Optimization with RAVEN

RAVEN Workshop

www.inl.gov

Idaho National Laboratory

Outline

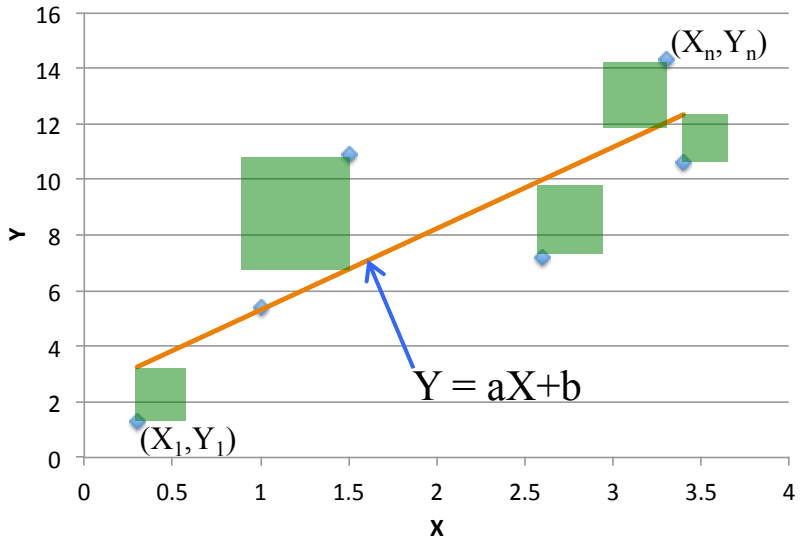
- Brief introduction on Optimization algorithms in RAVEN:
  - SPSA
- Optimization examples
  - Optimization of a deterministic model
  - Optimization of a stochastic model

Optimization algorithms in RAVEN

- RAVEN currently provides 2 optimizations schemes:
  - Gradient Descent (Finite Difference)
  - Simultaneous Perturbation Stochastic Approximation (SPSA)
- Embedded de-noising capabilities
- Multi-Objective Optimization can be performed with a blend of Optimization and Ensemble Modeling

Stochastic optimization

Try to fit straight line to a set of points (X_i, Y_i)



- Define target function to minimize
  - For example least squares

$$T(a,b) = \sum_{i=1}^{n} \left(aX_i + b - Y_i \right)^2$$

- Minimize with algorithm
  - Find "a" and "b" such that *T(a,b)* is minimal
  - For example "gradient descent" algorithm $-\nabla T(a,b)$ move in direction
- What if lots of decision variables (T=T(a,b,c, …)) or system is stochastic ($X_i \rightarrow Y_i$ not unique)?
  $-\nabla T(a,b)$ may be difficult or impossible to evaluate

⇒**Stochastic optimization algorithms**

Simultaneous Perturbation Stochastic Approximation (SPSA) Algorithm for Efficient Stochastic Optimization

