DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Guided Research

Sparse Grid Point Coarsening for Classification

Jieyi Zhang

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Guided Research

Sparse Grid Point Coarsening for Classification

Author:	Jieyi Zhang
Supervisor:	kilian Roehner
Advisor:	Prof. Dr. Hans-Joachim Bungartz
Submission Date:	24th April 2019

Contents

1 Introduction

Classification based on density estimation has been widely discussed in the literature on statistical learning and pattern recognition. A point or an area in the space can be classified using the density functions which represent the popularity of the points from each class. The corresponding class label at a specific point should be the class with the largest density at the same coordinate.
The estimated density function is discretized by the adaptive sparse-grid-based density estimation method on basis functions centered at grid points rather than on the data points. [4] Thus, the costs of evaluating the estimated density function are independent from the number of data points.
Originally, the sparse grid method was developed to solve partial differential equations [7]. With the sparse grid method, we use density estimation to approximate the population density functions. A sparse grid is initialized for each class. In this work, the size and the structure of the sparse grids can be adaptive to the datasets and thus can be different from each other. The class label of a specific point is then determined by the dominant density value.
To optimize the sparse grid size while preserving the accuracy of the model, we developed a coarsening algorithm in the SGpp Library which reduces the grid size by coarsening the points located in area where the classification confidence is high enough [6]. This algorithm removes points according to the scores of each points evaluated and the coarsening configuration given by the users. Hence, the coarsening results of each sparse grid can be different from each other.
In this thesis, we will first demonstrate the basics of sparse grid and density estimation, then present the coarsening algorithm and the testing results.

2 Introduction of Sparse Grid

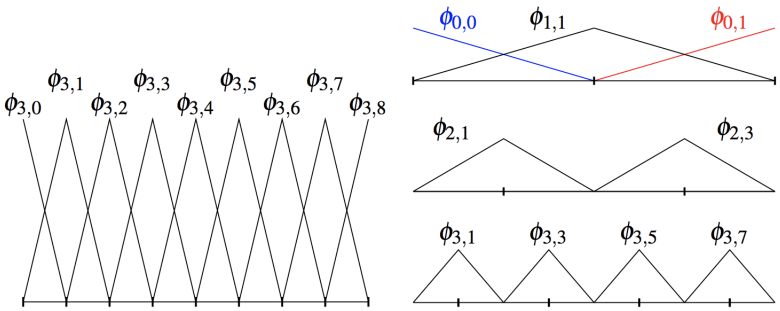

Figure 1: Nodal and hierarchical basis of level n = 3 [2].

First, we give a brief introduction about sparse grids, for more details, see [1]. The basic principle of sparse grids is a one-dimensional hierarchical system of basis functions, see Fig 1 as an example. One of the commonly used basic function is the hat function $\phi(x) := \max\{1-|x|, 0\}$. The one-dimensional hierarchical hat functions can then be expressed as $\phi_{l,i}$ depending on the level l and index i via translation and scaling as $\phi_{l,i}(x) := \phi\left(2^l x - i\right)$. Then,iIt can be extended to the d-dimensional case via a tensor product approach. See Figure 2 for an example in two dimension.

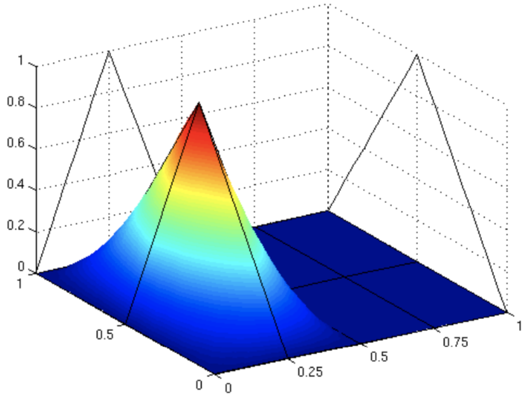

Figure 2: Two-dimensional basis function [2].

