

# Data Mining with Spare Grids

Seminar: Computational Aspects of Machine Learning

Sebastian Kreisel

October 7, 2015

Technische Universität München



# Overview

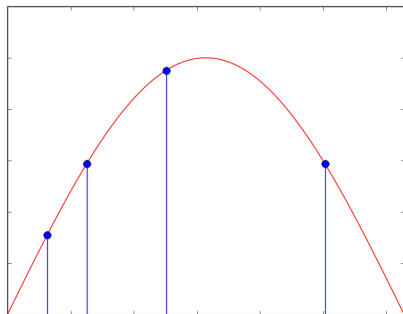
---

- Motivation for Sparse Grids
- Sparse Grids: Basics
- Sparse Grids: Machine Learning
- Examples with Data Sets
- Parallelization and Implementation

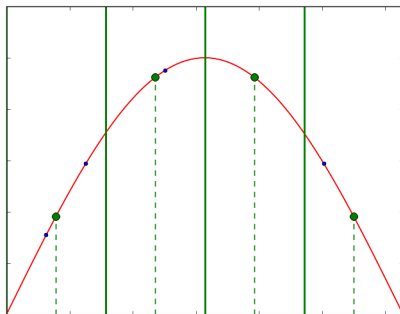
# Motivation for Sparse Grids

## Grid based approaches in ML

- Discretizes the space into a grid
- Basis-functions around grid points, not data points



Point based



Grid based

# Motivation for Sparse Grids

## Suitable for

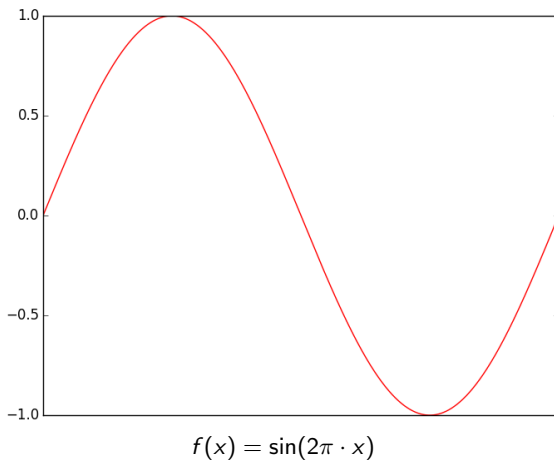
- Big datasets
- Easily/automatically classifiable data
- Medical, seismic, commercial data

## Curse of dimensionality

- The volume of a space is exponential in it's dimensions
- The amount of training data required becomes unmanageable
  - because of lacking computational/storage capacities
  - because data aquisition is expensive
- Becomes relevant for  $d > 3$
- **Applies to full-grid discretization**

