Geração de cenários

Prof. Alexandre Salles da Cunha

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Belo Horizonte, Brasil

acunha@dcc.ufmg.br

2021/2

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

UFMG - ICEx
DEPARTAMENTO DE CIÊNCIA DA
COMPUTAÇÃO

Relativas à esta apresentação

1. Alan J. King & Stein W. Wallace. Modeling with Stochastic Programming, Springer, 2012. [Cap. 4]
2. G. Cournueols & R. Tütüncü. Optimization Methods in Finance, Cambridge, 2007, [Cap. 16]. (Excelente panorama sucinto sobre Programação Estocástica)

Escopo

- O que considerar em uma geração de cenários
- Estabilidade de soluções
- Exemplo de geração de cenários

Dificilmente os dados brutos que você dispoẽ sobre o problema estarão no formato necessário para uso em um algoritmo de Programação Estocástica.

O que é disponível sobre o problema estocástico ?

1. É conhecida uma distribuição de dados contínua sobre as variáveis aleatórias do problema estocástico.
   Então, será necessário criar uma discretização desta distribuição para uso no algoritmo.
2. São disponíveis apenas dados históricos sobre as variáveis aleatórias.
   Será necessário representar os dados por um distribuição de probabilidades, contínua ou discreta. Se a opção for uma distribuição contínua, caimos no caso acima.

Conclusão: Vai ser necessário discretizar.

Duas visões complementares

1. A discretização (ou criação de árvore de cenários) é parte do processo de modelagem.
2. A discretização é parte do conjunto de procedimentos ou técnicas de solução empregados para resolver o problema estocástico original.

- As duas visões são corretas, mas o fato é que a discretização não é parte dos dados do problema, precisa ser definida e tem impacto muito grande nos resultados.
- Não existe método de discretização que funcione sempre. Assim, é razoável considerar que a discretização faz parte da modelagem do problema.
- Assim, é necessário (ou melhor dizendo, é desejável) ter algumas garantias de que a forma como a discretização é produzida não interfira substancialmente na qualidade da solução encontrada pelo programa estocástico.

Objetivos

1. Discutir por quê esta parte da modelagem, isto é, a construção dos cenários, é tão importante.

2. Tentar ajudar a decidir se a modelagem dos cenários foi bem sucedida, considerando as soluções dos programas estocásticos obtidas.

3. Apresentar algumas formas de como gerar cenários.

1. Dispomos de uma distribuição de probabilidades contínua para as grandezas aleatórias. O problema a resolver é:

$$\min g(x)_{x \in K_1} = \min_{x \in K_1} \left\{ c^T x + \min \mathbb{E}_{\boldsymbol{\xi}} Q(x, \boldsymbol{\xi}) \right\}$$

$\mathcal{Q}(x) = \min \mathbb{E}_{\boldsymbol{\xi}} Q(x, \boldsymbol{\xi})$ é a função recurso exata, não necessariamente linear por partes.

2. Criamos uma discretização: uma árvore de cenários $\mathcal{T}$ (um vértice representando o estágio atual e outro para cada possível cenário no segundo estágio). Seja S o conjunto de cenários associados a esta árvore, s um destes cenários e p_s sua probabilidade de realização. O problema que resolveremos é:

$$\min f(x)_{x \in K_1} = \min_{x \in K_1} \left\{ c^T x + \min \sum_{s \in S} p_s Q(x, \xi^s) \right\}$$

onde $\min \sum_{s \in S} p_s Q(x, \xi^s)$ aproxima a função exata $\min \mathbb{E}_{\boldsymbol{\xi}} Q(x, \boldsymbol{\xi})$

3. A discretização nos leva a um PPL estocástico em dois estágios.

Questões para se ter em mente

- A discretização deve produzir um erro de aproximação pequeno (entre o exato, com a distribuição contínua, por exemplo, e o discretizado), sem tornar o problema de programação estocástica intratável.
- Um bom procedimento de discretização não deve ser observado na solução do problema: o processo de discretização deve ser tal que não é a discretização que determina a solução de otimização, mas sim o modelo algébrico empregado e o modelo das variáveis aleatórias.
- Alguma forma de acesso à qualidade da discretização precisa ser empregada.

Caso em que temos um conjunto de dados

- Se ao invés de distribuições contínuas para as grandezas aleatórias, dispomos de um conjunto de dados históricos, estes dados é o que precisamos usar.
- As mesmas questões colocadas anteriormente são pertinentes neste caso: Equilíbrio entre representatividade dos cenários e tratabilidade computacional.
- Provavelmente, estes dados precisarão ser tratados para serem usados. Por exemplo, séries temporais precisam ser analisadas (agrupadas, classificadas) para construção de cenários futuros.
- Deve-se ter cuidado ao se inferir perfis de distribuições de probabilidades para dados históricos, e criar distribuições contínuas a partir destes dados. Ao fazer isso, estamos inserindo informação nos dados que podem não ser plenamente observados nas séries. Exemplo: estimar média, variância de um conjunto de dados e usar estes valores para criar uma distribuição normal...

Caso de interesse: problema linear estocástico de dois eságios.