In the d-dimensional case, the level $l = (l_1, \ldots, l_d)$ and index $\boldsymbol{i} = (i_1, \ldots, i_d)$ become vectors and the corresponding basis function $\phi_{\boldsymbol{l},\boldsymbol{i}} := \prod_{k=1}^{d} \phi_{l_k,i_k}(\boldsymbol{x})$ is the product of the one-dimensional basis functions respectively. This results in a set of subspaces W_l for which the grid points are the Cartesian product of the one-dimensional ones with level l_k in dimension k. In Figure 3, the grids of the two-dimensional hierarchical increments W_l up to level 3 in each dimension is shown. We can only choose those subspaces W_l which contribute most to the overall solution according to the hierarchical scheme of increments. To this end, we can cut off the tableau in Figure 3 along its diagonal. This leads to a sparse grid space $V_\ell^{(1)}$ of level ℓ.

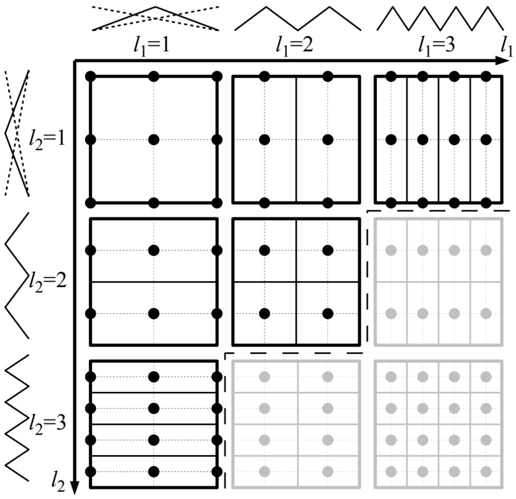

Figure 3: The grids of the two-dimensional hierarchical increments W_l up to level 3 in each dimension. According to the hierarchical scheme of increments, we choose the upper triangle of spaces shown in black which contribute most to the overall solution.

There are some efficient methods to construct sparse grids [1]. Since the sparse grid construction is a standard task in the context of sparse grids, and there are some libraries provide appropriate data structures and support corresponding methods, we are not discussing this part in details.
In order to optimize the size of sparse grids, we can apply spatial adaptivity. One way is to start with a coarse sparse grid and use proper algorithms to add points in those regions of the domain that are most important. In classification problems, an important domain should be the regions which near the borders of several classes or the classes are overlapping with each other. Figure 4 shows an example of this procedure. We define I as the set of all level index pairs corresponding to a sparse grid space $V_\ell^{(1)}$ and write a sparse grid function $f_N \in V_\ell$ as the linear combination

$$f_N(\boldsymbol{x}) = \sum_{(\boldsymbol{l},\boldsymbol{i}) \in I} \alpha_{\boldsymbol{l},\boldsymbol{i}} \phi_{\boldsymbol{l},\boldsymbol{i}}(x)$$

with $N = |I|$ basis functions $\phi_{\boldsymbol{l},\boldsymbol{i}}$ and coefficients $\alpha_{\boldsymbol{l},\boldsymbol{i}}$.

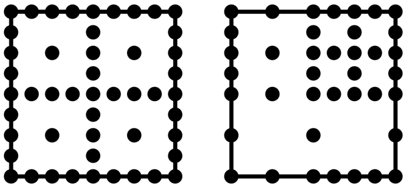

Figure 4: A regular sparse grid (left). A refined and coarsened sparse grid (right)[1]

