double p)`

Generazione da una Binomiale $X \sim \text{Bin}(n, p)$.

$P(X=k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$ $k=\{0,1,\dots,n\}$

Parametri

n	- numero di prove
p	- probabilita' di successo

Restituisce

realizzazione di una Binomiale con supporto $\{0,1,\dots,n\}$

Data

Febbraio 2016

7.17.2.5 double omprng::rchisq (unsigned int)

Generazione da ChiQuadro.

$X \sim \text{chisq}(df)$

$f(x) = 1/(\text{gamma}(df/2) * 2^{df/2}) * x^{df/2-1} * \exp(-x/2)$

$x > 0, df = 1, 2, 3, \dots$

$E(X) = df, \text{Var}(X) = 2 * df$

Parametri

df	- gradi di liberta'
------	---------------------

Restituisce

realizzazione di una ChiQuadro con df gradi di liberta'

7.17.2.6 void omprng::rdirichlet (const vector< double > & params, vector< double > & dir_sampled)

Generazione da una Dirichlet di dimensine d $(X_1, \dots, X_d) \sim \text{Dir}(a_1, \dots, a_d)$ $a_i > 0 \ i=\{1, \dots, d\}$ $\sum_{i=1}^d X_i = 1$.

Parametri

$params$	- vettore dei parametri della Dirichlet
$dir_sampled$	- realizzazione della dirichlet

Data

Febbraio 2016

7.17.2.7 unsigned int omprng::rdiscrete (std::vector< double > & weights)

Campionamento da una variabile aleatoria discreta con supporto $0:(K-1)$

$K = \text{size}(\text{inputvector})$

Parametri

$Logp$	- logaritmi probabilita'/pesi
--------	-------------------------------

Restituisce

realizzazione di una variabile aleatoria discreta con supporto $0:(K-1)$ e probabilita' rispettive $\exp\{Logp\}$

7.17.2.8 `double omprng::rexp (double theta)`

Generazione da Esponenziale.

$X \sim \text{exp}(\text{theta})$

$f(x) = 1 / \text{theta} * \exp(-x/\text{theta})$

$x > 0, \text{theta} > 0$

$E(X) = \text{theta}, \text{Var}(X) = \text{theta}^2$

Parametri

<i>theta</i>	- parametro di scala
--------------	----------------------

Restituisce

realizzazione di una esponenziale di parametro di scala *theta*

7.17.2.9 `double omprng::rgamma (double alpha, double beta)`

Generazione da Gamma.

$X \sim \text{gamma}(\text{alpha}, \text{beta})$

$f(x) = 1/(\text{gamma}(\text{alpha}) * \text{beta}^{\text{alpha}}) * x^{\text{alpha}-1} * \exp(-x/\text{beta})$

$x > 0, \text{alpha} > 0, \text{beta} > 0$

$E(X) = \text{alpha} * \text{beta}, \text{Var}(X) = \text{alpha} * \text{beta}^2$

Parametri

<i>alpha</i>	- parametro di forma
<i>beta</i>	- parametro di scala

Restituisce

realizzazione di una gamma di parametro di forma *alpha* e di parametro di scala *beta*

7.17.2.10 `double omprng::rnorm (double , double)`

Generazione da Gaussiana.

Parametri

<i>mu</i>	- media
<i>sigma</i>	- deviazione standard

Restituisce

realizzazione di una gaussiana con media *mu* e varianza sigma^2

7.17.2.11 `double omprng::runif ()`

Generazione da Uniforme tra 0 ed 1.

Restituisce

realizzazione di una uniforme tra 0 ed 1

7.17.2.12 `double omprng::runif (double a, double b)`

Generazione da Uniforme tra due valori fissati.

Parametri

<i>a</i>	- estremo inferiore
<i>b</i>	- estremo superiore

Restituisce

realizzazione di una uniforme tra a e b

7.17.2.13 unsigned int omprng::runifdiscrete (unsigned int *N*)

Campionamento da una variabile aleatoria discreta uniforme con supporto 0:(N-1)

Parametri

<i>N</i>	- numero di classi per il campionamento discreto
----------	--

Restituisce

x - realizzazione di una variabile aleatoria discreta uniforme supporto 0:(N-1)

7.17.2.14 void omprng::setNumThreads (int *nt*)

Imposta il numero di thread.

Parametri

<i>NumThread</i>	- numero di thread
------------------	--------------------

La documentazione per questa classe è stata generata a partire dai seguenti file:

- [omprng.hpp](#)
- [omprng.cpp](#)

7.18 Riferimenti per la struct PI

Pesi dei cluster in ogni gruppo.

```
#include <Struct.hpp>
```

Attributi pubblici

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

7.18.1 Descrizione dettagliata

Pesi dei cluster in ogni gruppo.