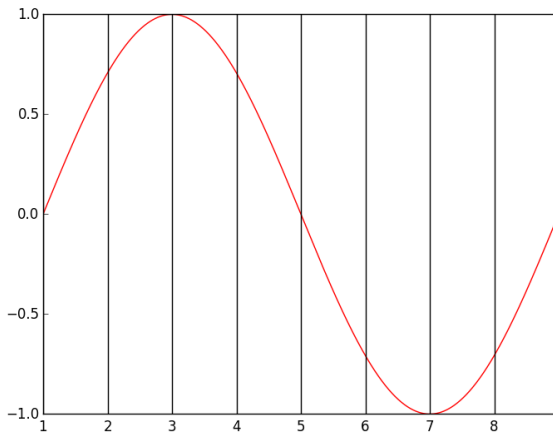
# Full Grid Discretization

## 1. Function to discretize



# Full Grid Discretization

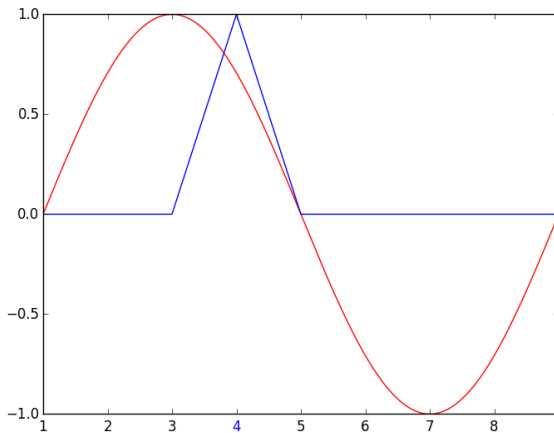
## 2. Full, regular grid



Eight centered gridpoints  $i \in \{1..8\}$

# Full Grid Discretization

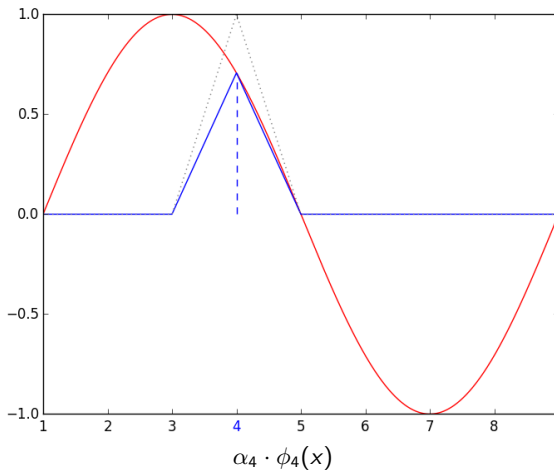
## 3. Basis function (standard hat function)



$\phi(x - 3)$  with  $\phi(x) = \max(\text{abs}(1 - x), 0)$

# Full Grid Discretization

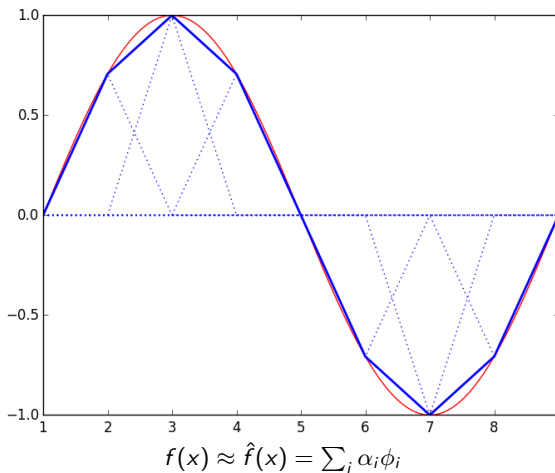
## 4. Coefficient $\alpha$ (Surplus)





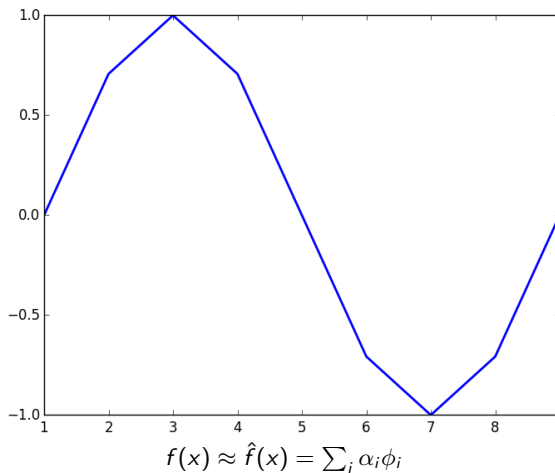
# Full Grid Discretization

Sum over all basis functions



# Full Grid Discretization

Sum over all basis functions



# Full Grid Discretization

## Full grid discretization in one dimension

1. A function  $f(x)$  to discretize
2. Gridpoints indexed by  $i \in \{1, 2, \dots\}$
3. Basis/Ansatz functions; i.e. hat function  $\phi(x) = \max(1 - |x|, 0)$
4. Coefficients  $\alpha_i$  (Surplusses)