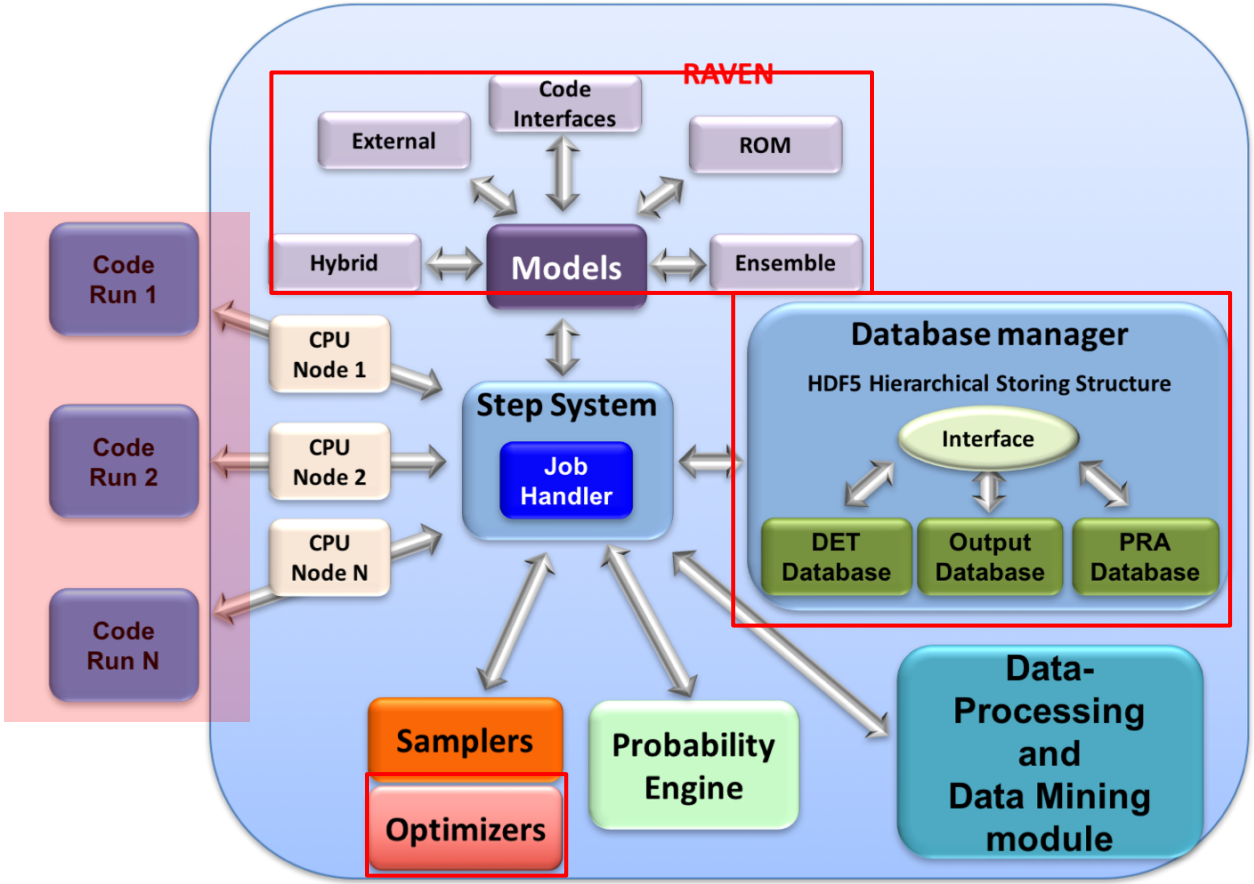
- No gradient formula or gradient measurements needed
- Approximate the gradient based on two function evaluations
  - Simultaneously approximate gradient for all variables
  - Regardless of the number of decision variables
- Converge to optimal in stochastic sense
  - Inaccurate estimation of gradient may temporarily lead to worse solution
- There has been a growing interest in stochastic optimization algorithms that do not depend on direct gradient information or measurements
  - Applicable in areas such as statistical parameter estimation, (nonlinear) feedback control, simulation-based optimization, signal and image processing, and experimental design

Advanced SPSA

- **Stochastic De-noising**
  - Find expected values of points and gradients
  - ⇒Multiple evaluations of each sample taken, which will be used to calculate expected values and reduces the stochastic noise
- **Multi-trajectory**
  - avoid local minima
  - ⇒Multiple initial starting points to be identified by the user. If one trajectory begins following another trajectory, the first is eliminated and only the second continues
- **Adaptive Stepping**
  - Slow convergence across a domain
  - ⇒Adaptive step sizing determination algorithm in RAVEN's SPSA algorithm. Gradient for direction information only. Step size based on previous performance

Optimization Modeling Within RAVEN

- All modeling steps that involve optimization are available in RAVEN
  - Set up the optimization problem
  - Perform optimization of the objective function