La documentazione per questa struct è stata generata a partire dal seguente file:

- [Struct.hpp](#)

7.19 Riferimenti per la classe RngStream

Membri pubblici

- **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)

Membri pubblici statici

- static bool **SetPackageSeed** (const unsigned long seed[6])

Membri privati

- double **U01** ()
- double **U01d** ()

Attributi privati

- double **Cg** [6]
- double **Bg** [6]
- double **Ig** [6]
- bool **anti**
- bool **incPrec**
- std::string **name**

Attributi privati statici

- static double **nextSeed** [6]

7.19.1 Documentazione dei membri dato

7.19.1.1 double RngStream::nextSeed [static],[private]

Valore iniziale:

```
=  
{  
  12345.0, 12345.0, 12345.0, 12345.0, 12345.0, 12345.0  
}
```

La documentazione per questa classe è stata generata a partire dai seguenti file:

- rngstream.hpp
- rngstream.cpp

7.20 Template per la classe TypeCategorical< DIM >

Classe dei tipi per dati con verosimiglianza categorica.

```
#include <Type.hpp>
```

Tipi pubblici

- using **THETA** = vector< double >
Vettore dei parametri latenti dei cluster o sub-cluster.
- using **Point** = unsigned int
Dato singolo.
- using **HYP** = vector< double >
Vettore degli iperparametri dei parametri latenti dei cluster o sub-cluster.
- using **STAT** = vector< unsigned int >
Vettore delle statistiche, aggiornamenti degli iperparametri dei parametri latenti dei cluster o sub-cluster.

7.20.1 Descrizione dettagliata

```
template<unsigned int DIM = 1>class TypeCategorical< DIM >
```

Classe dei tipi per dati con verosimiglianza categorica.

Data

Febbraio 2016

La documentazione per questa classe è stata generata a partire dal seguente file:

- [Type.hpp](#)

Capitolo 8

Documentazione dei file

8.1 Riferimenti per il file Cluster.hpp

Strutture dei dati per la gestione dei cluster in base al modello scelto. In queste classi si definisce solo come gestire i parametri latenti e gli altri parametri del cluster. Non sono presenti campionamenti, ma solo metodi che stampano e fissano i valori dei cluster e sub-cluster.

```
#include <vector>
#include <iostream>
#include "Type.hpp"
```

Composti

- class [GenericCluster](#)< [ClassType](#), DIM >

Modello Generico di Cluster.

- class [CategoricalCluster](#)

Gestione informazioni cluster e sub-cluster per dati di verosimiglianza Categorical.

8.1.1 Descrizione dettagliata

Strutture dei dati per la gestione dei cluster in base al modello scelto. In queste classi si definisce solo come gestire i parametri latenti e gli altri parametri del cluster. Non sono presenti campionamenti, ma solo metodi che stampano e fissano i valori dei cluster e sub-cluster.

Data

Febbraio 2016

8.2 Riferimenti per il file Document.hpp

In questo file sono presenti le classi che gestiscono i documenti o, piu' in generale, i gruppi di dati. La classe generica fornisce l'interfaccia comune, mentre le classi derivate e specializzate sono specifiche del modello.

```
#include <unordered_map>
#include <utility>
#include <vector>
#include "Functions.hpp"
#include "Type.hpp"
#include <iostream>
#include <algorithm>
#include <iomanip>
#include <tuple>
#include <fstream>
```

Composti

- class [GenericDocument< Type, DIM >](#)
Classe generica per i gruppi.
- class [CategoricalDocument< DIM >](#)
Classe derivata per il modello Dirichlet-Categorical.

8.2.1 Descrizione dettagliata

In questo file sono presenti le classi che gestiscono i documenti o, piu' in generale, i gruppi di dati. La classe generica fornisce l'interfaccia comune, mentre le classi derivate e specializzate sono specifiche del modello.

Data

Febbraio 2016

8.3 Riferimenti per il file Functions.hpp

```
#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>
```

Composti

- struct [greater_for_pair](#)
Criterio di confronto tra due elementi che sono coppie (unsigned int, double), da utilizzare nel sort di un vettore; il confronto è sul secondo valore nella coppia.