$$f(x) \approx \hat{f}(x) = \sum_i \alpha_i \phi_i(x)$$

# Full Grid Discretization

## Level of detail

- Level  $l \in \{1, 2, 4, 8 \dots\}$  defining the number of gridpoints  $i$

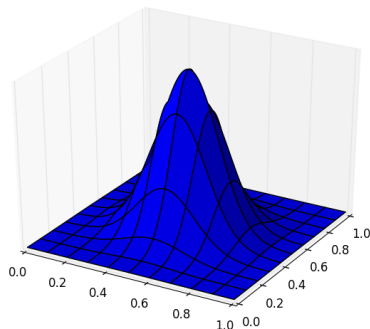
## $d > 1$ dimensions

- For each dimension  $\hat{f} = \sum \dots$
- Tensor product over all dimensions  $\prod \hat{f}(x)$

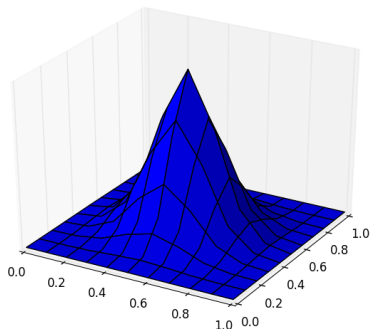
$$\underbrace{\sum_i \alpha_i \phi_i(\mathbf{x}_1)}_{d=1} \cdot \underbrace{\sum_j \alpha_j \phi_j(\mathbf{x}_2)}_{d=2} \cdot \dots$$

# Full Grid Discretization

Full grid with  $d = 2$



$$f(x) = \mathcal{N}(\vec{x})$$



$$\text{Full grid discretization } \hat{f}_1(x_1) \cdot \hat{f}_2(x_2)$$

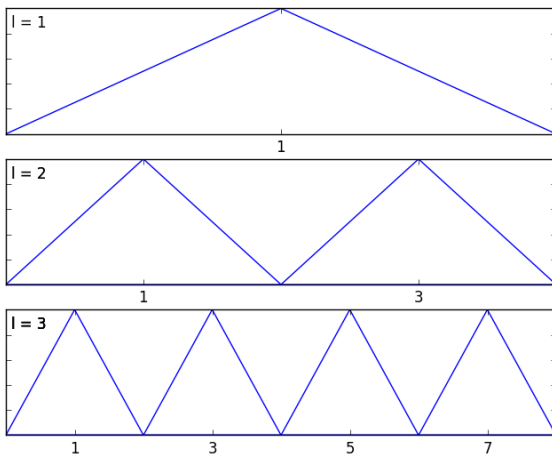
# Full Grid Discretization

## Summary

- Gridpoints  $i \in \{1..2^l\}$  defining  $\phi_i(x)$
- For *each* dimension: **sum over basisfunctions**  $\hat{f}(x) = \sum_i \alpha_i \phi_i(x)$
- For *all* dimensions: **product over dimensions**  $\prod \hat{f}(x)$

# Sparse Grids – Basics

## Hirachial Basis



# Sparse Grids – Basics

## Hirachial basis (vs nodal basis)

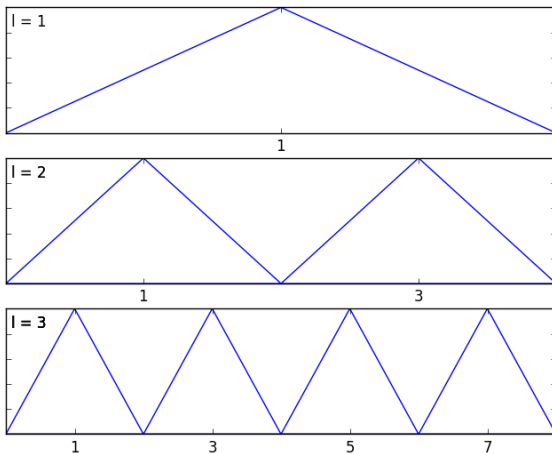
- Grouping gridpoints into levels  $l \in \{1, 2, 3 \dots\}$
- Basis function by index **and** level:  $\phi_{l,i}(x)$