RAVEN Examples

Example 1 – No stochasticity

Workflow

Optimization Algorithm

↓

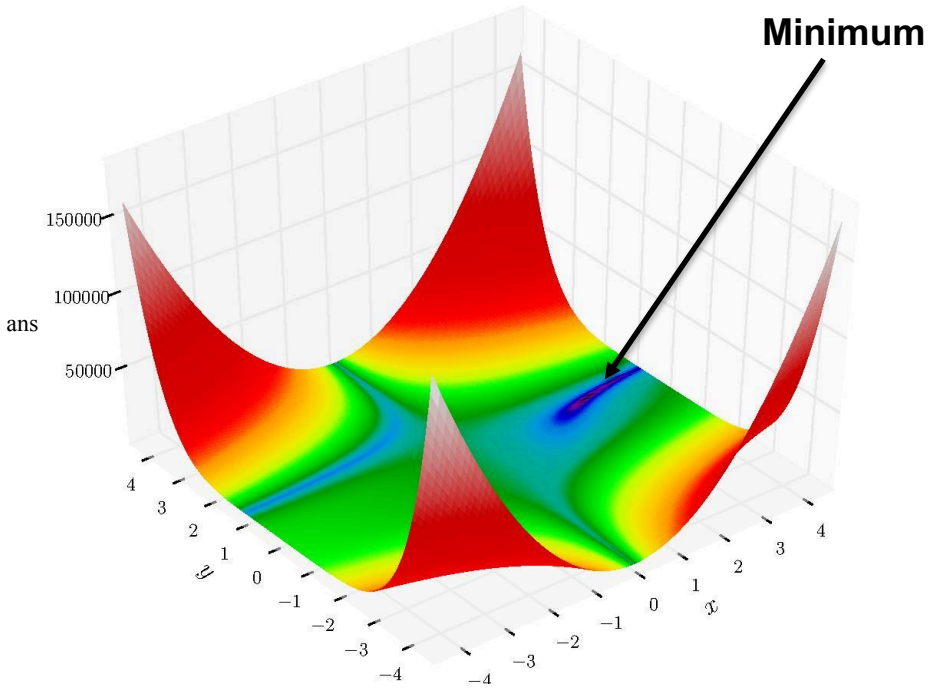
Optimize Model

↓

Export the Results

finite_difference.xml

External model employed: beale.py

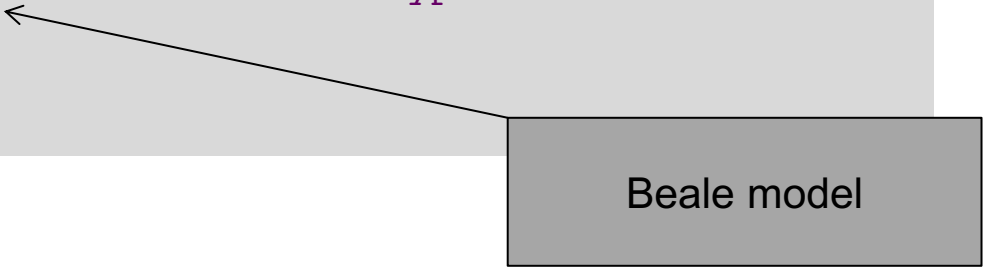


Optimal Minimum: f(3.0,0.5) = 0.0

Construct the Model to optimize

Models	Optimizers	Databases	DataObjects	Steps

```
<Models>
  <ExternalModel ModuleToLoad="beale" name="beale" subType="">
    <variables>x,y,ans</Features>
  </ExternalModel>
</Models>
```



beale.py model:

```
def evaluate(x,y):
  return (1.5 -x +x*y)**2 + (2.25 -x +x*y*y)**2 + (2.625 -x +x*y*y*y)**2
def run(self,inputs):
  self.ans = evaluate(self.x,self.y)
```

Input file name: finite_difference.xml

Select the optimization strategy

Models	Optimizers	Databases	DataObjects	Steps

```
<Optimizers>
  <FiniteDifferenceGradientOptimizer name="opter">
    <initialization>
      <limit>2000</limit> <initialSeed>02201986</initialSeed>
      <thresholdTrajRemoval>1.e-5</thresholdTrajRemoval>
      <writeSteps>every</writeSteps>
    </initialization>
    <convergence>
      <gradientThreshold>1e-1</gradientThreshold>
      <realtiveThreshold>1e-7</realtiveThreshold>
      <gainGrowthFactor>1.5</gainGrowthFactor>
      <gainShrinkFactor>1.25</gainShrinkFactor>
    </convergence>
    <variable name="x">
      <upperBound>4.5</upperBound><lowerBound>-4.5</lowerBound>
      <initial>3.0</initial>
    </variable>
    <variable name="y">
      <upperBound>4.5</upperBound><lowerBound>-4.5</lowerBound>
      <initial>3.0</initial>
    </variable>
    <TargetEvaluation class="DataObjects" type="PointSet">optOut</TargetEvaluation>
    <objVar>ans</objVar>
  </FiniteDifferenceGradientOptimizer>
</Optimizers>
```