- A árvore de cenários é na verdade *um galho* (contém a raiz e todos os demais nós são folhas).
- Contém um nó para o *hoje* no primeiro estágio e
- um nó para cada possível *amanhã*, no segundo estágio.

Dilema:

- Muitos nós no segundo estágio (representando bem uma distribuição contínua): podem dificultar a resolução do problema de otimização.
- Poucos nós no segundo estágio: facilitam a resolução do problema de otimização, mas não representam adequadademente a incerteza ou a distribuição (contínua, por exemplo).

O que faz uma discretização boa ?

- Deve funcionar como se a distribuição original fosse empregada, ou seja, não deve afetar a solução ótima.
- Duas discretizações distintas podem fornecer resultados idênticos dependendo do modelo empregado. Exemplo: duas distribuições com mesma média e variância (e momentos cruzados) em um modelo do tipo de Markowitz.
- Quanto mais aspectos forem necessários a considerar em uma discretização (momentos, valores extremos, co-variâncias, por exemplo), mais cenários serão necessários para uma boa discretização. Mas se o modelo de otimização é insensível a alguns deles, por que considerá-los ?

Em Programação Estocástica, a qualidade de uma discretização é determinada pelo problema de otimização que a emprega.

Testes de estabilidade

Visam ajudar a esclarecer se, o resultado do modelo de otimização depende menos do modelo algébrico que relaciona as variáveis, ou mais de um procedimento de geração de cenários ruim. Estamos testando o modelo de otimização ou o procedimento de geração de cenários ?

1. *In-sample stability*
2. *Out-of-sample stabilitily*

Simplificando a notação, usando apenas $f(\cdot)$ para representar as funções envolvidas, exata ou aproximada. Assumimos que as variáveis de segundo estágio nas definições seguintes são implícitas.

- Problema estocástico com dois estágios.
- $\min_x f(x, \xi)$: representa o problema estocástico de dois estágios verdadeiro, exato, para uma distribuição de probabilidades ξ. Assumimos que seja impraticável resolver este problema, razão pela qual, formulamos o próximo problema.
- $\min_x f(x, \mathcal{T})$: representa o problema estocástico linear de dois estágios, para uma árvore de cenários $\mathcal{T}$, que *aproxima* ξ.

In-sample stability

- Não possui uma versão determinística, verifica consistência interna do modelo, certifica uma robustez do processo de discretização.
- Destacamos dois casos, dependendo do processo de discretização propriamente, que pode ser:
  - Estocástico
  - Determinístico

In-sample stability: discretização é estocástica

- Assumimos que o procedimento de discretização possa ser executado diversas vezes, produzindo diversas árvores de cenários, digamos: $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \ldots,$.
- Defina $\hat{x}_i \in \arg\min f(x, \mathcal{T}_i)$.
- Se

$$f(\hat{x}_i, \mathcal{T}_i) \approx f(\hat{x}_j, \mathcal{T}_j), \text{ para todo } i \neq j,$$

  o modelo apresenta *in-sample stability*.
- Isso nos faz crer que basta rodar o gerador de discretizações e tomar a primeira árvore de cenários produzida que temos consistência interna do modelo.

In-sample stability: discretização é determinística

- O procedimento de discretização é determinístico, mas podemos controlar a dimensão das árvores de cenários. Assim, assumimos que $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \ldots$, representam árvores de cenários distintas, de tamanhos diferentes, mas não muito díspares.
- Usamos o mesmo critério anterior para caracterizar a estabilidade interna.
- Os valores de $f(\hat{x}_i, \mathcal{T}_i)$ devem variar pouco quando tomamos árvores de diferentes tamanhos e há estabilidade interna.
- Se, ao acrescentar alguns poucos cenários, há uma discrepância grande no valor de $f(\cdot, \cdot)$, há indícios de algum problema de modelagem.

Por quê medimos a estabilidade em termos de $|f(\hat{x}_i, \mathcal{T}_i) - f(\hat{x}_j, \mathcal{T}_j)|$ e não por meio de $\|\hat{x}_i - \hat{x}_j\|$?

- Problemas estocásticos costumam ter funções objetivos mais ou menos "planas", o que significa dizer que soluções razoavelmente distintas podem ser mais ou menos equivalentes em termos daquilo que a função objetivo representa.
- Estabilidade na solução ($\hat{x}$) é muito mais difícil de ser obtida.
- Não se deve prosseguir com um modelo que não possui estabilidade *in-sample*.

Se $f(\hat{x}_i, \boldsymbol{\xi}) \approx f(\hat{x}_j, \boldsymbol{\xi})$, caracterizamos a *out-of-sample stability*.

- Vamos assumir que, embora determinar $\min f(x, \boldsymbol{\xi})$ seja impraticável, que consigamos avaliar o *out-of-sample value* $f(\hat{x}, \boldsymbol{\xi})$ para um $x = \hat{x}$ específico. Isto significa que, para x fixo em $\hat{x}$, devemos avaliar a expectativa do recurso mínimo associado a $\hat{x}$.
- Por exemplo, se $\boldsymbol{\xi}$ é discreto mas contém muitos cenários de forma que seja necessário discretizar em árvores de cenários $\mathcal{T}$ que aproximam $\boldsymbol{\xi}$. Neste exemplo, obter $f(\hat{x}, \boldsymbol{\xi})$ corresponde a resolver um grande número de problemas de segundo estágio para $x = \hat{x}$.
- Em casos em que $\boldsymbol{\xi}$ é uma distribuição contínua, avaliar $f(\hat{x}, \boldsymbol{\xi})$ pode envolver técnicas de simulação.