$$\hat{f}(x) = \sum_{l,i} \alpha_{l,i} \cdot \phi_{l,i}(x)$$



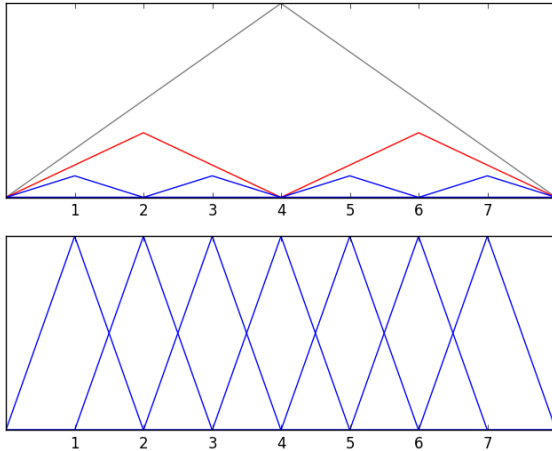
# Sparse Grids – Basics

## Hirachial Basis



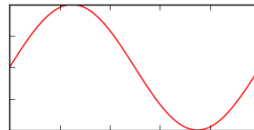
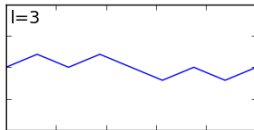
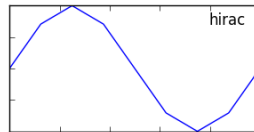
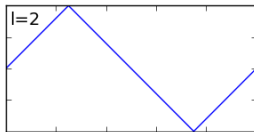
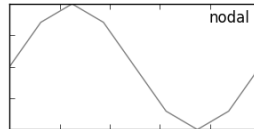
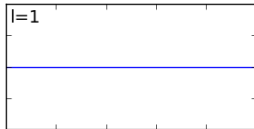
# Sparse Grids – Basics

## Hirachial vs. nodal basis



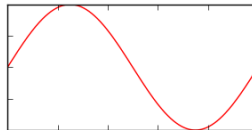
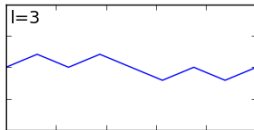
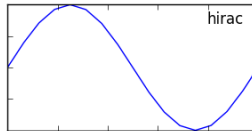
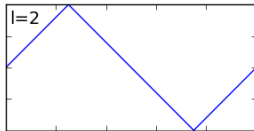
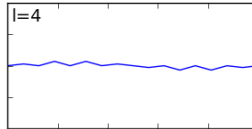
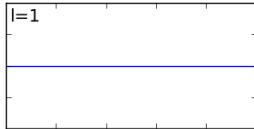
# Sparse Grids – Basics

## Full grid discretization: Hirachial basis



# Sparse Grids – Basics

## Full grid discretization: Hirachial basis



# Sparse Grids – Basics

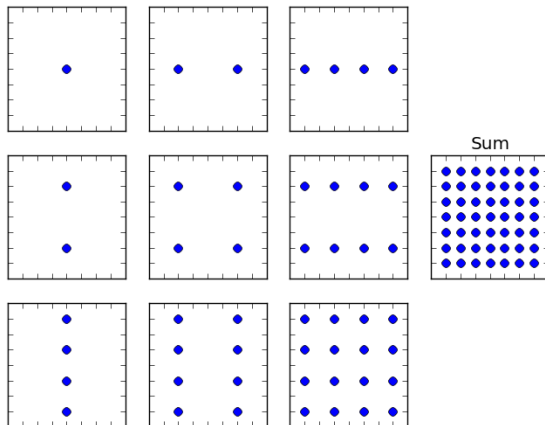
---

## Basis function subspaces

- Combination of levels and dimensions
- Notion of hierarchical subspaces
- Defined by the levels of detail in all dimensions  $(l_x, l_y, \dots)$

