Optimization is a prominent problem throughout Computer Science. We take advantage of the separable nature of certain classes of optimization problems, and solve them by distributing and communicating stages of the problem via UDP; a relatively unexplored exercise in distributed optimization.

Loss Functions

- Quadratic Loss: Least Squares
- Logistic Loss: Logistic Regression
- Hinge Loss: Support Vector Machines

Overview

System Design

Function Parameters: Step size, Matrices

Primal Update Library: Haskell Functions

Dual Update Library: Haskell Functions

Framework Module: Iterative Descent

Optimal Solution

Evaluation Graphs

System Architecture

Initial parameters

<table>
<thead>
<tr>
<th>Node</th>
<th>Neighbors</th>
<th>IP Address</th>
<th>Port Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2, n)</td>
<td>187.217.6.219</td>
<td>4007</td>
</tr>
<tr>
<td>2</td>
<td>(1, 3)</td>
<td>187.217.6.220</td>
<td>3402</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>n</td>
<td>(n-1, 1)</td>
<td>187.217.6.247</td>
<td>5013</td>
</tr>
</tbody>
</table>

Results

Difference in Objective Function

Future Work

- Extend function libraries to cover more families of loss functions
- Incorporate second order information in the updates

Conclusion

- Designed and implemented framework for distributed solving of three classes of optimization
- Communication over UDP; previously only simulated or through message passing
- Extensible and portable framework; foundation to build upon