Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen · Bolzano

Adaptive Density Estimation

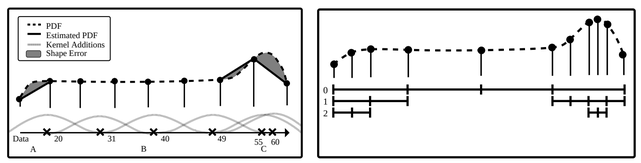
Arturas Mazeika, Michael H. Böhlen, and Andrej Taliun
{arturas,boehlen,taliun}@inf.unibz.it

Poster # 1/2

1. Introduction

Density information is common statistical information that is used for approximate query answering, query optimization, clustering, etc. Typically, histogram are used to roughly estimate the density. Kernel based estimators generalize histograms. They ensure that the estimated density function is continuous and that derivatives exist. This demonstration demos the APDF tree [1, 2], an adaptive tree that supports the effective and efficient computation of continuous density information.

Figure 1 (a) illustrates kernel additions. The data is illustrated by the six crosses at the bottom of Figure 1 (a). The data points represent the age of six persons. We draw a kernel function around each data point. The addition of all kernels yields the estimation of the probability density function (PDF, dashed line).



(a) Uniform Partition (b) The APDF Tree

Figure 1: *The Idea of the APDF Tree (1D)*

The kernel additions are computed on a uniform partition, and the values of the PDF are interpolated between partition points. The shaded regions quantify the *shape error* (SE): the difference between the PDF and the linear interpolations. Figure 1 (a) illustrates the key problem with a uniform partitioning: a fine partitioning yields a good but expensive estimation whereas a coarse partitioning yields a bad but fast estimation. The ADPF tree is an adaptive tree structure that yields a fast and precise estimation of the PDF.

Figure 1 (b) illustrates the APDF tree for the same dataset. The APDF tree allocates more partition points in non-linear areas of the PDF and fewer partition points in linear areas of the PDF. This yields not only a bounded, but a tight control of the shape error, and makes the estimation fast and precise. The poster describes the computation of the APDF partitions with a tight control of the shape error and demos the implementation for different datasets.

2. Overview of the APDF Method

We use the PlaneSphere dataset (cf. Figure 2) to illustrate the construction of the APDF tree. The dataset consists of two structures: a plane (distributed in the unit square) and a sphere (2D normal distributed data in the bottom-right corner of the unit square).

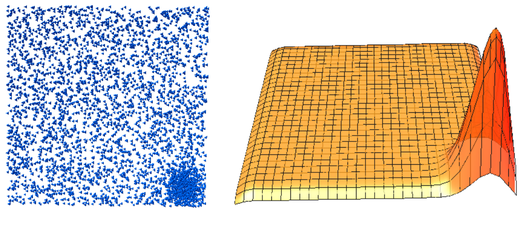
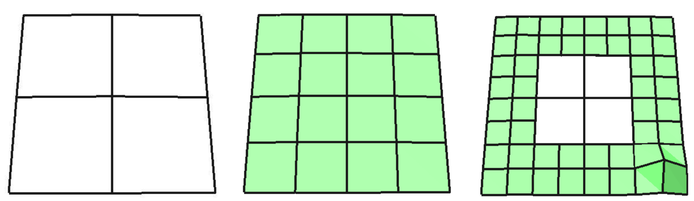
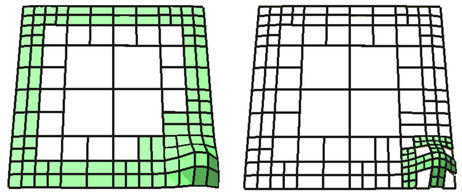


Figure 2: *The PlaneSphere Data and its PDF*

Figure 3 illustrates the creation of the APDF tree for the PlaneSphere dataset. The computation starts with a sparse uniform partition and performs the kernel additions on this partition (cf. Figure 3 (a). The APDF method then identifies areas where the PDF is non linear and splits these areas. In our example all areas are split after the first iteration (cf. Figure 3 (b)). This process continues until the approximated shape error is uniformly low in all areas of the APDF tree. In iteration 2 (cf. Figure 3 (c)) only the areas at the borders are split. Since the PDF is linear in the center of the space, no additional points are introduced there. The 3rd iteration (cf. Figure 3 (d)) adds further partition points at the borders of the universe. The rectangles are split in X and Y direction in the corners of the universe since the PDF is non-linear in both directions in the corners. The rest of the rectangles are split in one direction since the PDF is non-linear only in one direction. The 4th iteration completes the creation of the APDF tree. Only the area with the sphere is split. The PDF is non-linear at the peak of the PDF and at the boundary of the sphere.



