Part I: An Overview of Sho

Sumit Basu
an introduction
The Sho Team

I’m Erin Renshaw
John Platt
Chuck Jacobs
Sumit Basu

...plus a great deal of help from the fabulous folks of Developer Division
First, a Poll

- Who has used Matlab?
- Who has used Python?
- Who has used R?
- How many have wanted to (or had to) connect these to compiled libraries?
So What is Sho?
Matrix Math and Numeric Libraries

- **Sho Arrays**
  - Dense and sparse arrays, from many types
  - Data Types: Double, Float, Int, Boolean, Object
  - Compact, math-like syntax: \( \mathbf{x} = \text{inv}(\mathbf{A}^T \mathbf{A}) \mathbf{A}^T \mathbf{b} \)
  - Operations on submatrices: \( \mathbf{A}[:,3] = \mathbf{b} \)
  - **FAST**: many operations speeded up by MKL

- **Classes for common decompositions**
  - SVD, LU, QR, Cholesky, Schur, Eigen

Visualization
Console Features

• Tab Completion and Intellisense

```csharp
>>> Forms.Fo<TAB>
FolderBrowserDialog
FontDialog
Form
FormBorderStyle
FormClosedEventArgs
FormClosedEventHandler
FormClosingEventArgs
FormCollection
...
```
How is this Different from Matlab, R, etc.? 

**Typical environment**

- math & visualization

**MSFT data stack**

- Powerful data structures
- .NET libraries
- Great languages/IDEs
- Excel, SQL, Sharepoint
- LINQ and PLINQ
- Web and cloud access

Algorithmic Folks have this, want this

Prototypers want this, have this
The Architecture of Sho

<table>
<thead>
<tr>
<th>Python-specific</th>
<th>CLS libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console</td>
<td>Packages</td>
</tr>
<tr>
<td>IronPython</td>
<td>Numerics</td>
</tr>
<tr>
<td></td>
<td>Plots and Charts</td>
</tr>
<tr>
<td></td>
<td>.NET</td>
</tr>
</tbody>
</table>
Sho is .NET All the Way Down

• IronPython talks directly to .NET
  – Instantiate arbitrary .NET objects
  – Subclass objects and override behaviors
  – Use interfaces, enumerators, etc.
  – No wrapping or decoration needed

• Example: Making a Form

```python
>>> f = System.Windows.Forms.Form()
>>> f.Text = “Hello World”
>>> f.ShowDialog()
```
Using a .NET Library from Sho

```python
>>> load('C:\Windows...\System.Speech.dll')
<Assembly System.Speech, Version=3.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35>

>>> from System.Speech.Recognition import SpeechRecognitionEngine, DictationGrammar
>>> sre = SpeechRecognitionEngine()
>>> sre.SetInputToDefaultAudioDevice()
>>> sre.LoadGrammar(DictationGrammar())
>>> res = sre.Recognize()

the quick brown fox jumped over the lazy dog

>>> res.Text
'the quick brown fox jumped over the least got when'
```
Using a C# File Directly from Sho

>>> load("IFilterReader.cs")
>>> import IFilter

>>> txt = IFilter.DefaultParser.Extract ("MySong.pdf")

>>> print txt[:1000]
MySong: Automatic Accompaniment Generation for Vocal Melodies
Ian Simon University of Washington Seattle, WA
iansimon@cs.washington.edu Dan Morris Microsoft Research Redmond, WA dan@microsoft.com
Sumit Basu Microsoft Research Redmond, WA sumitb@microsoft.com

ABSTRACT

MySong: Automatic Accompaniment Generation for Vocal Melodies
Ian Simon University of Washington Seattle, WA
iansimon@cs.washington.edu Dan Morris Microsoft Research Redmond, WA dan@microsoft.com
Sumit Basu Microsoft Research Redmond, WA sumitb@microsoft.com
Using a Web Service from Sho

c:\src\sho\playpen>"c:\Program Files\Microsoft SDKs\Windows\v6.0A\Bin\wsdl.exe"


Writing file 'c:\src\sho\playpen\LiveSearchService.cs'.

>>> load("LiveSearchService.cs")
>>> from livesearch import LiveSearchService, SearchRequest, SourceType

>>> sc = LiveSearchService()

>>> r = SearchRequest()

>>> r.AppId = "XXXXXXXX"

>>> r.Sources = System.Array.CreateInstance(SourceType, 1)

>>> r.Sources[0] = SourceType.Web

>>> r.Query = "Internships at Microsoft Research"

>>> response = sc.Search(r)

>>> for elt in response.Web.Results:
   print '['+elt.Title+']', elt.Description, '\n', elt.Url, '\n
[Apply for an Internship - Microsoft Research] Intern applicants with strong academic achievement in fields such as Computer Science, Electrical Engineering, Math, or Social Sciences with a focus on technology are preferred.