Funzioni

- void [ComputeLogStirlingNumbers](#) (unsigned int N, vector< long double > &logstirling)

Calcola i numeri di Stirling di prima specie $s(n,m)$, per $n = 0, \dots, N$ e li memorizza nel vettore in ingresso. Per definizione: $s(0,0) = s(1,1) = 1$, $s(n,0) = 0$ per $n > 0$, $s(n,m) = 0$ per $m > n$, $s(n,m) = s(n-1,m-1) + (n-1)*s(n-1,m)$. Il calcolo e' in scala logaritmica per ottenere maggior precisione.

- `template<typename T>`

`T Kahan_algorithm (vector< T > &numbers)`

Kahan_algorithm Fa le somme di vettori e riduce l'errore numerico, dovuto all'arrotondamento che fa la macchina.

- `unsigned int Antoniak (double alpha, double beta, unsigned int njk, const std::vector< long double > &LogStirling, omprng &Gen)`

Antoniak si occupa del campionamento dei tavoli in ogni gruppo, equazione 3.3 Relazione_Parisi_Perego.pdf.

- `unsigned int FindBestNumTable (double alpha, unsigned int K, unsigned int njk, vector< long double > &StirlingNumber)`

Campionare gli $m_{j,b}$, ovvero il numero di tavoli che in un documento j hanno il cluster b campiona i tavoli temporanei dall'equazione 3.24 Relazione_Parisi_Perego.pdf.

8.3.1 Descrizione dettagliata

In questo file sono raccolte tutte quelle funzioni di supporto all'algoritmo che non sono specifiche del modello scelto.

Data

Febbraio 2016

8.3.2 Documentazione delle funzioni

8.3.2.1 `unsigned int Antoniak (double alpha, double beta, unsigned int njk, const std::vector< long double > & LogStirling, omprng & Gen)`

Antoniak si occupa del campionamento dei tavoli in ogni gruppo, equazione 3.3 Relazione_Parisi_Perego.pdf.

Parametri

<i>Alpha</i>	- parametro di concentrazione del processo di dirchlet che governa i cluster nel gruppo
<i>Beta</i>	- peso globale del cluster k di cui vogliamo campionare i tavoli
<i>njk</i>	- numero di elementi del gruppo j nel cluster k
<i>LogStirling</i>	- Vettore dei numeri di Stirling in scala logaritmica
<i>Gen</i>	- Generatore di numeri casuali

Restituisce

numero di tavoli che nel gruppo j servono il piatto k

8.3.2.2 `void ComputeLogStirlingNumbers (unsigned int N, vector< long double > & logstirling)`

Calcola i numeri di Stirling di prima specie $s(n,m)$, per $n = 0, \dots, N$ e li memorizza nel vettore in ingresso. Per definizione: $s(0,0) = s(1,1) = 1$, $s(n,0) = 0$ per $n > 0$, $s(n,m) = 0$ per $m > n$, $s(n,m) = s(n-1,m-1) + (n-1)*s(n-1,m)$. Il calcolo e' in scala logaritmica per ottenere maggior precisione.

Parametri

<i>N</i>	- valore massimo di n
<i>logstirling</i>	- vettore in cui memorizzare i numeri di Stirling in scala logaritmica

8.3.2.3 unsigned int FindBestNumTable (double *alpha*, unsigned int *K*, unsigned int *njk*, vector< long double > &
StirlingNumber)

Campionare gli \tilde{m}_{jb} , ovvero il numero di tavoli che in un documento *j* hanno il cluster *b* campiona i tavoli temporanei dall'equazione 3.24 Relazione_Parisi_Perego.pdf.

Parametri

<i>alpha</i>	- parametro di concentrazione del processo di dirchlet che governa i cluster nel gruppo
<i>K</i>	- numero corrente dei cluster
<i>njk</i>	- numero di elementi del gruppo j nel cluster k
<i>StirlingNumber</i>	- Vettore dei numeri di Stirling in scala logaritmica

8.3.2.4 `template<typename T > T Kahan_algorithm (vector< T > & numbers)`

Kahan_algorithm Fa le somme di vettori e riduce l'errore numerico, dovuto all'arrotondamento che fa la macchina.

Parametri

<i>numbers</i>	- elementi da sommare tra di loro
----------------	-----------------------------------

Restituisce

somma

8.4 Riferimenti per il file HDP_MCMC.hpp

```
#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>
```

Composti

- class [HDP_MCMC< MODEL, DOCUMENT, DIM >](#)
[HDP_MCMC](#).

