High Dimensional Polynomial Interpolation on Sparse Grids: Smolyak's Algorithm

Dirk Krueger

University of Pennsylvania

and NBER

Felix Kubler

Stanford University

Victor's Quantitative Macroeconomics Class
University of Pennsylvania, November 25, 2003

The Problem

- Approximate the function $f : [-1, 1]^d \rightarrow \Re$ by interpolating it at points $(x^i, y^i)_{i \in I}$ where $x^i \in \mathcal{H} \subset [-1, 1]^d$ and $y^i \in f(x^i)$
- Need to choose grid $\mathcal{H}$ and interpolating function $\hat{f}$

Example: OLG model with Aggregate Uncertainty and m Generations

- State space $x = (z, k_1, k_2, \ldots k_J) \in \mathcal{H}$. Euler equations:

$$\begin{aligned} u'(c_j(x)) &= \beta \sum_{z'|z} (1 + r(z', K'))u'(c_{j+1}(x')) \\ \text{where } K' &= \sum_{s} k'_j(x) \text{ and } x' = (z', k'_1(x), \ldots , k'_J(x)) \end{aligned}$$

which together with the budget constraints

$$c_j(x) + k'_j(x) = y_j + (1 + r(x))k_{j-1}$$

determine the optimal policies $(c_j(x), k'_j(x))_{j=1}^{J}$.

Example: Remarks

- State space: if bounds on $k_j \in [\underline{k}_j, \bar{k}_j]$, straightforward to map into $[-1, 1]$.
- In order to solve for c_j, k'_j on $(x_i)_{i \in I}$ need to know c_{j+1} on the entire state space. Thus need to interpolate c_{j+1}.

Back to General Problem

- Simplest choices
  - $\mathcal{H} = \{a_1, \ldots , a_n\}^d$. Note that $\#(\mathcal{H}) = n^d$ increases exponentially in d (curse of dimensionality).
  - $\hat{f} =$sum of tensor products of one dimensional polynomials

    $$\begin{aligned} \mathcal{P}^m &= \{p : p(x) = x^q, q = 0, 1, 2, \ldots m, \text{ for } x \in [-1, 1]\} \\ \mathcal{T} &= \{\tau : \tau(x) = x_1^{i_1} \cdot x_2^{i_2} \cdot \ldots \cdot x_d^{i_d}, \text{ for } x^{i_j} \in \mathcal{P}^m\} \end{aligned}$$

    Again, $\#(\mathcal{T}) = (m + 1)^d$ grows exponentially in d

- Key: make grid and set of interpolating functions smaller

Reducing the Set of Interpolationg Functions

- Set of complete polynomials of degree k in d dimensions. Define

$$|\mathbf{i}| = i_1 + i_2 + \ldots + i_d$$

- Set of complete polynomials is given by

$$\mathcal{C}^k = \{c : c(x) = x_1^{i_1} \cdot \ldots \cdot x_d^{i_d} \text{ for } |\mathbf{i}| \leq k\}$$

Reducing the Grid Size

1. Determine grid size. Define

   $$\begin{aligned} m_1 &= 1 \\ m_i &= 2^{i-1}+1 \text{ for } i = 2, 3, 4, \ldots \end{aligned}$$

   Choice of i determines size of grid. For example, $m_2 = 3, m_3 = 5, m_4 = 9, \ldots$ Note: for a given choice of i, m_i determines the maximal number of grid points along each dimension.

2. Choose the grid points along a single dimension

- Chebychev extrema: with a total of m_i points, choose

$$\xi^i_j = -\cos\left(\frac{\pi(j-1)}{m_i-1}\right) \text{ for } j = 1, \ldots, m_i$$

- Collect in set $\mathcal{G}^i = \{\xi^i_1, \ldots, \xi^i_{m_i}\}$ and define $\mathcal{G}^1 = \{0\}$.

- Example:

$$\begin{aligned} i &= 1, m_i = 1, \mathcal{G}^i = \{0\} \\ i &= 2, m_i = 3, \mathcal{G}^i = \{-1, 0, 1\} \\ i &= 3, m_i = 5, \mathcal{G}^i = \{-1, -\cos\left(\frac{\pi}{4}\right), 0, -\cos\left(\frac{3\pi}{4}\right), 1\} \end{aligned}$$

- Note: $\mathcal{G}^i \subset \mathcal{G}^{i+1}$, since the sets consist of Chebychev extrema.