Easy to build GUIs

- Create WinForms GUIs in Sho
- Use GUIs built in Visual Studio
- WPF windows/applications/XAML
Sho Packages

- **Released**
  - Statistics/Random Number Generation
  - Optimization
  - Signal Processing
  - Database (SQL) Access
  - Cluster Computing (HPC)
  - Cloud Computing (Azure)
- **To Be Released Soon**
  - ShoCGI (dynamic web apps)
  - Kinect
- **Under Development**
  - Machine Learning
  - Image Processing
  - Graph Layout
# original function with one parameter value and data directory
>>> res = basisboost.clusterproc((15,2,1), 0.05, 
   "c:/researchjournal/briefcase/sentimentdata/cluster")

# set up cluster
>>> cl = HPC.clustersetup("RR1-N13-09-H44", 
   "/MSR-ARRAYS/SCRATCH/MSR-POOL/09-H44/redmond/sumitb")

# run on cluster
>>> session = HPC.clusterrun(cl, basisboost.clusterproc, 
   (15,2,1), 
   basisboost.createseq(20*list(drange(0.05,0.8,0.05))), 
   "c:/researchjournal/briefcase/sentimentdata/cluster")

# get results
>>> clusterresults = session.getresults()

Why waste only one computer’s time when I could waste thousands of computers’ time at once?
ShoCGI

- Create dynamic web applications using your Sho code with a minimum of extra work
- Cookie support and persistent session store
- Include Sho Plots in your pages
- Create multi-page applications with logins, etc., in a single Python file in less than a hundred lines of code
Kinect for Sho

• Based MSR’s KSDK (to be released this spring)
• Get real-time skeleton data
• Use Sho viz tools to make real-time, animated skeletons in about 50 lines of code
SHO and Visual Studio

Python Tools for Visual Studio

- Syntax highlighting
- Smart indenting
- Intellisense
- Go to definition
- Find all references
- Cross-language debugging
- Inspect python and .net values
PYTHON TOOLS FOR VISUAL STUDIO DEMO
using ShoNS.Array;
using ShoNS.Visualization;
using ShoNS.MathFunc;
namespace CSharpExample
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create an array to work with
            DoubleArray x = ArrayRandom.RandomDoubleArray(10, 10);
            x = x * x.T;
            SVD svd = new SVD(x);
            // Create a figure and display a barchart of the
            // logs of the singular values
            IFigure f1 = ShoPlotHelper.Figure();
            DoubleArray logVals = ArrayMath.Log(svd.D.Diagonal + 1.0);
            f1.Bar(logVals);
        }
    }
}
How Do I Get Started?

• The Site/Blog/Forum/etc.: http://research.microsoft.com/sho

• The Book:
  – In PDF and HTML

• The API Reference:
  – MSDN-style docs for C#
Take-Home Messages

• Sho has linear algebra, visualization, and packages, both in the REPL and from C#
• Sho lets you move smoothly between script and compiled code
• Sho lets you use existing .NET libraries

http://research.microsoft.com/sho
Part II: Sho Workshop

Sumit Basu and Jan Vitek
Format and Logistics

• Install Sho and PTVS
  – [http://research.microsoft.com/sho](http://research.microsoft.com/sho)
  – [http://pytools.codeplex.com](http://pytools.codeplex.com) (for editing/debugging)

• Option 1:
  – Pick one of the following small exercises
  – Join up with others interested in the same problem
  – Jan and I will wander around helping where needed

• Option 2:
  – We can walk through the problems together

• Code and data for exercises is at
1: Plotting Data

• Plot country populations and areas
• Materials:
  – CSV file containing country names, populations, areas
• Goal:
  – Plot areas vs. populations
  – Add names as labels
  – Fit a line/curve to the data and plot it
2: Text Analysis

- Frequency analysis of text files
- Materials
  - Small corpus of text files (paper abstracts)
- Goals
  - Compute frequency of all words
  - Create bar chart of N most frequent words
3: Labeling Tool (GUI)

• Create a labeling tool for labeling sentence sentiment
• Materials
  – File containing sentences to be labeled
• Goal
  – Simple GUI for labeling
4: Voice Calculator

• Do simple math with voice I/O
• Materials:
  – Speech recognition and synthesis example loop
• Goal
  – Make audio-only calculator allowing commands like add, multiply, subtract, divide