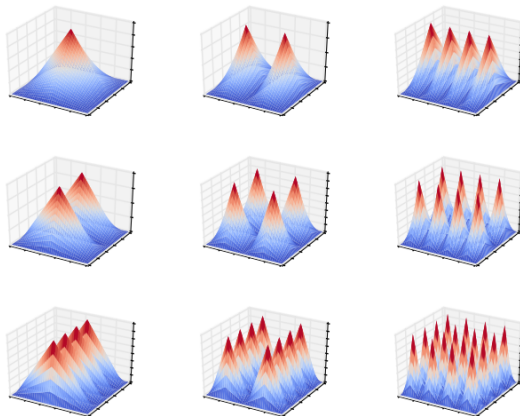
# Sparse Grids – Basics

## Hierarchical gridpoints



# Sparse Grids – Basics

## Hierarchical subspaces



# Sparse Grids – Basics

---

## Sparse grid – Changes

- Throwing away certain subspaces
- Finding those is a *a-priori* solvable optimization problem
- The resulting grid is now **sparse**

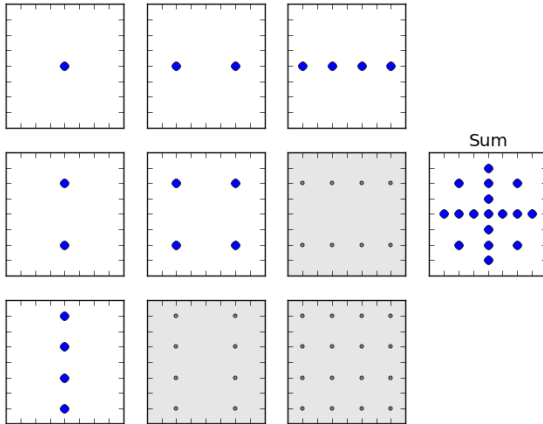
## Profit

- Reducing the computational effort “a lot”
- Maintaining “high” accuracy



# Sparse Grids – Basics

## A *sparse* grid



# Sparse Grids – Basics

---

## Boundry and smoothness

- Boundries need special treatment
- The function needs to be sufficient smooth  
 $D^2f$  needs to be bounded

## Adaptivity

- *A-posteriori* modifications to better fit the function
- Picking a single gridpoint and adding level of detail around it
- Prone to overfitting and huge computational effort

# Sparse Grids – Basics

---

## Summary

- Hierachial basis through grouping gridpoints into levels
- Creating “subspaces” through combination of levels in dimensions
- Selecting and combining subspaces

## To keep in mind

- Smoothness requirement for  $f(x)$
- Boundry treatment
- Accuracy–effort trade-off
- Adaptivity options (*a-posteriori*)

## Machine learning tasks

- Classification
- Regression

## Least squares

$$\hat{c} = \arg \min_f \left( \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2 + \lambda \|\nabla f\| \right)$$

## Sparse grid setting

- Do least squares in a sparse grid setting (“space”)
- Discretize  $\hat{c}$  using a sparse grid

## Least squares: Sparse grid discretized

$$\hat{c} = \arg \min_{\alpha} \left( \frac{1}{N} \sum_{i=1}^N (y_i - \sum_j \alpha_j \phi_j(x_i))^2 + \lambda \sum_j \alpha_j^2 \right)$$

## Matrix formulation

$$\left(\frac{1}{N}BB^T + \lambda C\right)\alpha = \frac{1}{N}By$$

$$\mathbb{R}^{M \times N} \ni B = \begin{bmatrix} \phi_1(x^{(1)}) & \dots & \phi_1(x^{(N)}) \\ \vdots & \ddots & \vdots \\ \phi_M(x^{(1)}) & \dots & \phi_M(x^{(N)}) \end{bmatrix} \quad C = \lambda I$$

## Observations

- $BB^T \in \mathbb{R}^{M \times M}$  where  $M$  = number of gridpoints
- Scales only linear in  $N$  = number of datapoints