3. Choose the multi-dimensional grid. Define the "size" of the grid as $q > d$, for a given dimension d of the problem. Now define the grid $\mathcal{H}$ as

$$\mathcal{H}(q, d) = \bigcup_{q-d+1 \leq |\mathbf{i}| \leq q} \left(\mathcal{G}^{i_1} \times \mathcal{G}^{i_2} \times \ldots \times \mathcal{G}^{i_d} \right)$$

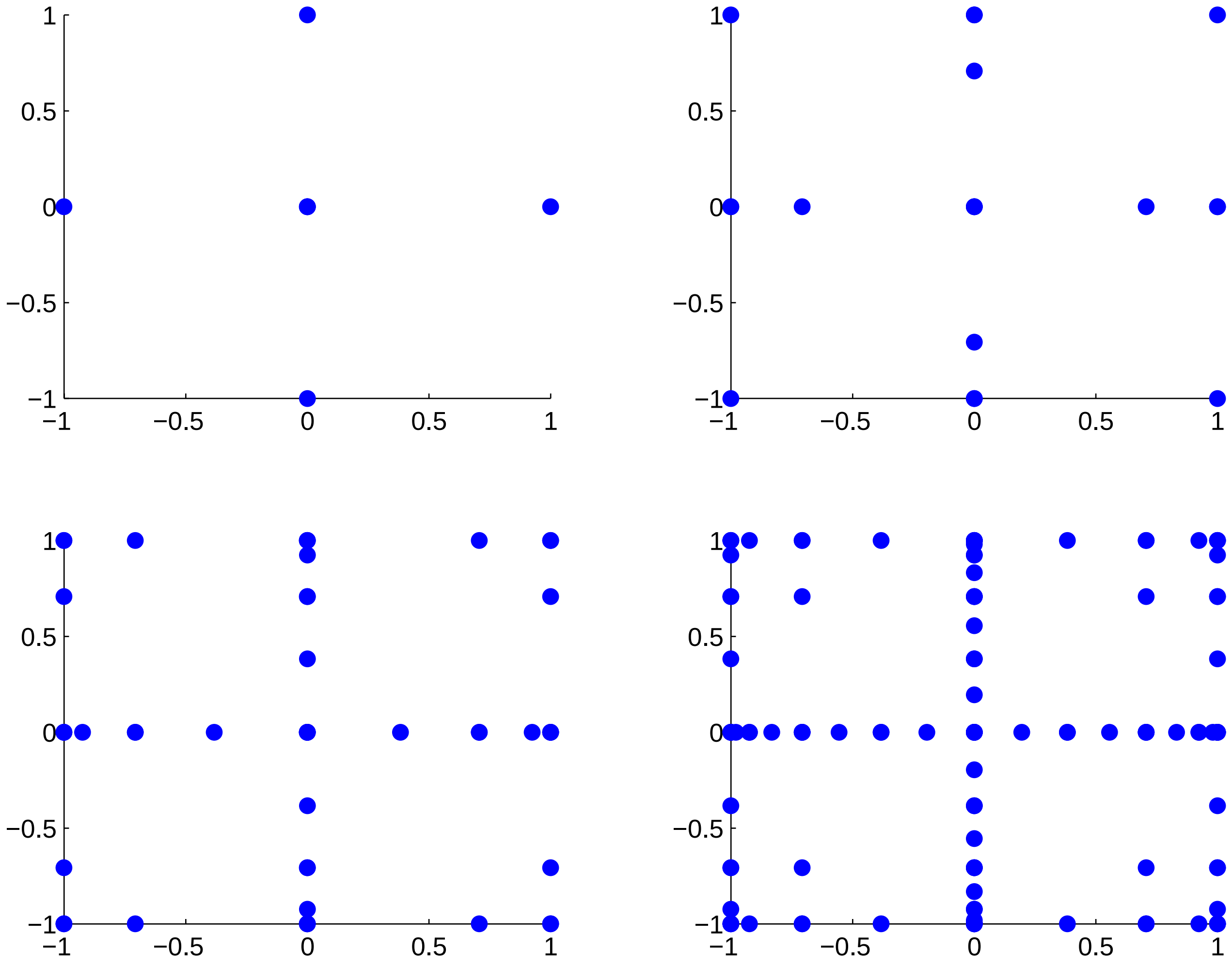
   - Note: since $i_j \geq 1$, we have $\max_j\{i_j\} = q - d$. Thus the maximal number of points along one dimension is $m_{i_j} = 2^{q-d} + 1$.

