Superpowered Optimization in Python
With Gurobi and Anaconda

Renan Garcia, Ph.D., Gurobi Optimization
Michael Grant, Ph.D., Continuum Analytics
Renan Garcia, Ph.D.

- Optimization Support Engineer at Gurobi Optimization
- Ph.D. in Industrial and Systems Engineering, Georgia Tech
- Expert in optimization modeling and software development
- Over a decade of experience implementing decision support systems
Michael C. Grant, Ph.D.

- Senior Solution Architect at Continuum Analytics
- Ph.D. in Electrical Engineering, Stanford University
- Subject matter expert in nonlinear convex optimization
- Developer of the CVX modeling framework
Overview

• The Proposition
• Introducing Gurobi Optimization
• About Continuum Analytics & Anaconda
• The Gurobi Python module
• A simple MIP model using Gurobi & Spyder
• Portfolio optimization using Gurobi, Jupyter, Pandas, & Bokeh
• Wrap up / Call to action
The Proposition

• The Gurobi Python module provides the intuitive feel of a *modeling* language inside a powerful *programming* language

• The Anaconda platform delivers a full application development stack as well as a great interactive environment for data analysis and modeling

• Bringing developers and analysts/modelers together using Python means collaboration has never been easier
The Gurobi Optimizer
Inside the Gurobi Optimizer

• Algorithms for continuous optimization (LP, QP, QCP)
  – Simplex, Barrier
• Algorithms for discrete optimization (MIP, MIQP, MIQCP)
  – Parallel branch-and-bound
• Other algorithms
  – Presolve, parameter tuning, irreducible inconsistent subsystems…
• Command-line interface, full-featured interactive shell
• Programming interfaces
  – C, C++, Java, .NET, Python, MATLAB, R
Gurobi Model Components

• Decision variables
• Objective function
  – minimize $x^TQx + c^Tx + \alpha$
• Constraints
  – $Ax = b$ (linear constraints)
  – $l \leq x \leq u$ (bound constraints)
  – some $x_h$ integral (integrality constraints)
  – some $x_i$ lie within second order cones (cone constraints)
  – $x^TQ_jx + q_j^Tx \leq \beta_j$ (quadratic constraints)
  – some $x_k$ in SOS (special ordered set constraints)
Example: Mixed Integer Program

• Decision variables
• Objective function
  – minimize $x^TQx + c^Tx + \alpha$
• Constraints
  – $Ax = b$ (linear constraints)
  – $l \leq x \leq u$ (bound constraints)
  – some $x_h$ integral (integrality constraints)
  – some $x_i$ lie within second order cones (cone constraints)
  – $x^TQ_jx + q_j^Tx \leq \beta_j$ (quadratic constraints)
  – some $x_k$ in SOS (special ordered set constraints)
Benchmarks – Open Source MIP

• Mittelmann MIPLIB2010 tests, P=1 (>1X means Gurobi wins):
  – [http://plato.la.asu.edu/bench.html](http://plato.la.asu.edu/bench.html)

<table>
<thead>
<tr>
<th></th>
<th>January 2012</th>
<th>January 2013</th>
<th>January 2015</th>
<th>November 2015</th>
<th>% Solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBC</td>
<td>10X</td>
<td>13X</td>
<td>17X</td>
<td>26X</td>
<td>55%</td>
</tr>
<tr>
<td>SCIP</td>
<td>6X</td>
<td>7X</td>
<td>7X</td>
<td>9X</td>
<td>80%</td>
</tr>
<tr>
<td>GLPK</td>
<td>22X</td>
<td>27X</td>
<td>-</td>
<td>-</td>
<td>1%</td>
</tr>
<tr>
<td>LPSOLVE</td>
<td>19X</td>
<td>24X</td>
<td>-</td>
<td>-</td>
<td>6%</td>
</tr>
</tbody>
</table>

• GLPK and LPSOLVE are not currently tested in the public benchmarks
Why Gurobi?

- Performance leader
  - Average 2X improvement with each major release
- Consistent track record of innovation
  - First cloud offering, client-server support, distributed algorithms, …
- Outstanding support
  - Direct access to Ph.D.-level optimization experts
- No surprises
  - Flexible licensing and transparent pricing
- Strong academic offering
  - Free full-featured academic licenses, Take Gurobi with You program
About Continuum Analytics & Anaconda
Continuum Analytics

• Founded in 2012 by Travis Oliphant and Peter Wang
• Key contributors to the Python open source ecosystem
  – Travis: primary developer, NumPy; founding contributor, SciPy
  – PyTables, Pandas, Jupyter/IPython, Matplotlib
  – New projects: Blaze, Numba, Conda
• Commercial efforts:
  – Enterprise software & support
  – Training: Python for Science, Python for Finance
  – Consulting: custom software development
Anaconda for Data Science
Empowering Everyone on the Team

Data Scientist
• Advanced analytics with Python & R
• Simplified library management
• Easily share data science notebooks & packages

Developer
• Support for common APIs & data formats
• Common language with data scientists
• Python extensibility with C, C++, etc.