3 Density-Estimation based Classification with Adaptive Sparse Grids

While learning Bayesian classifiers, continuous variables are normally handled by assuming that they follow a Gaussian distribution or by discretization. This work introduces a Bayesian classifier which estimates the true density of the data points from each class using adaptive sparse grid for supervised classification.
Here, we present our approach to approximating the likelihood in Bayesian classifier with sparse grid. Suppose we have a data set $\mathcal{S} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\} \subset R^d$ of samples drawn from an unknown distribution with unknown probability density function f. The goal is to construct an estimated density function $\hat{f}$ of f based on the data S. To this end, there are generally two methods, namely parametric and non-parametric density estimation. Here, the adaptive sparse-grid based density estimation belongs to the nonparametric methods, and it requires only the given data samples to estimate the density and no additional information about the data is necessary.
Suppose f_ϵ is an initial guess of the density function underlying the data $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$. Our task is to find $\hat{f}$ in a suitable function space V such that

$$\hat{f} = \underset{u \in V}{\operatorname{argmin}} \int_\Omega (u(\boldsymbol{x}) - f_\epsilon(\boldsymbol{x}))^2 \, \mathrm{d}\boldsymbol{x} + \lambda \|\mathrm{L}u\|_{L^2}^2.$$

Here, the left term ensures that the function $\hat{f}$ fits the initial guess f_ϵ as close as possible, while the right term $\|\mathrm{L}u\|_{L^2}^2$ is a penalty or regularization term imposing a smoothness constraint. $\lambda > 0$ is the regularization parameter which controls the trade-off between smoothness and fidelity. This equation is equivalent to

$$\int_\Omega u(\boldsymbol{x}) s(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \lambda \int_\Omega \mathrm{Lu}(\boldsymbol{x}) \cdot \mathrm{Ls}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \frac{1}{M} \sum_{i=1}^{M} s(\boldsymbol{x}_i)$$

for all test functions $s \in V$ and $f_\epsilon = \frac{1}{M} \sum_{i=1}^{M} \delta_{\boldsymbol{x}_i}$, where $\delta_{\boldsymbol{x}_i}$ is the Dirac delta function centered on $\boldsymbol{x}_i$. See [3] for more detailed explanation.
Here, we to solve the problem in the second equation with Galerkin projection. Hence, we define a finite-dimensional function space $V_N \subset V$ as the span of the

basis functions in $\Phi = \{\phi_1, \ldots, \phi_N\}$ centered at grid points. And also set the test space to V_N.

Next, we solve this problem by employing sparse grids, namely the sparse-grid-based density estimation method.

The approximation $\hat{f}_N \in V_N$ of the density estimator $\hat{f}$ is a linear combination $\hat{f}_N = \sum_{i=1}^{N} \alpha_i \phi_i$ where the coefficients $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ are the solution of a system of linear equations $\boldsymbol{A\alpha} = \boldsymbol{b}$ stemming from the second equation. The system matrix $\boldsymbol{A}$ and the right hand side $\boldsymbol{b}$ depend on the choice of the discretization, i.e., the basis functions $\phi_1, \ldots, \phi_N$ and thus the space V_N.

Then, we apply a sparse grid discretization discussed in the previous section. Let Φ be the set of hierarchical basis functions of the (adaptive) sparse grid space $V_\ell^{(1)} \subset H_{\text{mix}}^2$ of level $\ell \in N$. According to the Galerkin approach, we need to find $\hat{f}_N \in V_\ell^{(1)}$ such that

$$\int_\Omega \hat{f}_N(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \lambda \int_\Omega \mathrm{L} \hat{f}_N(\boldsymbol{x}) \cdot \mathrm{L}\, \phi(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \frac{1}{M} \sum_{i=1}^{M} \phi\left(\boldsymbol{x}_i\right)$$

holds for all $\phi \in \Phi$. Since $\hat{f}_N$, the sparse grid function, is a linear combination of the basis functions in Φ with the coefficients $\boldsymbol{\alpha}$, we solve the previous equation by solving the system of linear equations

$$(\boldsymbol{R} + \lambda \boldsymbol{C})\boldsymbol{\alpha} = \boldsymbol{b}$$

