

Phrase

+ Attribute<string> label

+ Attribute<uint> dimension

+ Attribute<uint> dimension_input

+ vector<string> column_names

uint AllocationBlockSize

bool own_memory_

bool bimodal_

bool empty_

uint length_

uint input_length_

uint output_length_

uint max_length_

float **data_

EventGenerator<PhraseEvent> events

+ Phrase(MemoryMode mem, Multimodality mod)

+ Phrase(Phrase& src)

+ Phrase(Json::Value& root)

+ operator=(Phrase& src) : Phrase&

+ ~Phrase()

+ ownMemory() : bool

+ bimodal() : bool

+ size() : uint

+ inputSize() : uint

+ outputSize() : uint

+ empty() : bool

+ getValue(uint index, uint dim) : float

+ getPointer(uint index) : float *

+ getPointer_input(uint index) : float *

+ getPointer_output(uint index) : float *

+ connect(float *ptr, uint length)

+ connect(float *ptr_to_input, float *ptr_to_output, uint length)

+ connect_input(float *ptr, uint length)

+ connect_output(float *ptr, uint length)

+ disconnect()

+ record(vector<float> obs)

+ record_input(vector<float> obs_in)

+ record_output(vector<float> obs_out)

+ clear()

+ clearInput()

+ clearOutput()

+ mean() : vector<float>

+ toJson() : Json::Value

+ fromJson(Json::Value root)

+ mean() : vector<float>

+ standardDeviation() : vector<float>

+ minmax() : vector<pair<float, float>>

+ rescale(vector<float> offset, vector<float> gain)

trim()

reallocateLength()

onAttributeChange(AttributeBase *attr_pointer)

TrainingSet

+ Attribute<uint> dimension

+ Attribute<uint> dimension_input

+ Attribute<vector<string>> column_names

own_memory_

bool bimodal_

set<string> labels_

map<int, shared_ptr<Phrase>> phrases_

map<string, TrainingSet> sub_training_sets_

+ TrainingSet(MemoryMode mem, Multimodality mod)

+ TrainingSet(TrainingSet& src)

+ TrainingSet(Json::Value& root)

+ operator=(TrainingSet& src) : TrainingSet&

+ ~TrainingSet()

+ ownMemory() : bool

+ bimodal() : bool

+ empty() : bool

+ size() : uint

+ begin() : map<int, shared_ptr<Phrase>>::iterator

+ end() : map<int, shared_ptr<Phrase>>::iterator

+ rbegin() : map<int, shared_ptr<Phrase>>::reverse_iterator

+ rend() : map<int, shared_ptr<Phrase>>::reverse_iterator

+ cbegin() : map<int, shared_ptr<Phrase>>::const_iterator

+ cend() : map<int, shared_ptr<Phrase>>::const_iterator

+ crbegin() : map<int, shared_ptr<Phrase>>::const_reverse_iterator

+ crend() : map<int, shared_ptr<Phrase>>::const_reverse_iterator

+ addPhrase(int index, string label)

+ addPhrase(int index, Phrase phrase)

+ addPhrase(int index, shared_ptr<Phrase> phrase)

+ removePhrase(int index)

+ removePhrasesOfClass(string& label)

+ getPhrase(int index) : shared_ptr<Phrase>

+ getPhrasesOfClass(string label) : TrainingSet *

+ clear()

+ labels() : set<string>

+ toJson() : Json::Value

+ fromJson(Json::Value root)

+ mean() : vector<float>

+ standardDeviation() : vector<float>

+ minmax() : vector<pair<float, float>>

+ rescale(vector<float> offset, vector<float> gain)

+ normalize()

onPhraseEvent(PhraseEvent& e)

onAttributeChange(AttributeBase *attr_pointer)

update()

GMM (Model<SingleClassGMM, GMM>)

+ Results<GMM> results

+ GMM(bool bimodal)

+ GMM (GMM& src)

+ GMM(Json::Value& root)

+ operator=(GMM& src)

+ reset()

+ filter(vector<float>& obs)

updateResults()

HierarchicalHMM (Model<SingleClassHMM, HMM>)

+ Results<HMM> results

+ vector<double> prior

+ vector<double> exit_transition

+ vector<vector<double>> transition

bool forward_initialized_

vector<double> frontier_v1_

vector<double> frontier_v2_

+ HierarchicalHMM(bool bimodal)

+ HierarchicalHMM(HierarchicalHMM& src)

+ HierarchicalHMM(Json::Value& root)

+ operator=(HierarchicalHMM& src)

+ removeClass(string& label)

+ clear()

+ addExitPoint(int state, float proba)

+ reset()

+ filter(vector<float>& obs)

+ toJson() : Json::Value

+ fromJson(Json::Value& root)

joinTraining()

updateTransitionParameters()

updatePrior()

updateTransition()

updateExitProbabilities()

addModelForClass(string& label)

normalizeTransitions()

forward_init(vector<float>& obs)

forward_update(vector<float>& obs)

likelihoodAlpha(int exitNum, vector<double>& likelihoodVector)

updateResults()

Model

+ shared_ptr<SharedParameters> shared_parameters

+ Configuration<ModelType> configuration

+ EventGenerator<TrainingEvent> training_events

+ map<string, SingleClassModel> models

map<string, thread> training_threads_

atomic<bool> cancel_required_

atomic<bool> is_training_

atomic<bool> is_joining_

uint models_still_training_

mutex event_mutex_

+ Model(bool bimodal)

+ Model(Model& src)

+ Model(Json::Value& root)

+ ~Model()

+ size() : uint

+ hasClass(string& label) : bool

+ getIndex(string& label) : int

+ removeClass(string& label)

+ clear()

+ trained() : bool

+ training() : bool

+ train(TrainingSet* set)

