Bayesian Estimation with Sparse Grids

Kenneth L. Judd and Thomas M. Mertens

Institute on Computational Economics
August 07, 2007
Outline

1. Introduction

2. Sparse grids
   Construction
   Integration with sparse grids

3. Bayesian Estimation

4. Numerical results

5. Conclusion/Future Research
Introduction

- Bayesian estimation involves integration of the posterior.
- We present a numerical quadrature approach for Bayesian estimation.
- We use sparse grids to deal with high dimensionality.
- It is an alternative to the use of simulations.
Bayesian Estimation

- Get data $Y$.
- Obtain likelihood function $\mathcal{L}(Y|\theta)$.
- Use prior information $p(\theta)$.
- Posterior: $p(\theta|Y) \sim \mathcal{L}(Y|\theta) \cdot p(\theta)$
- We then compute Bayesian estimators:

$$M = \int h(\theta) \mathcal{L}(Y|\theta)p(\theta)d\theta$$
Introduction

- Main difficulty is dimensionality of the problem.
- Monte-Carlo methods converge at a rate independent of dimension — but slowly.
- We want a faster method.
- There is hope: the function is very smooth.
Introduction

• Pseudo-random schemes give you convergence \( O(N^{-\frac{1}{2}}) \).
• Equidistributional sequences give convergence of order 1 for \( C^1 \)-functions.
• There is convergence of \( O(N^{-k}) \) for periodic \( C^k \)-functions.
• For very smooth functions, there is essentially no "curse of dimensionality".

Related literature

- MCMC: Peter Rossi, “Bayesian Statistics and Marketing”
- Sparse grids have been used in economics and finance.
- Examples in economics are: Kuebler and Krueger, and Winschel.
Approximation

- Goal: Approximate function $f(x)$ with basis functions.
- We can write the approximation $\hat{f}(x)$ as:

  $$\hat{f}(x) = \sum_{(l,i)} u(l,i)\phi(l,i)(x) \approx f(x)$$

- Basis functions could for instance be:
Approximation
Approximation
Approximation
• Why can we truncate the sum at level $l$?
• Look at the approximation again.
• The absolute value of coefficients shrinks to zero.
• More precisely: $u_{(l,i)} \leq c \cdot 2^{-2 \cdot ||l||_1}$. 

Exponents
Sparse grids — Construction

- The goal is to generalize the one-dimensional approximation.
- This requires to specify:
  - the basis function
  - the grid
Sparse grids — Construction

- Choice of basis functions and associated grid.

\[ \phi(l,i) = \prod_j \phi(l_j,i_j) \]
Sparse grids — Construction

• "Curse of Dimensionality" for full grids — grid size grows at $n^d$. 
Sparse grids — Construction

Sparse grids:

\[ n=1 \quad n=2 \quad n=3 \]
Approximation

- Note: We can now still write the approximation as

\[ \hat{f}(\mathbf{x}) = \sum_{(l,i)} u_{(l,i)} \phi_{(l,i)}(\mathbf{x}) \approx f(\mathbf{x}) \]

- where \( l = (l_1, \ldots, l_d) \), \( i = (i_1, \ldots, i_d) \)
- \( \phi_{(l,i)} = \prod_j \phi_{(l_j,i_j)} \)
- It is a generalization of the 1-d concept.
- But: sparse grids come at a cost: you have to have bounded second mixed derivatives.
### Features

<table>
<thead>
<tr>
<th># grid points</th>
<th>Full grids ( n^d )</th>
<th>Sparse grids ( O(n \log(n)^{d-1}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( L_2 )-error</td>
<td>( O(N^{-2}) )</td>
<td>( O(N^{-2} \cdot \log(N)^{d-1}) )</td>
</tr>
<tr>
<td>( L_\infty )-error</td>
<td>( O(N^{-2}) )</td>
<td>( O(N^{-2} \cdot \log(N)^{d-1}) )</td>
</tr>
</tbody>
</table>
Why is that such a big thing?

- Just imagine: 32 grid points per dimension, 30 dimensions.
- Sparse grids: 4,518,180 grid points.
- That fits into a computer’s RAM.
- 142724769270595988105828596944950000000000000000000000 points in full grids don’t!
Having the approximation, we can now integrate

\[ \int \sum_{(l,i)} u_{(l,i)} \phi_{(l,i)}(x) = \sum_{(l,i)} u_{(l,i)} \int \phi_{(l,i)}(x) \]

How about computing moments?