(a) Initial Partition (b) 1st Iteration (c) 2nd Iteration



(d) 3rd Iteration (e) 4th Iteration

Figure 3: *Creation of the APDF Tree*

3. Individual Steps

This section zooms into the 3rd iteration and presents the core steps of one iteration of the APDF tree construction: *split*, *tree optimization*, *kernel additions*, *unsplit*, and the second tree optimization.

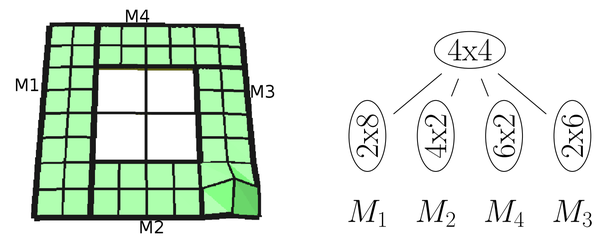


Figure 4: *Start of the Iteration (cf. Figure 3 (c))*

Figure 4 shows the APDF tree at the beginning of the iteration. The tree consists of the root node with a uniform partition of size 4x4 (4 partition points per coordinate). The 12 outer rectangles of the root are split and the 4 inner rectangles are not split. The 12 split rectangles are organized into 4 nodes with a local uniform partition of sizes 2x8, 4x2, 6x2, and 2x6. The tree illustration shows the structure of the nodes and the parent-child connections. The areas that were split in the previous iteration are colored green. These are the only areas that can have a too high shape error.

Splitting is the first step in the iteration. The split step processes the set of nodes C that were introduced in the previous step. It scans C and predicts if the shape error is too high along one of the coordinates. If the shape error is too high, it splits along the respective coordinate. Since the split predicts the shape error it can (and often will) introduce too many splits. The unsplit step is later used to remove unnecessary splits.

Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen · Bolzano

Adaptive Density Estimation

Arturas Mazeika, Michael H. Böhlen, and Andrej Taliun
{arturas,boehlen,taliun}@inf.unibz.it

Poster # 2/2

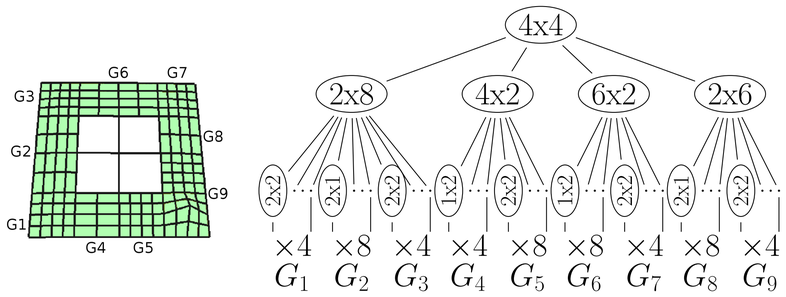


Figure 5: *Split*

Figure 5 illustrates the result of the split step. All areas were split further. In the corners of the unit square the PDF is non-linear in both dimensions, therefore we split the areas according to both dimensions. In the other areas the PDF is non-linear according to one dimension only, therefore the rectangles are split according to one dimension only. The split step produces a lot of small rectangles (groups of rectangles are indicated by G_i). The tree optimization step groups the rectangles into fewer nodes.

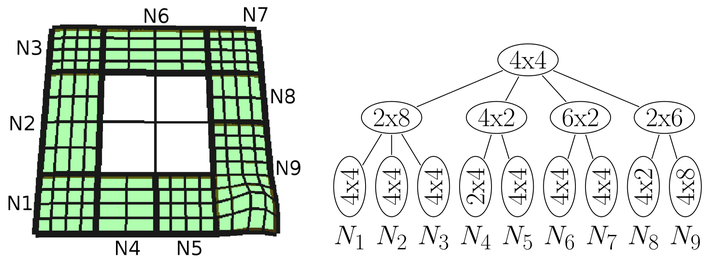
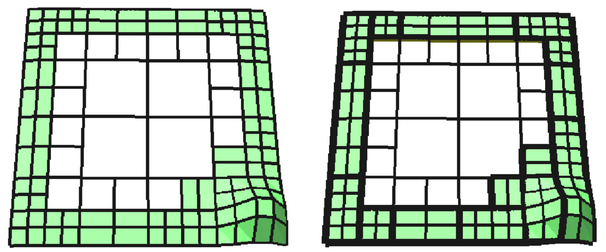