   - Remember that it needs a polynomial of degree 2^{q-d} to interpolate $2^{q-d} + 1$ points along one dimension. Thus, once q is chosen, for given dimension d this determines the grid $\mathcal{H}(q, d)$ as well as the degree of complete polynomials $k = 2^{q-d}$ that is used.

Example

- Let $d = 2$ and $q = 3, 4$

- $$\begin{aligned} \mathcal{H}(3,2) &= \mathcal{G}^1 \times \mathcal{G}^2 \cup \mathcal{G}^2 \times \mathcal{G}^1 \\ &= \{(0,-1),(0,0),(0,1)\} \cup \{(-1,0),(0,0),(1,0)\} \\ \mathcal{H}(4,2) &= \mathcal{G}^1 \times \mathcal{G}^3 \cup \mathcal{G}^2 \times \mathcal{G}^2 \cup \mathcal{G}^3 \times \mathcal{G}^1 \end{aligned}$$


1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1

- Number of grid points

Table 1: Size of the Grid

d	q	$2^{q-d}+1$	$\#\mathcal{H}(q,d)$	$\left[2^{q-d}+1\right]^d$
2	3	3	5	9
2	4	5	13	25
2	5	9	29	81
10	13	9	1581	9^{10}
d	$q = d+1, 2, 3$	$2^{q-d}+1$	$1+2d$ $1+4d+2d(d-1)$ $\ldots+\frac{8}{6}d(d-1)(d-2)$	$\left[2^{q-d}+1\right]^d$

Interpolation

- Define

$$p^i(x) = p^{i_1}(x_1) \cdot \ldots \cdot p^{i_d}(x_d) \text{ for } i = (i_1, \ldots, i_d) \text{ and } x \in [-1, 1]^d$$

- Interpolate f on $\mathcal{H}(q, d)$ by weighted sum of complete polynomials

$$\hat{f}^{q,d}(x) = \sum_{q-d+1 \leq |\mathbf{i}| \leq q} (-1)^{q-|\mathbf{i}|} \begin{pmatrix} d-1 \\ q-|\mathbf{i}| \end{pmatrix} p^i(x)$$

- Main advantages: $\mathcal{H}(q, d)$ relatively small and only few polynomials (only complete polynomials of degree $2^n = 2^{q-d}$) to evaluate.

Which One-Dimensional Polynomials to Use?

- Chebychev polynomials

$$T_j(x) = \cos\Big(j \cos^{-1}(x)\Big) \text{ for } x \in [-1,1]$$

- One dimensional interpolation

$$\begin{aligned}
p^{i_j}(x_j) &= \sum_{k=0}^{m_i-1} \xi_k(f) T_k(x_j) \text{ for } x_j \in [-1,1] \\
\text{where } \xi_k(f) &= \frac{2}{d_k m_i} \sum_{j=1}^{m_i} T_k(\xi_j) f(\xi_j) \\
\text{and } d_0 &= d_{m_i} = 2 \text{ and } d_j = 1 \text{ for } j = 1, \ldots, m_i - 1
\end{aligned}$$

Properties of the Method

- Method is (almost) optimal within the set of polynomial approximations (Barthelmann, Novak and Ritter, 1999);
  http://www.minet.uni-jena.de/~novak/schrift.html
- Method is universal, that is, almost optimal for many different function spaces.
- Works well in stochastic OLG models with up to 15 generations (but problem has no kinks);
  http://www.econ.upenn.edu/~dkrueger/olgcomp.pdf

Potential Applications

- OLG models with aggregate uncertainty and more than 2 generations
- International RBC models with more than two countries
- Can be used in problems that need to approximate one dimension very accurately, but other dimensions very sparsely.