Out-of-sample stability

- Se avaliar $f(\hat{x}, \xi)$ for impraticável, um teste mais fraco que pode caracterizar *out-of-sample* stability é

$$f(\hat{x}_i, \mathcal{T}_j) \approx f(\hat{x}_j, \mathcal{T}_i)$$

In-sample vs out-of-sample stability

- Um modelo que apresenta *in-sample-stability* pode ser ruim, não apresentando *out-of-sample stability*.
- Por exemplo, se o processo de discretização sistematicamente (in-sample) evita uma região da distribuição original e esta região tem impacto central na função objetivo.
- O inverso, estabilidade *out-of-sample* e instabilidade *in-sample* também pode ocorrer. Observe que o argumento usado para o teste out-of-sample é a solução $\hat{x}$.

Exemplo de geração de cenários

- Geração de cenários a partir de dados históricos.
- Vamos empregar modelos auto-regressivos para gerar uma série de dados para construção de cenários.
- O exemplo vai considerar retornos de 3 ativos financeiros, mas pode ser aplicado para outros contextos.

1. Dados históricos para um vetor r de variáveis aleatórias.
2. r_t : representa o vetor em um dado período de tempo $t \in \{1, \ldots, T\}$
3. Um modelo auto-regressivo é definido da seguinte forma:

$$r_t = D_0 + D_1 r_{t-1} + D_2 r_{t-2} + \cdots D_p r_{t-p} + \epsilon_t,\ t = 1+p, \ldots, T \quad (1)$$

onde p representa o número de intervalos de tempo empregados no modelo, $\epsilon_t \approx N(0, \Sigma)$ é um vetor de perturbações independentemente distribuídas de média zero, e as matrizes $D_0, D_1, \ldots, D_p$ devem ser estimadas via regressão linear, por exemplo, a partir dos dados históricos e do modelo (1).

4. Com o modelo $D_0, D_1, \ldots, D_p$ aprendido e a matriz de covariância Σ relativa aos dados históricos, geramos perturbações ϵ para construir uma árvore de cenários.

Exemplo: dados de retornos para 3 ativos de 1960 a 2003

1. 3 ativos: ativo 1 (s), ativo 2 (b) e ativo 3 (m)
2. Modelo a ser aprendido: regressão linear, com $p = 1$:

   Para todo $t = 2, \ldots, T$:

$$\begin{pmatrix} s_t \\ b_t \\ m_t \end{pmatrix} \approx \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix} + \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix} \begin{pmatrix} s_{t-1} \\ b_{t-1} \\ m_{t-1} \end{pmatrix}$$

Exemplo: dados de retornos para 3 ativos de 1960 a 2003

1. Usando os dados do arquivo `retornos.csv` e o notebook associado, obtemos:
2. $D_0 = d = \begin{pmatrix} 0.078 \\ 0.047 \\ 0.016 \end{pmatrix}$
3. $D_1 = \begin{pmatrix} -0.058 & 0.219 & 0.448 \\ -0.053 & -0.078 & 0.707 \\ 0.033 & -0.044 & 0.746 \end{pmatrix}$
4. Desvio padrão dos erros de predição do ajuste:
   $\sigma_s = 0.165, \ \ \sigma_b = 0.103, \ \ \sigma_m = 0.021$
5. Criamos uma matriz Σ diagonal, com diagonal igual a $\sigma_s, \sigma_b, \sigma_m$ e utilizamos uma $N(0, \Sigma)$ para gerar um cenário para os retornos do próximo período (ano $= 2004$) a partir dos dados disponíveis para 2003 e D_0, D_1.

Exemplo: Cenário para 2004, o primeiro nó da árvore de cenários

1. Input para a construção da primeira ramificação, a partir do nó 2003 da árvore de cenários:
   - `retornos.csv`: $s_{2003} = 0.2868, b_{2003} = 0.0054, m_{2003} = 0.0098$
   - $\sigma_s = 0.165, \ \sigma_b = 0.103, \ \sigma_m = 0.021$
2. Usando um gerador de números aleatórios para uma distribuição normal multivariada $N(0, \Sigma)$, obtemos, por exemplo:
   - $\epsilon^s_{2004} = -0.186, \epsilon^b_{2004} = 0.052, \epsilon^m_{2004} = 0.007.$
   - $s_{2004} = 0.285, b_{2004} = -0.013, m_{2004} = 0.026.$
3. Geramos diversos outros cenários para 2004, e para cada um deles, cenários para 2005. A partir destes, geramos cenários para 2006, etc, ...
4. Para esta aplicação específica, é necessário identificar e corrigir possibilidades de arbitragem nos cenários gerados (assunto não tratado aqui).