Sparse Grids

SuYuzhang

Report in November,2022

Main job: Investigating sparse grid methods.

1 Intepolation

The sparse grid method is a method used for interpolation or integration. It selects a subset of the full tensor product collocation points to construct the interpolation. It is an approximation of the full tensor product method, which is a linear combination of tensor product formulas. It sacrifices the accuracy of logarithmic loss and obtains a large computational reduction, which effectively alleviates the curse of dimensionality in the case of high dimensions.

1.1 1D Interpolation

Let $f : [0, 1] \to \mathbb{R}$ be a function in 1D, We can approximate it with the following interpolation formula :

$$U^k(f) = \sum_{j=1}^{m_k} f(\xi_j^k) l_j^k(\xi) \ , k \geq 1 \ , U^0 = 0$$

U is the interpolation operator with the set of support nodes $\mathcal{X}^k = \{\xi_j^k \mid \xi_j^k \in [0, 1], j = 1, 2..., m_k\}$ and interpolation basis functions $l^k = \{l_j^k \mid l_j^k \in \mathcal{C}[0, 1], j = 1, 2..., m_k\}$, the interpolation basis functions satisfy $l_j^k(\xi_i^k) = \delta_{ij}$. Here k and m_k refer to the depth of interpolation and the total number of support nodes at depth k, respectively. The chioce of support nodes and interpolation basis functions can be as follows[1],

(1) Equidistant nodes with piecewise linear basis functions

$$m_k = \begin{cases} 1 & if \ k = 1 \\ 2^{k-1} + 1 & if \ k > 1 \end{cases}$$

$$\xi_j^k = \begin{cases} 0.5 & for \ j = 1 \ if \ m_k = 1 \\ \frac{j-1}{m_k - 1} & for \ j = 1, ..., m_k \ if \ m_k > 1 \end{cases}$$

$$l_j^k(\xi) = 1, k = 1$$

$$l_j^k(\xi) = \begin{cases} 1 - (m_k - 1) \mid \xi - \xi_j^k \mid, & if \mid \xi - \xi_j^k \mid < \frac{1}{m_k - 1} \\ 0, & otherwise, \end{cases}$$

(2) Chebyshev Gauss–Lobatto nodes (CGL) with Lagrange polynomial basis functions

$$m_k = \begin{cases} 1 & if\ k=1 \\ 2^{k-1}+1 & if\ k>1 \end{cases}$$

$$\xi_j^k = \begin{cases} 0.5 & for\ j=1\ if\ m_k=1 \\ (1-\cos(\frac{(j-1)\pi}{m_k-1}))/2 & for\ j=1,...,m_k\ if\ m_k>1 \end{cases}$$

$$l_j^k(\xi) = \begin{cases} 1, & k=1 \\ \displaystyle\prod_{i=1,i\neq j}^{m_k} \frac{\xi-\xi_i^k}{\xi_j^k-\xi_i^k}, & k>1, j=1,...,m_k \end{cases}$$

It is easy to see that the points are nested, that is $\mathcal{X}^k \subset \mathcal{X}^{k+1}$

1.2 Hierarchical interpolation

Due to the property of nested points the interpolation can be written as a hierarchical form[2]. We first define the incremental interpolant $\Delta^k = U^k - U^{k-1},\ \forall k \geq 1$. Thus,

$$\Delta^k(f) = U^k(f) - U^{k-1}(f)$$

and we have $U^{k-1}(f) = U^k(U^{k-1}(f))$. Using this, we obtain

$$\Delta^k(f) = \sum_{\xi_j^k \in \mathcal{X}^k} f(\xi_j^k) l_j^k - \sum_{\xi_j^k \in \mathcal{X}^k} U^{k-1}(f)(\xi_j^k) l_j^k$$

$$= \sum_{\xi_j^k \in \mathcal{X}^k} (f(\xi_j^k) - U^{k-1}(f)(\xi_j^k)) l_j^k$$

due to $f(\xi_j^k) - U^{k-1}(f)(\xi_j^k) = 0\ \forall\ \xi_j^k \in \mathcal{X}^{k-1}$, we have

$$\Delta^k(f) = \sum_{\xi_j^k \in \mathcal{X}_\Delta^k} (f(\xi_j^k) - U^{k-1}(f)(\xi_j^k)) l_j^k$$

where $\mathcal{X}_\Delta^k$ denotes the nodes added by interpolation from depth $k-1$ to depth k due to the property of nested nodes, we can easy to see it have $m_k^\Delta = m_k - m_{k-1}$ nodes. Thus we can rewrite above formula as,

$$\Delta^k(f) = \sum_{j=1}^{m_k^\Delta} \underbrace{(f(\xi_j^k) - U^{k-1}(f)(\xi_j^k))}_{w_j^k} l_j^k$$