Ops
• Validated source of up-to-date packages including indemnification
• Agile Enterprise Package Management
• Supported across platforms

Data Engineer
• Powerful & efficient libraries for data transformations
• Robust processing for noisy dirty data
• Support for common APIs & data formats

Business Analyst
• Collaborative interactive analytics with notebooks
• Rich browser based visualizations
• Powerful MS Excel integration

Computational Scientist
• Rich set of advanced analytics
• Trusted & production ready libraries for numerics
• Simplified scale up & scale out on clusters & GPUs
Anaconda for Data Science
Empowering Everyone on the Team

Data Scientist
- Advanced analytics with Python & R
- Simplified library management
- Easily share data science notebooks & packages

Developer
- Support for common APIs & data formats
- Common language with data scientists
- Python extensibility with C, C++, etc.

Ops
- Validated source of up-to-date packages including indemnification
- Agile Enterprise Package Management
- Supported across platforms

Data Engineer
- Powerful & efficient libraries for data transformations
- Robust processing for noisy dirty data
- Support for common APIs & data formats

Business Analyst
- Collaborative interactive analytics with notebooks
- Rich browser based visualizations
- Powerful MS Excel integration

Computational Scientist
- Rich set of advanced analytics
- Trusted & production ready libraries for numerics
- Simplified scale up & scale out on clusters & GPUs
Why Python?

- Designed for readability
- Functional, object-oriented, declarative
- Great glue language for legacy C/Fortran
- Rich standard library
- Incredible community library
- Great online resources
- Great name

“The second-best language for everything”
— Peter Wang, CTO
The Anaconda Python Distribution

• Windows, Mac, Linux; 32/64-bit
• Python 2.x, 3.x — your choice (though do pick 3.x!)
• ~150 packages in the default installation
• conda package and environment management
  – ~350 packages available with a simple conda install
  – Create isolated “silos” for multiple projects
• Like R? Anaconda has that too
# Conda Package Management

<table>
<thead>
<tr>
<th>Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Install</td>
<td><code>conda install pandas numpy=1.9 gurobi</code></td>
</tr>
<tr>
<td>Update</td>
<td><code>conda update gurobi</code></td>
</tr>
<tr>
<td>List</td>
<td><code>conda list</code></td>
</tr>
<tr>
<td>Search</td>
<td><code>conda search gurobi</code></td>
</tr>
<tr>
<td>Remove</td>
<td><code>conda remove matplotlib</code></td>
</tr>
</tbody>
</table>
conda environment management

Create:
conda create -n lp_project anaconda gurobi

Select:
source activate lp_project (Linux/Mac)
activate lp_project (Windows)

Return to default:
source deactivate (Linux/Mac)
deactivate (Windows)

Remove:
conda remove -n lp_project --all
Obtaining Anaconda

From Continuum: https://www.continuum.io/downloads

From Gurobi: http://www.gurobi.com/downloads/get-anaconda
The Gurobi Python Module
Building Models with Gurobi

• Option 1: use a modeling language
  – Very easy to build models
    • Optimization modeling constructs built into language
    – Attractive choice for non-programmers

• Option 2: use a full programming language
  – Much more powerful and flexible development environment
    • Complete access to solver functionality
    • Richer set of language features
  – Natural choice when deploying and/or integrating with applications
Gurobi Python Environment

- High-level optimization modeling constructs embedded in Python

- Design goals:
  - Require minimal programming skills to get started
  - Bring "feel" of a modeling language to the Python interface
  - Allow for code that is easy to write and maintain
  - Maintain unified design across all of our interfaces
  - Remain lightweight and efficient compared to solver alone
  - Support all solver and programming needs
Essential Gurobi Python Constructs

• Objects represent model components (Model, Var, Constr)
• Overloaded operators
  – Arithmetic (+, -, ×, ÷), constraints (≤, =, ≥)
• Aggregate sum operator (quicksum)
• Python provides the rest for representing data, indices and subscripts
  – Lists, tuples, dictionaries, loops, generator expressions, …

Ex: $x_i + y_i \leq 5, \forall i \in I \iff$ for i in I: m.addConstr(x[i] + y[i] <= 5)
Building a Gurobi Model: Six Steps

Create a model
\[ m = \text{Model}() \]

Add variables
\[ x = m.\text{addVar}(...) \]

Commit changes
\[ m.\text{update}() \]

Set objective
\[ m.\text{setObjective}(...) \]