Optimizer Type

Convergence Settings

Parameter Space

Objective

Inform Optimizer where the output is stored

Input file name: finite_difference.xml

Set up the data containers

Models	Optimizers	Databases	DataObjects	Steps

Input

Output Model

Optimization Export

```
<DataObjects>
    <PointSet name=inputPlaceHolder'>
      <Input>x,y</Input>
    </PointSet>
    <PointSet name=optOut'>
        <Input>x,y</Input>
        <Output>ans</Output>
    </PointSet>
    <PointSet name=opt_export'>
        <Input>trajID</Input>
        <Output>x,y,ans,varsUpdate</Output>
    </PointSet>
</DataObjects>
```

Input file name: finite_difference.xml

Optimize a Model and Create a Database

Models	Optimizers	Databases	DataObjects	Steps

```
<Steps>
  <MultiRun name="optimize">
    <Input          class="DataObjects" type="PointSet"     >inputPlaceHolder</Input>
    <Model          class="Models"      type="ExternalModel">beale</Model>
<Optimizer class="Optimizers"  type="FiniteDifferenceGradientOptimizer">opter</Optimizer>
    <SolutionExport  class="DataObjects" type="PointSet">opt_export</SolutionExport>
    <Output          class="DataObjects" type="PointSet">optOut</Output>
  </MultiRun>
  <IOStep name="print">
    <Input          class="DataObjects" type="PointSet"  >opt_export</Input>
    <Output         class="OutStreams"  type="Print"     >opt_export_fd</Output>
    <Output         class="OutStreams"  type="Plot">
       optimization_history_input_space_fd
    </Output>
    <Output         class="OutStreams"  type="Plot">
       optimization_history_iterations_fd
    </Output>
  </IOStep>
</Steps>
```

Input file name: finite_difference.xml

Ready to Run?

RUN?