We define w_j^k as the 1D hierarchical surpluses, which is the difference between the actual function value and the value obtained using the interpolant at the $\mathcal{X}_\Delta^k$. The l_j^k we call it hierarchical basis functions. By shifting the terms we can get,

$$U^k(f) = U^{k-1}(f) + \Delta^k(f) = \sum_{i=1}^{k} \Delta^i(f)$$

the set of support nodes can be rewritten as $\mathcal{X}^k = \bigcup_{i=1}^k \mathcal{X}_\Delta^i$.

1.3 Multi-variate interpolation

The multi-variate interpolation formula could simply use tensor product of univariate interpolation formula to construct, given as

$$U^{k_1} \otimes \cdots \otimes U^{k_d}(f) = \sum_{\xi^{k_1}_{j_1} \in \mathcal{X}^{k_1}} \cdots \sum_{\xi^{k_d}_{j_d} \in \mathcal{X}^{k_d}} f(\xi^{k_1}_{j_1}, \ldots, \xi^{k_d}_{j_d}) \cdot (l^{k_1}_{j_1} \otimes \cdots \otimes l^{k_d}_{j_d})$$

where $\mathbf{k} = [k_1, \ldots, k_d]$ represents the depth of interpolation used in each dimension. The hierarchical form can be rewritten as,

$$U^{k_1} \otimes \cdots \otimes U^{k_d}(f) = \sum_{i_1=1}^{k_1} \cdots \sum_{i_d=1}^{k_d} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d})(f)$$

2 Sparse Grids

Sparse grid interpolation is a linear combination of tensor products of one-dimensional interpolation. In the high-dimensional case, it obtains a large reduction in computational effort by sacrificing some interpolation accuracy. It is an approximation of the full tensor product method and also known as *Smolyak algorithm*[3], the algorithm give the formula as,

$$A(q,d)(f) = \sum_{|\mathbf{k}| \le d+q} (\Delta^{k_1} \otimes \cdots \otimes \Delta^{k_d})(f) = A(q-1,d)(f) + \Delta A(q,d)(f)$$

with $A(-1,d) = 0$, and $|\mathbf{k}| = k_1 + \ldots k_d$. The d-dimensional incremental sparse interpolant $\Delta A(q,d)(f)$, can be written as,

$$\begin{aligned} \Delta A(q,d)(f) &= \sum_{|\mathbf{k}| = d+q} (\Delta^{k_1} \otimes \cdots \otimes \Delta^{k_d})(f) \\ &= \sum_{|\mathbf{k}| = d+q} \sum_{\mathbf{j}} \underbrace{(l^{k_1}_{j_1} \otimes \cdots \otimes l^{k_d}_{j_d})}_{l^{\mathbf{k}}_{\mathbf{j}}} \cdot \underbrace{(f(\xi^{k_1}_{j_1}, \ldots, \xi^{k_d}_{j_d}) - A(q-1,d)(f)(\xi^{k_1}_{j_1}, \ldots, \xi^{k_d}_{j_d}))}_{w^{\mathbf{k}}_{\mathbf{j}}} \end{aligned}$$

where $\mathbf{j} = (j_1, \ldots, j_d)$ denotes the multi-index. As for the 1D case, the coefficients $w^{\mathbf{k}}_{\mathbf{j}}$ are defined as *hierarchical surpluses*. Most of the time people use its explicit form,

$$A(q,d) = \sum_{q-d+1 \le |\mathbf{k}| \le q} (-1)^{q-|\mathbf{k}|} \binom{d-1}{q-|\mathbf{k}|} \cdot (U^{k_1} \otimes \cdots \otimes U^{k_d})(f)$$

References

[1] N. Agarwal and N. R. Aluru, “A domain adaptive stochastic collocation approach for analysis of mems under uncertainties,” *Journal of Computational Physics*, vol. 228, no. 20, pp. 7662–7688, 2009.

[2] A. Klimke, “Piecewise multilinear sparse grid interpolation in matlab,” 2003.

[3] S. A. Smolyak, “Quadrature and interpolation formulas for tensor products of certain classes of functions,” in *Doklady Akademii Nauk*, vol. 148, no. 5. Russian Academy of Sciences, 1963, pp. 1042–1045.