Add constraints
\[ m.\text{addConstr}(...) \]

Optimize
\[ m.\text{optimize}() \]
## Installing the Gurobi Python Module

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add the channel</td>
<td><code>conda config --add channels gurobi</code></td>
<td>Add the Gurobi channel to your conda environment.</td>
</tr>
<tr>
<td>Install Gurobi</td>
<td><code>conda install gurobi</code></td>
<td>Install the Gurobi Python module.</td>
</tr>
<tr>
<td>Retrieve license</td>
<td><code>grbgetkey xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</code></td>
<td>Retrieve your Gurobi license key.</td>
</tr>
<tr>
<td>Test license</td>
<td><code>&gt;&gt;&gt; import gurobipy</code></td>
<td>Test the Gurobi license.</td>
</tr>
<tr>
<td>Test token server</td>
<td><code>gurobi_cl --tokens</code></td>
<td>Test the Gurobi token server.</td>
</tr>
</tbody>
</table>
A Simple MIP Model using the Spyder IDE
Spyder

- **Scientific Python Development Environment**
- Familiar *integrated development environment* (IDE) for users of MATLAB, Visual Studio, etc.
- Built-in editor, file explorer, documentation browser
- Graphical interface for debugging (ipdb)
- IPython console for interactive code execution

http://pythonhosted.org/spyder
Spyder Demo
Portfolio Optimization in Jupyter
Modern Portfolio Theory In One Slide

• First offered by Harry Markowitz in 1952
• Recognized by a Nobel Memorial Prize in Economics
• Associates with each stock an expected return and a risk or volatility measure, and a matrix of correlations between pairs of stocks
• The goal is to study the tradeoff between risk and return
  – Minimize risk for a given target (expected) return
  – Maximize return given an upper bound on risk
• The most basic models are quadratic programs (QPs)
(okay, two slides)

\[ \begin{align*}
\text{minimize} & \quad x^T \Sigma x \\
\text{subject to} & \quad \bar{1}^T x = 1 \\
& \quad \rho^T x = \gamma \\
& \quad x \geq 0
\end{align*} \]

\[ \begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \\
\text{subject to} & \quad \sum_{i=1}^{n} x_i = 1 \\
& \quad \sum_{i=1}^{n} \rho_i x_i = \gamma \\
& \quad x_i \geq 0 \quad i = 1, 2, \ldots, n
\end{align*} \]

\begin{align*}
\begin{array}{l}
x_i \\
\rho_i \\
\sigma_{ii} \\
\sigma_{ij} / \sqrt{\sigma_{ii} \sigma_{jj}}
\end{array}
\end{align*}

\begin{align*}
\begin{array}{l}
\text{amount of investment allocated to stock } i \\
\text{expected return for stock } i \\
\text{variance of the performance for stock } i \\
\text{correlation between stocks } i \text{ and } j
\end{array}
\end{align*}
Jupyter (formerly IPython)

• A web application for interactive data analysis
• Create notebooks containing live code, text, equations, and visualizations
• A great way to prototype new algorithms, document experiments, share reproducible results
• Originally for Python, now there are kernels for over 50 different languages

http://www.jupyter.org
Pandas

- Fast and efficient DataFrame and Series objects
- Read/write CSV, HDF5, Excel, SQL, plain text
- Missing data handling
- Slicing, fancy indexing, subsetting
- Merges, joins
- Split-apply-combine operations (groupby)

http://pandas.pydata.org
Bokeh

Interactive visualization for web browsers

- High performance with streaming data and big data
- In-browser interactivity on otherwise *static* pages
- Bindings for Python, Scala, Julia, R

http://bokeh.pydata.org
Jupyter Demo

https://notebooks.anaconda.org/mcg/portfolio
Closing thoughts
Developers vs. Analysts?

Analysts benefit from Jupyter:
- Interactive, iterative development
- Notebooks combine code, results, visualizations, documentation

Developers benefit from Spyder:
- Traditional IDE organization
- Graphical debugging

Both benefit from straightforward collaboration:
- Code from notebooks readily inserted into standard Python modules
Wrap up

• The Gurobi Python interface brings the expressive power of a modeling language into the powerful Python ecosystem
• Spyder offers a traditional IDE for Python development, including full-featured editing and graphical debugging
• IPython and the Jupyter notebook system provide a powerful paradigm for interactive development, collaboration, and sharing
• The Anaconda platform delivers a suite of powerful open-source tools and connects to world-class commercial software like Gurobi
• Need training, consulting, enterprise support? Continuum can help
Thank you!

Download Gurobi and/or Anaconda and try them for yourself:

www.gurobi.com/get-anaconda

Email: sales@gurobi.com
Sales: sales@continuum.io

Twitter: @Gurobi
Twitter: @ContinuumIO