8.4.1 Descrizione dettagliata

Classe di [HDP_MCMC](#) per l'esecuzione dell'algoritmo

Data

Febbraio 2016

8.5 Riferimenti per il file Model.hpp

Classe che si occupa di gestire tutti i cluster, campionamenti che dipendono dal modello scelto e funzioni che dipendono dal modello.

```
#include "Cluster.hpp"
#include "Functions.hpp"
#include "omprng.hpp"
#include <vector>
#include <unordered_map>
#include <utility>
#include <cmath>
#include <iostream>
#include <sstream>
#include <fstream>
```

Composti

- class [ModelGeneric< Type, DIM >](#)
Modello Generico di Model.
- class [CategoricalModel< DIM >](#)
Modello Dirichlet-Categorical.

8.5.1 Descrizione dettagliata

Classe che si occupa di gestire tutti i cluster, campionamenti che dipendono dal modello scelto e funzioni che dipendono dal modello.

Data

Febbraio 2016

8.6 Riferimenti per il file omprng.hpp

Generatore di numeri casuali in parallelo per OpenMP.

```
#include <omp.h>
#include <iostream>
#include "rngstream.hpp"
#include "sys/time.h"
#include <cmath>
#include <vector>
```

Composti

- class [omprng](#)
Libreria Omprng per la generazione dei numeri casuali in OpenMp.

8.6.1 Descrizione dettagliata

Generatore di numeri casuali in parallelo per OpenMP.

8.7 Riferimenti per il file PosteriorAnalysis.hpp

```
#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"
```

Composti

- class [GenericPosteriorAnalysis< Type, DIM >](#)

Classe generica per l'analisi a posteriori. Classe virtuale dove tutti i metodi sono null. Ogni classe che eredita deve definire tutti i metodi della classe base e, se necessario, può aggiungere altri metodi. Invoca gli script R per l'analisi delle catene MCMC. Calcola l'indice LPML, riconosce i topic e individua il miglior clustering secondo il criterio dei minimi quadrati.

- class [CategoricalPosteriorAnalysis< DIM >](#)

Namespace

- [CoeffSimilitudine](#)

Namespace per il calcolo del coefficiente di similitudine. Contiene tre funzioni template utili al calcolo del coefficiente di similitudine tra vettori di un qualsiasi tipo.

Funzioni

- `template<typename T >`
`double CoeffSimilitudine::VectorNorm (vector< T > &v)`
Calcola la norma di un vettore.
- `template<typename T >`
`T CoeffSimilitudine::ScalarProduct (vector< T > &v1, vector< T > &v2)`
Calcola il prodotto scalare tra due vettori.
- `template<typename T >`
`double CoeffSimilitudine::Coeff (vector< T > &v1, vector< T > &v2)`
Calcola il coefficiente di similitudine tra due vettori.

8.7.1 Descrizione dettagliata

Contiene le classi che gestiscono l'analisi a posteriori dei risultati prodotti dall'algoritmo. Per ogni modello esiste una classe derivata e specializzata. La classe generica fornisce l'interfaccia comune a tutte le classi specifiche di un modello.

Data

Febbraio 2016

8.8 Riferimenti per il file Struct.hpp

Raccolta delle strutture che vengono utilizzate nei metodi di [HDP_MCMC.hpp](#) Si e' scelto di creare un file a parte con la dichiarazione delle strutture, perche' il loro utilizzo e' comune a piu' metodi.

```
#include <vector>
#include <utility>
#include <unordered_map>
#include "Type.hpp"
```

Composti

- struct [BETA](#)
Pesi globali dei cluster.
- struct [NJK](#)
Numero degli elementi del gruppo j che sono nel cluster k.
- struct [PI](#)
Pesi dei cluster in ogni gruppo.
- struct [NUMTABLE](#)
Tavoli.
- struct [C](#)
Statistiche.
- struct [CLUSTER](#)
- struct [DATACOUNT](#)

Ridefinizioni di tipo (typedef)

- using **POINT** = [TypeCategorical](#)< 1 >::Point
- using **STAT** = [TypeCategorical](#)< 1 >::STAT

8.8.1 Descrizione dettagliata

Raccolta delle strutture che vengono utilizzate nei metodi di [HDP_MCMC.hpp](#) Si e' scelto di creare un file a parte con la dichiarazione delle strutture, perche' il loro utilizzo e' comune a piu' metodi. Ogni volta che si aggiunge un nuovo modello e dei nuovi tipi di dati, bisogna aggiornare la lista in cui si rinomina i tipi di dati

Data

Febbraio 2016

8.9 Riferimenti per il file Type.hpp

Strutture dati per i tipi di modello e dei cluster.

```
#include <vector>
#include <iostream>
```

Composti

- class `TypeCategorical< DIM >`

Classe dei tipi per dati con verosimiglianza categorica.

8.9.1 Descrizione dettagliata

Strutture dati per i tipi di modello e dei cluster.