with $R_{ij} = (\phi_i, \phi_j)_{L^2}$, $C_{ij} = (\mathrm{L}\phi_i, \mathrm{L}\phi_j)_{L^2}$ and $b_i = \frac{1}{M} \sum_{j=1}^{M} \phi_i\left(\boldsymbol{x}_j\right)$, where we used an arbitrary ordering of the N sparse grid basis functions $\phi_{\boldsymbol{l},\boldsymbol{i}}$ and coefficients $\alpha_{\boldsymbol{l},\boldsymbol{i}}$. Therefore, to obtain an estimated density function $\hat{f}_N$, we define the sparse grid level $\ell \in N$–or the number of refinement steps—and solve the linear equation system to get the coefficient vector $\boldsymbol{\alpha}$ of the linear combination corresponding to the sparse grid function $\hat{f}_N$. The level of the sparse grid and the number of refinement or coarsening steps are parameters that need to be chosen by the users.

4 Spatially Adaptivity of Sparse Grid

An adaptive sparse grid enable us to achieve better approximation accuracy while optimize the size of the sparse grid. The spatially adaptivity of sparse grid includes refinement and coarsening of the grid points. In the case of classification problem, each class has its own sparse grid. The algorithms that we have developed evaluate each grid point in each sparse grid whether at the current area the classification result is confident enough. If the density estimation for some classes at this area are similar, then this area is ambiguous and the grid point here should be refined to achieve higher classification confidence. On the contrary, if for a specific area, one class dominates significantly, then we are able to remove some of the grid points at this area without reduce the classification accuracy largely.

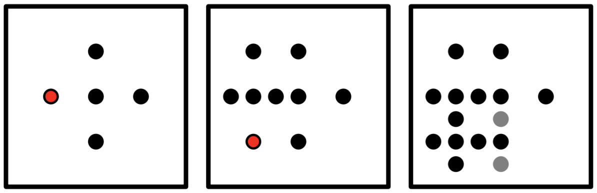

Figure 5: Starting with a grid of level 2 (left), we refine one grid point, creating all children in the hierarchical tree of basis functions (middle), and repeat this once more (right). Also, it is necessary that for each grid point all hierarchical ancestors exist. If they are missing, they have to be created recursively (gray grid points) [5].

First, we give a short introduction about the basic idea of sparse grid refinement. An adaptive refinement can select which grid points in a sparse grid structure should be refined next, due to local error estimation. Often, all 2d children in the hierarchical structure are added to the current grid, if they haven't been created yet, see Figure 5. Note that it usually has to be ensured that all missing parents have to be created. Alternatively, the hierarchical children of all possible refinement candidates, i.e. grid points for which not all children exist yet, could be considered to create only those which will contribute most to the problem's solution [5].
In the next subsection, we will present the idea behind our sparse grid coarsening algorithm.

4.1 Grid-Point-Based Coarsening Algorithm

The basic idea of coarsening is to remove those points which locate in area where the popularity of one class is significantly dominating the popularity of the other classes. To this end, we need to find out the grid points whose density function value of one class is significantly higher than the other ones. Besides, the neighbors of the grid points to be coarsened should also lie in the area where the classification confidence is high. What's more, in order to keep the structure of sparse grids, the points which are coarsened should not be the child points.

In this algorithm, we first find out the most dominant and the second most dominant class label and the corresponding density function values, and compute the difference between the popularity of these two classes. If the difference value is large, then the current point is significantly dominated by the class which has the maximum density value. In order to evaluate whether the current observed point lies in the border of two or more classes, we need to evaluate its neighbors as well. In the ideal case, the neighbors should also be dominated by the same class as the current observed grid point. Besides, the closer the

Algorithm 1 Grid point based coarsening functor

```
Data: Sparse grid points with evaluated surpluses
Result: Score for the current sparse grid point
$C := all\ classes$
$p := current\ grid\ point$
$f_i := density\ estimation\ value\ of\ class\ i\ at\ p$
$s_1 = \max(f_i), i \in C$
$l_1 = \underset{i}{\operatorname{argmax}}(f_i), i \in C$
$s_2 = \max(f_i), i \in C\ and\ i \neq l_1$
$l_2 = \underset{i}{\operatorname{argmax}}(f_i), i \in C\ and\ i \neq l_1$
$score = s_1 - s_2$
$N := All\ neighboring\ grid\ points$
$d = \sum_{n \in N} 1/euclidean\ distance(n, p)$
for $n \in N$ do
    $f'_i := density\ estimation\ of\ class\ i\ at\ n$
    $d' = euclidean\ distance(n, p)$
    $score_n = f'_{l_1} - f'_{l_2}$
    $score = score + score_n/(d \cdot d')$
end
return $-score$
```