+ train(TrainingSet *set, string& label)

+ cancelTraining()

+ cancelTraining(string& label)

+ reset()

+ filter(vector<float>& observation)

+ toJson() : Json::Value

+ fromJson(Json::Value& root)

joinTraining()

onTrainingEvent(TrainingEvent& e)

checkTraining()

checkConfigurationChanges()

addModelForClass(string& label)

SingleClassGMM

+ ClassParameters<GMM> parameters

+ ClassResults<GMM> results

+ vector<GaussianDistribution> components

+ vector<float> mixture_coeffs

+ vector<double> beta

vector<double> current_regularization

+ SingleClassGMM(shared_ptr<SharedParameters> p)

+ SingleClassGMM(SingleClassGMM& src)

+ SingleClassGMM(shared_ptr<SharedParameters> p, Json::Value& root)

+ operator=(SingleClassGMM& src) : SingleClassGMM&

+ reset()

+ filter(vector<float>& obs) : double

+ toJson() : Json::Value

+ fromJson(Json::Value& root)

allocate()

obsProb(float *obs, int mixtureComponent) : double

obsProb_input(float *obs_input, int mixtureComponent) : double

obsProb_bimodal(float *obs_input, float *obs_output, int mixtureComponent) : double

emAlgorithmInit(TrainingSet *set)

emAlgorithmUpdate(TrainingSet *set) : double

initParametersToDefault(vector<float>& dataStddev)

initMeansWithKMeans(TrainingSet *set)

initCovariances_fullyObserved(TrainingSet *set)

normalizeMixtureCoeffs()

addCovarianceOffset()

updateInverseCovariances()

likelihood(vector<float>& obs, vector<float>& obs_output) : double

regression(vector<float>& obs_input)

updateResults()

singleClassProbabilisticModel

+ string label

+ shared_ptr<SharedParameters> shared_parameters

+ EventGenerator<TrainingEvent> training_events

+ TrainingEvent training_status

CircularBuffer<double> likelihood_buffer_

mutex training_mutex_

bool is_training_

bool cancel_training_

+ SingleClassProbabilisticModel(shared_ptr<SharedParameters> p)

+ SingleClassProbabilisticModel(SingleClassProbabilisticModel& src)

+ SingleClassProbabilisticModel(shared_ptr<SharedParameters> p, Json::Value& root)

+ operator=(SingleClassProbabilisticModel& src) : SingleClassProbabilisticModel&

+ ~SingleClassProbabilisticModel()

+ isTraining() : bool

+ train(TrainingSet *set)

+ cancelTraining()

+ reset()

+ filter(vector<float>& obs) : double

+ toJson() : Json::Value

+ fromJson(Json::Value& root)

allocate()

emAlgorithmInit(TrainingSet *set)

emAlgorithmUpdate(TrainingSet *set) : double

emAlgorithmTerminate()

emAlgorithmHasConverged(int step, double log_prob, double old_log_prob) : bool

cancelTrainingIfRequested() : bool

check_training()

SingleClassHMM

+ ClassParameters<HMM> parameters

+ ClassResults<HMM> results

+ vector<double> alpha

+ vector<double> alpha_h[3]

+ vector<SingleClassGMM> states

+ vector<float> prior

+ vector<float> transition

bool forward_initialized_

vector<double> previous_alpha_

vector<double> beta_

vector<double> previous_beta_

vector<vector<double>> gamma_sequence_

vector<vector<double>> epsilon_sequence_

vector<vector<vector<double>>> gamma_sequence_per_mixture_

vector<double> alpha_seq_

vector<double> beta_seq_

vector<double> gamma_sum_

vector<double> gamma_sum_per_mixture_

bool is_hierarchical_

vector<float> exit_probabilities_

int window_minindex_

int window_maxindex_

double window_normalization_constant

+ SingleClassHMM(shared_ptr<shared_parameters> p)

+ SingleClassHMM(SingleClassHMM& src)

+ SingleClassHMM(shared_ptr<SharedParameters> p, Json::Value& root)

+ operator=(SingleClassHMM& src) : SingleClassHMM&

+ addExitPoint(int stateIndex, float proba)

+ reset()

+ filter(vector<float>& obs) : double

+ toJson() : Json::Value

+ fromJson(Json::Value& root)

allocate()

initParametersToDefault(vector<float>& dataStddev)

initMeansWithAllPhrases(TrainingSet *set)

initCovariances_fullyObserved(TrainingSet *set)

initMeansCovariancesWithGMMEM(TrainingSet *set)

setErgodic()

setLeftRight()

normalizeTransitions()

forward_init(float *obs, float *obs_output) : double

forward_update(float *obs, float *obs_output) : double

backward_init(double ct)

backward_update(double ct, float *obs, float *obs_output)

emAlgorithmInit(TrainingSet *set)

emAlgorithmTerminate()

emAlgorithmUpdate(TrainingSet *set) : double

baumWelch_forwardBackward(shared_ptr<Phrase> currentPhrase, int phraseIndex) : double

baumWelch_forward_update(vector<double>::iterator obs_likelihoos) : double

baumWelch_backward_update(double ct, vector<double>::iterator obs_likelihoos)

baumWelch_gammaSum(TrainingSet *set)

baumWelch_estimateMixtureCoefficients(TrainingSet *set)

baumWelch_estimateMeans(TrainingSet *set)

baumWelch_estimateCovariances(TrainingSet *set)

baumWelch_estimatePrior(TrainingSet *set)

baumWelch_estimateTransitions(TrainingSet *set)

addCyclicTransition(double proba)

updateAlphaWindow()

regression(vector<float>& obs_input)

updateResults()

updateExitProbabilities(float *exit_probas)