index
Tensors c:\users\pabst\desktop\vorominkf\finalcode\tensors.py

Module: Tensors
Author: Dominik Pabst

This module provides some tools for handeling tensors for the module VorominkEstimation

A tensor of rank r is a linear mapping T, which takes r vectors from R^d.
A symmetric tensor T is determined by the values T(e_i1,...,e_ir), where
1 <= i1 <= ... <= ir <= d and e_i is the i-th standard vector in R^d.
In this code a tensor is represented by a dictionary, which has a key (i1,...,ir) for each choice of i1,...,ir.
For example the value corresponding to the key (1,1,2) represents T(e1,e1,e2).

Modules

Tools itertools math numpy

Functions

Tvoronoi_measure(in_data, eta, R, res, r, s)
Estimates the Voronoi tensors with rank parameters r,s of the input data for different radii

Args:
in_data (numpy.ndarray): Input data
eta (list): Grid process (entries are the individual points)
R (list): Entries are the radii, for which the Voronoi tensor is estimated
res (float): Resolution of the grid process (compare Tools.grid_process)
r,s (int): Rank parameters of the tensors, that will be estimated

Returns:
list: Entries are the estimated Voronoi tensors. Therefore the list has the same length as the input R.

Tvoronoi_measure_1(in_data, eta, R, res, r, s)
Help function for Tvoronoi_measure

Tvoronoi_measure_2(in_data, eta, R, res, r, s)
Help function for Tvoronoi_measure

emptyTensor(dim, rank, LIST=False)
Creates an empty tensor (a tensor whose entries are all zero)

Args:
dim (int): Dimension of the tensor (Minkowski tensor of a set in R^d has dimension d)
rank (int): Rank of the tensor (Minkowski or Voronoi tensor with rank parameters r,s has rank r+s)
LIST (boolean,optional): If True the entries of the tensor are not 0, but empty lists.
Can be used to bundle several tensors.

Returns:
dict: Empty tensor

evaluate(T, x)
Computes the values of a tensor at a given argument

Args:
T (dict): Tensor, which shall be evaluated
x (list): Entries are points in R^d, where d is the dimension of t. Length has to be the rank of T.

Returns:
float: Computed value

get_tensors(V, R, dim, r, s)
Computes the Minkowski tensors from the Voronoi tensors by solving a Least Squares Problem

Args:
V (list): Entries are the Voronoi tensors (size should be at leat dim+1)

R (list): Corresponding radii to the Voronoi tensors in V
dim (int): Dimension of the set (resp. the data) for which the Voronoi tensors are given
r,s (int): Rank parameters of the Voronoi tensors

Returns:
list: Entries are the computed Minkowski tensors Phi_d,...,Phi_0 (in this order)

perm2(n)
Help function for Tvoronoi_measure

tensorproduct1(x, y, e, r, s)
Help function for Tvoronoi_measure

tensorproduct2(x, y, e, r, s)
Help function for Tvoronoi_measure

tensorsum1(dist, ind, in_data, eta, r, s, R)
Help function for Tvoronoi_measure

tensorsum2(dist, ind, in_data, eta, r, s, R)
Help function for Tvoronoi_measure

Data

gamma = <ufunc 'gamma'>