Algorithm 2 Grid point coarsening for classification

```
Data: Sparse grid points with evaluated surpluses
Result: Score for the current sparse grid point
$C := all\ classes$
for $c \in C$ do
    $P := all\ grid\ points\ in\ c$
    for $p \in P$ do
        $score_p = Grid\ point\ based\ coarsening\ functor\ (p)$
    end
end
```

neighbors are, the more important they are in the evaluation. Hence, we compute the scores of the neighbors using similar measurement and the euclidean distance to compute the weight for each neighbor. Similarly, the larger the score of the neighbors is, the more likely it is that the point is in an area where the classification confidence is high, and thus the point could be coarsened. As the returned value, we use $-score$ instead of $score$, since the coarsening function will coarsen the points with the lowest score.

This algorithm is implemented in the SGpp library using C++. The basic idea of the pipeline can be described as the following:

After reading the data set, initialization, and the pre-computation part, the grid points which are not parents will be scored by the coarsening functor. After scoring, the function will choose the grid points that need to be removed according to the scores and the coarsening configuration which is written in the corresponding .json file. Two main configuration parameters for coarsening are $NumCoarsening$, the number of coarsening, and *threshold*, the threshold value. The function will choose $NumCoarsening$ candidates and compare whether it is lower than then threshold. If the condition is fulfilled, the candidates will then be removed. And all the surpluses of the current sparse grid will be updated.

5 Experiments

We test our algorithm performance on the 2-dimensional Ripley dataset. The well-known Ripley dataset problem consists of two classes where the data for each class have been generated by a mixture of two Gaussian distributions.

5.1 Test on the Ripley dataset

In Figures 6-8 we show the visualization of the results. Figure 6 demonstrate the result for coarsening one point. In Figure 7, the result for coarsening 2 points and the Figure 8 visualize coarsening 3 points.

In each graph, the background heatmaps represent the popularity of each class, class 0 on the left side and class 1 on the right side. The lighter the color is, the higher popularity the current class has. In the graphs, the black dots are the sparse grid points that has children and these parent points cannot be coarsened. The white dots are the child points in the sparse grid that could be considered for coarsening. The sizes of the white points represent the scores of the current point, and the larger the point is, the lower score it has and it will be more likely to be removed. The red cross on the white point means that the current grid point is removed according to the algorithm.

While coarsening one point, the function will coarsen the point that has the lowest score, and in this case, it is the white point close to (0.45,0.75). As we can see in Figure 6, the density of class 0 is significantly lower than the one of class 1, and so do its neighbors. Hence, we can say it is very likely that the points near this area will be labeled class 1. Therefore, we can remove the current sparse grid point and the accuracy should not be largely influenced.

The similar results can be observed while coarsening two and three points. In Figure 9, the score on validation set for each iteration is plotted. With initial level of 5, we achieve a score at around 0.76 without removing any points. As we increase the number of coarsened point, the score only slightly decreased. However, in order to coarsen more than three points in this case, some error with the Cholesky decomposition needs to be solved in the library.