Figure 6: *Tree Optimization*

The tree optimization (TO) step is applied to the APDF tree to speed up the computation of subsequent kernel additions. The TO step scans the nodes C that were split and groups nodes of the same granularity into new nodes of *local uniform partitions* (LUP). The algorithm produces hyper-rectangular shaped nodes (cf. Section 2). Figure 6 shows the APDF tree after the tree optimization. The TO step reorganizes the 52 nodes introduced by the split step into 9 nodes. This effectively reduces the time for the subsequent kernel additions. The kernel addition step updates the new partition points with kernel additions.



(a) Unsplit (b) Tree Optimization (TO)

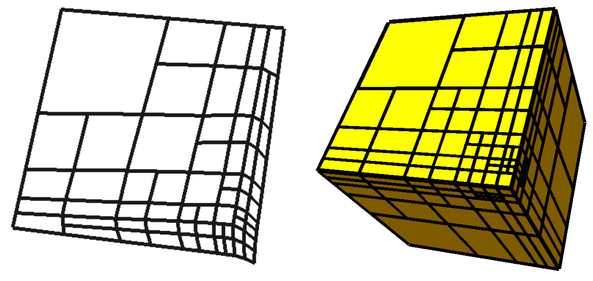
Figure 7: *Unsplit and Tree Optimization Steps*

The unsplit (US) step scans the nodes and removes partition points that did not increase the precision of the estimator. At these points the shape error was already low and the split step over-split the area. The partition after the US step is illustrated in Figure 7 (a).

The second tree optimization completes the iteration (cf. Figure 7 (b)). Again contiguous areas of the same shape are grouped into the same node to get large areas with a local uniform partition.

4. Evaluation of the APDF Tree

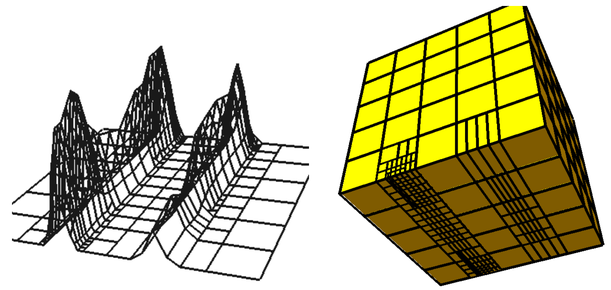
This section describes the third part of the demo: the comparison of APDF partitions for different datasets and dimensions. Figure 8 illustrates the APDF trees for the Linear dataset. The distribution of the data is linear according to each coordinate. The density increases from corner $(0, 0, 0)$ towards the opposite corner of the universe where it decreases very fast.



(a) 2D APDF (b) 3D APDF

Figure 8: *The Linear Dataset*

The dataset nicely illustrates the directional splits of the APDF tree. The APDF tree does not allocate any partition points in the linear area of the PDF, and introduces directional splits in non-linear areas of the PDF.



2D APDF 3D APDF

Figure 9: *The Click-Stream Dataset*

Figure 9 illustrates the APDF tree for real world click stream data. Country, time of visit, and URL of the retrieved document are mapped to X, Y, and Z coordinates. The PDF reveals that most clicks come from two countries and that the countries have a different distribution along the time dimension. The APDF tree introduces directional splits in non-linear areas of the PDF and does not split the rest of the universe.

5. Summary

The poster discusses the efficient estimation of continuous density information, which is important statistical information that is used in many areas (e.g., approximate query answering, clustering, query optimization). Our estimator not only provides an upper but also a lower bound for the error. This allows to control the space and time complexity of the estimation. A variety of data sets will be available that allow to investigate the estimation of density information in general and the construction of the APDF tree in particular.

References

[1] A. Mazeika, M. H. Böhlen, A. Taliun Adaptive Density Estimation, Demo. In *VLDB*, 2006.

[2] APDF-DS Method. http://www.inf.unibz.it/dis/projects/3dvdm/.