Example 2 – Intrinsic stochasticity

Workflow

Optimization Algorithm

↓

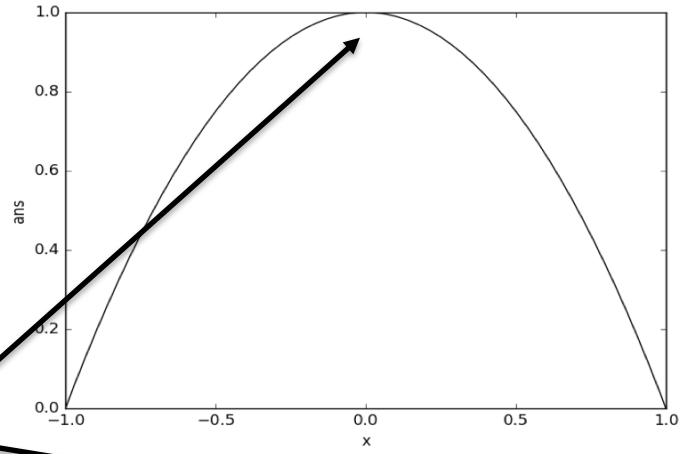
Optimize Model

↓

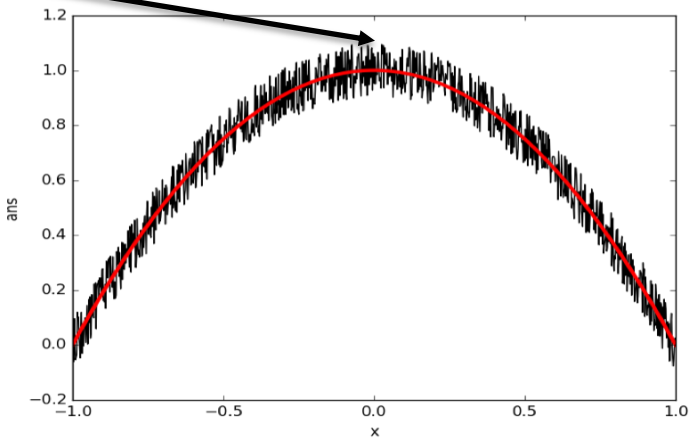
Export the Results

spsa_stochastic.xml

External model employed: stoch_parabola.py



Maximum

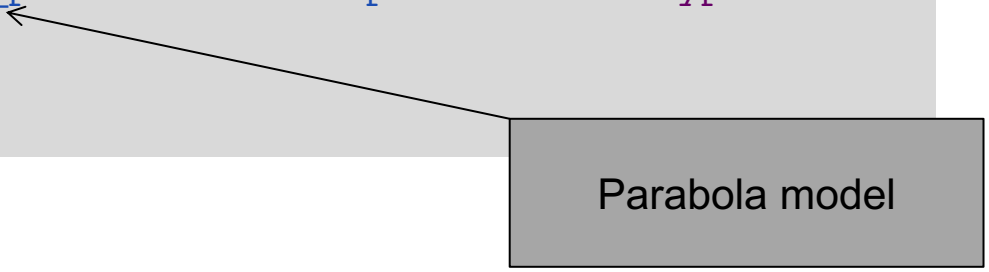


Optimal Maximum: f(0.0) = 1.0

Construct the Model to optimize

Models	Optimizers	Databases	DataObjects	Steps

```
<Models>
  <ExternalModel ModuleToLoad="stoch_parabola" name="parabola" subType="">
    <variables>x,ans</Features>
  </ExternalModel>
</Models>
```



stoch_parabola.py model:

```
def evaluate(x,y):
  ran = random()/10.
  return ran -sum(l*l for l in values)+1.
def run(self,inputs):
  self.ans = evaluate(inputs.values())
```

Input file name: spsa_stochastic.xml

Select the optimization strategy

Models	Optimizers	Databases	DataObjects	Steps

```
<Optimizers>
  <SPSA name="opter">
    <initialization>
      <limit>2000</limit> <initialSeed>02201986</initialSeed>
      <thresholdTrajRemoval>1.e-5</thresholdTrajRemoval>
      <writeSteps>every</writeSteps>
      <type>max</type>
    </initialization>
    <convergence>
      <gradientThreshold>1e-1</gradientThreshold>
      <gainGrowthFactor>1.5</gainGrowthFactor>
      <gainShrinkFactor>1.25</gainShrinkFactor>
    </convergence>
    <variable name="x">
      <upperBound>1.</upperBound><lowerBound>-1.</lowerBound>
      <initial>-0.75</initial>
    </variable>
    <TargetEvaluation class="DataObjects" type="PointSet">optOut</TargetEvaluation>
    <objVar>ans</objVar>
    <parameter>
      <numGradAvgIterations>29</numGradAvgIterations>
    </parameter>
  </SPSA>
</Optimizers>
```

Optimizer Type

Convergence Settings

Parameter Space

Objective

Inform SPSA where the output is stored

Number of grad evaluations

Input file name: spsa_stochastic.xml

Set up the data containers

Models	Optimizers	Databases	DataObjects	Steps

Input

Output Model

Optimization Export

```
<DataObjects>
    <PointSet name=inputPlaceHolder'>
      <Input>x,y</Input>
    </PointSet>
    <PointSet name=optOut'>
        <Input>x</Input>
        <Output>ans</Output>
    </PointSet>
    <PointSet name=opt_export'>
        <Input>trajID</Input>
        <Output>x,ans,varsUpdate</Output>
    </PointSet>
</DataObjects>
```

Input file name: spsa_stochastic.xml

Optimize a Model and Create a Database

Models	Optimizers	Databases	DataObjects	Steps

```
<Steps>
  <MultiRun name="optimize">
    <Input          class="DataObjects" type="PointSet"     >inputPlaceHolder</Input>
    <Model          class="Models"      type="ExternalModel">beale</Model>
    <Optimizer      class="Optimizers"  type="SPSA".        >opter</Optimizer>
    <SolutionExport class="DataObjects" type="PointSet".    >opt_export</SolutionExport>
    <Output         class="DataObjects" type="PointSet".    >optOut</Output>
  </MultiRun>
  <IOStep name="print">
    <Input          class="DataObjects" type="PointSet"  >opt_export</Input>
    <Output         class="OutStreams"  type="Print"     >opt_export_spsa</Output>
    <Output         class="OutStreams"  type="Plot">
       optimization_history_input_space_spsa
    </Output>
    <Output         class="OutStreams"  type="Plot">
       optimization_history_iterations_spsa
    </Output>
  </IOStep>
</Steps>
```

Input file name: spsa_stochastic.xml

Ready to Run?

RUN?