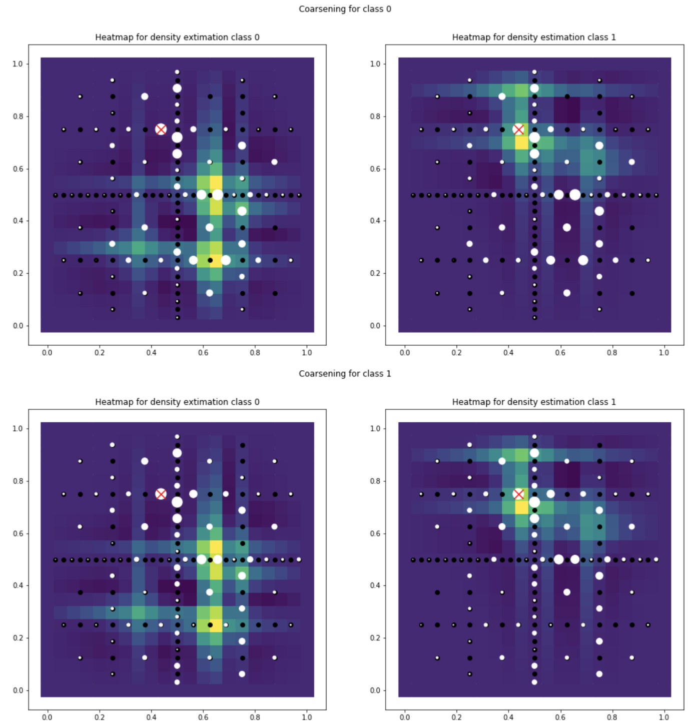

Figure 6: Coarsen one point for class 0. Left: heatmap for class 0, right: heatmap for class 1.

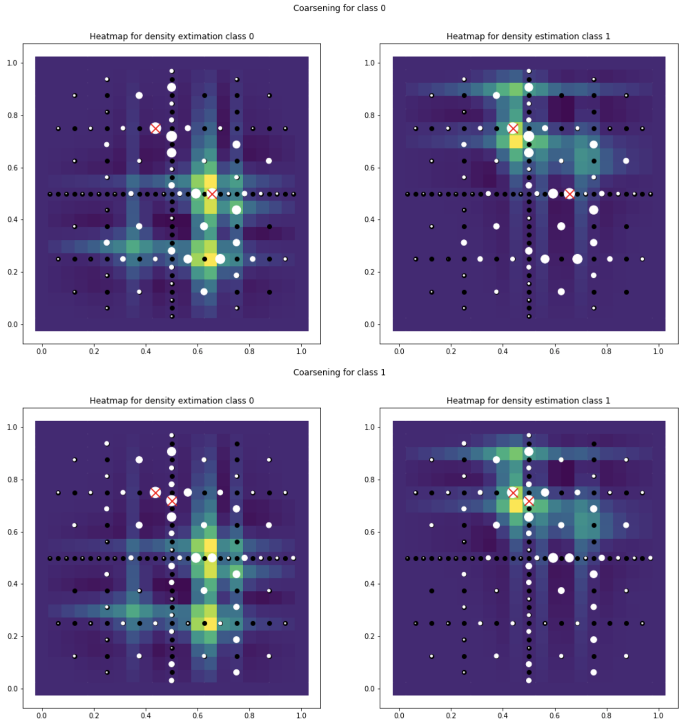

Figure 7: Coarsen two points for class 0. Left: heatmap for class 0, right: heatmap for class 1.