Use the same grid: since \( \|f - \hat{f}\| < \delta \)

\[ \| \int x^2 (f(x) - \hat{f}(x))dx \| < \delta \int x^2 dx \]
Bayesian Estimation

- Let’s look at the problem again.
- Compute expected value of functions of $\theta$

$$M = \int h(\theta) \mathcal{L}(Y|\theta)p(\theta)d\theta$$

- Problem: cannot sample from $\mathcal{L}(Y|\theta)p(\theta)$ directly.
- Use MCMC methods: Gibbs or Metropolis-Hastings.
Technique: MCMC

- Create Markov-chain that has true distribution as stationary distribution.
- First: Gibbs sampling.
- Pick starting point.
- Pick successively draw from each marginal distribution to get to the next step.
Technique: MCMC

• Now: Metropolis-Hastings.
• Pick starting point.
• Draw sample $\psi$ from proposal distribution $Q(\theta'_t | \theta_{t-1})$ (i.e. $Q(\theta'_t | \theta_{t-1}) \sim N(\theta_{t-1}, \Sigma)$).
• Accept draw only if $u < \frac{p(\psi)\mathcal{L}(Y|\psi)}{p(\theta_{t-1})\mathcal{L}(Y|\theta_{t-1})}$ where $u \sim U([0, 1])$ otherwise $\theta_t = \theta_{t-1}$. 

• Both Gibbs and Metropolis-Hastings samplers have a burn-in phase.
• These are the samples you need to get to the stationary distribution.
• These draws are not used to compute the integral.
Technique: MCMC

- There are further complications with MCMC
- doesn’t use all information about the density at a draw.
- takes long to converge.
- error estimation is comparably hard.
- no possibility to check for identification.
Our approach

- Take “burn-in” samples.
- Find peak of the posterior using samples as starting points.
- Compute Hessian at the peak.
- Take out Gaussian function.
- Repeat until only little mass is left.
- Treat rest with sparse grids.
In pictures

- Treat remainder with sparse grids.
Advantages

• We then have the approximation:

\[ p(\theta|Y) = \sum G(\theta) + \sum_{(l,i)} u_{(l,i)} \phi_{(l,i)}(\theta) \]

• We can then integrate since we know

\[ \sum \int h(\theta) G(\theta) \]

and

\[ \int \sum_{(l,i)} h(\theta) u_{(l,i)} \phi_{(l,i)}(\theta) \]
Advantages

- Precise error estimation for normalized posterior.
- Otherwise have to rely on relative accuracy.
- From one approximation, you can compute all moments.
- You get a normal approximation for free.
- Check for identification.
Numerical results

\[ f(x) = \prod_{i=1}^{d} \frac{1}{2s} \left[ 1 + \cos \left( \frac{x(i) - \mu}{s} \pi \right) \right] \]
Numerical results
## Integration problem

<table>
<thead>
<tr>
<th># points</th>
<th>145</th>
<th>321</th>
<th>705</th>
<th>1537</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse grids</td>
<td>0.8083</td>
<td>9.26e-4</td>
<td>1.14e-6</td>
<td>1.13e-9</td>
</tr>
<tr>
<td>MC</td>
<td>0.1685</td>
<td>0.0974</td>
<td>0.0334</td>
<td>0.0236</td>
</tr>
</tbody>
</table>

**Table:** $L_\infty$-error quadrature versus MC, 2D
Numerical results

![Numerical results graph](image)

- Grid points – log scale
- $L_\infty$ error – log scale

Data points and trend lines indicating the relationship between grid points and $L_\infty$ error on a log scale.
Numerical results

"Sum of normals"
Integration problem

<table>
<thead>
<tr>
<th># points</th>
<th>1537</th>
<th>3329</th>
<th>7169</th>
<th>15361</th>
<th>32769</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse grids</td>
<td>0.43</td>
<td>0.03</td>
<td>3.12e-006</td>
<td>1.21e-010</td>
<td>1.41e-014</td>
</tr>
<tr>
<td>MCMC</td>
<td>0.05</td>
<td>0.036</td>
<td>0.021</td>
<td>0.01</td>
<td>0.007</td>
</tr>
</tbody>
</table>

**Table:** Error for MCMC versus integration, 2D
Numerical results

\[ f(x) = a \cdot e^{-\frac{1}{2} \cdot (x-\mu) \Sigma (x-\mu)} + b \cdot e^{(x-\mu)^2 \Sigma (x-\mu)^2} \]
Numerical results
Numerical results
Result

- This solves three problems:
  - 1. Method is fast.
  - 2. Error estimation is accurate.
Conclusion

- Frank Schorfheide said at the macro annual:
- "This is heavy computing: Don’t try this at home!"
- We’ll put a Matlab code on the web.
- Please do try it at home!
- We can improve along many dimensions: polynomial exactness, better code, and adaptivity.
Applications

- Sparse grids have been used in several contexts.
- Numerical Integration
- Projection methods (e.g. Kuebler and Krueger)
- Solution to partial differential equations