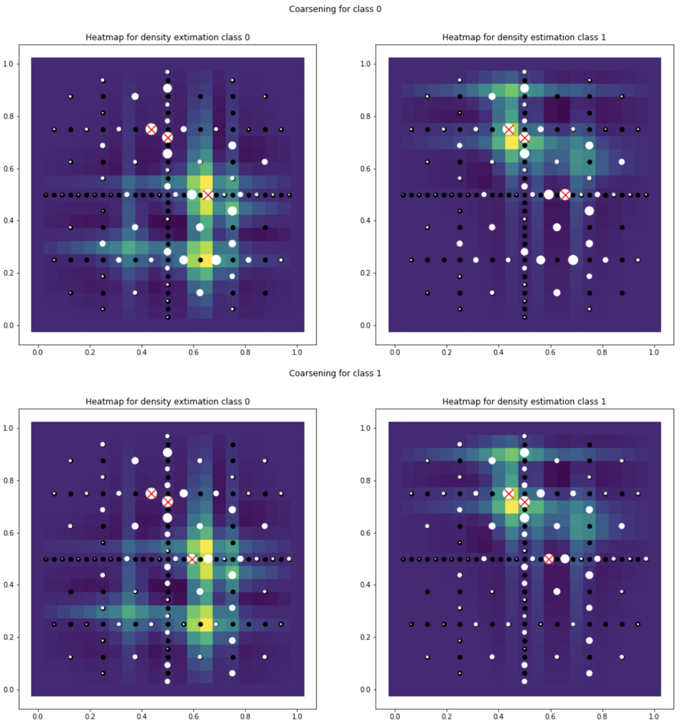

Figure 8: Coarsen three points for class 0. Left: heatmap for class 0, right: heatmap for class 1.

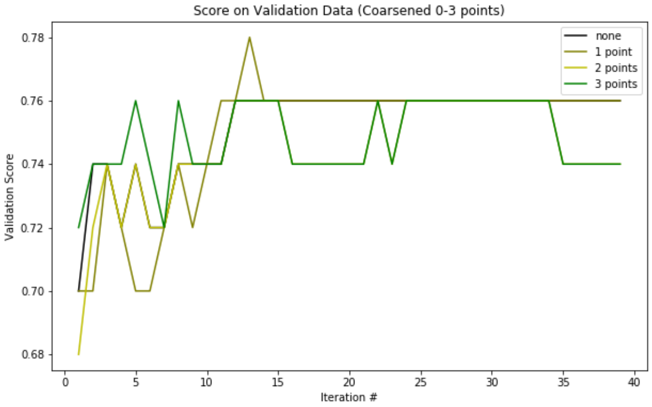

Figure 9: Score on Validation Data (Coarsened 0-3 points)

5.2 Complexity of the Algorithm

The algorithm in this work is grid point based, which means that the complexity of this algorithm does not depend on the size of the dataset. In Algorithm 2, it first loops over all the classes. In each class, it evaluates the coarsening score for all the grid points and in the evaluation functor, it loops over all the neighbors. In D dimension, the number of neighbors is $2 \cdot D$. We define the number of classes as C and the maximum number of grid points in each class is P, then the algorithm has order of $D \cdot C \cdot P$ time complexity.

6 Conclusion and Future work

In this work, we present a grid-point-based coarsening algorithm for sparse grid in classification problem. This algorithm optimizes the size of the sparse grids by removing the points that lies in domain which is less important in density estimation. The advantage of a grid-point-based coarsening functor is that the run time complexity of the algorithm depend on the size of the sparse grid and it will not be influenced by the number of the data points. So it can also be applied to large dataset. The threshold for coarsening score and the number of grid points that need to be removed can be predefined by the users. After introducing the algorithm, we tested it with a two-class and two-dimensional Ripley dataset. The coarsening results visualized in Chapter 5 show that the points that were removed by the algorithm consist with the choice when we observe it manually.

The grid point coarsening algorithm should still be tested on more dataset, for example, dataset with more than two classes and those in higher dimensions. Besides, the border of the sparse grid should also be considered while removing the grid points. Hence, this algorithm could still be improved in the future.

References

[1] HJ Bungartz and M. Griebel. *Sparse Grids.* Acta Numerica, 2004.

[2] J. Garcke. *Sparse Grids in a Nutshell.* Springer, 2012.

[3] G. Hooker M. Hegland and S. Roberts. Finite element thin plate splines in density estimation. 2000.

[4] B. Peherstorfer, D. Pflueger, and H. Bungartz. Density estimation with adaptive sparse grids for large data sets. 2014.

[5] D. Pflueger. Spatially adaptive sparse grids for high-dimensional problems. 2010.

[6] Dirk Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems.* Verlag Dr. Hut, München, August 2010.

[7] C. Zenger. Parallel algorithms for partial differential equations. 1991.