Parallel Large Scale Feature Selection for Logistic Regression

Sameer Singh\textsuperscript{1} Jeremy Kubica\textsuperscript{2} Scott Larsen\textsuperscript{2} Daria Sorokina\textsuperscript{3}

\textsuperscript{1}Department of Computer Science
University of Massachusetts, Amherst

\textsuperscript{2}Google Inc, Pittsburgh

\textsuperscript{3}Department of Computer Science
Carnegie Mellon University

SIAM Data Mining, 2009
Outline

1 Motivation
   - Logistic Regression
   - Feature Selection

2 Single Feature Optimization
   - Method
   - Histogram Approximation
   - Parallelization

3 Experiments
   - UCI Datasets
   - RCV1
   - Parallelization
Outline

1 Motivation
   ■ Logistic Regression
   ■ Feature Selection

2 Single Feature Optimization
   ■ Method
   ■ Histogram Approximation
   ■ Parallelization

3 Experiments
   ■ UCI Datasets
   ■ RCV1
   ■ Parallelization
Logistic Regression

\[ P(y = 1 \mid \vec{x}_i, \vec{\beta}) = \frac{e^{\vec{\beta} \cdot \vec{x}}}{1 + e^{\vec{\beta} \cdot \vec{x}}} \]

\[ \vec{\beta} = \arg\max_{\vec{\beta}} \sum_{i=1}^{N} \left( y_i \ln p_i + (1 - y_i) \ln(1 - p_i) \right) \]
Logistic Regression

\[
P(y = 1 \mid \vec{x}_i, \vec{\beta}) = \frac{e^{\vec{\beta} \cdot \vec{x}}}{1 + e^{\vec{\beta} \cdot \vec{x}}}
\]

\[
\vec{\beta} = \operatorname{argmax}_{\vec{\beta}} \sum_{i=1}^{N} \left( y_i \ln p_i + (1 - y_i) \ln (1 - p_i) \right)
\]
Feature Selection

Features

X

Motivation  Feature Selection

Sameer Singh  (UMass, Amherst)  Parallel Large Scale Feature Selection  SDM 2009
Feature Selection

Features

X
Feature Selection
Feature Selection

X

Features
Feature Selection

For $D$ features, train the model $O(2^D)$ times
Forward Feature Selection

\[ X \]

Features
Forward Feature Selection

Features

X
Forward Feature Selection

Features

X
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection

Features

X
Forward Feature Selection

Features

X
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection

Motivation

Feature Selection

Sameer Singh (UMass, Amherst)

Parallel Large Scale Feature Selection

SDM 2009
Forward Feature Selection

Features

X
Forward Feature Selection

Motivation

Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection

Features

X
Forward Feature Selection

Features

X
Forward Feature Selection

X

Features
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection

Motivation

Feature Selection

Sameer Singh (UMass, Amherst)

Parallel Large Scale Feature Selection

SDM 2009
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection

Motivation

Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection
Forward Feature Selection

$X$
Forward Feature Selection
Forward Feature Selection

Motivation

Feature Selection

Sameer Singh (UMass, Amherst)
Forward Feature Selection

For \( D \) features, train the model \( O(D^2) \) times
Outline

1. Motivation
   - Logistic Regression
   - Feature Selection

2. Single Feature Optimization
   - Method
   - Histogram Approximation
   - Parallelization

3. Experiments
   - UCI Datasets
   - RCV1
   - Parallelization
Single Feature Optimization

$\vec{\beta}$
Single Feature Optimization

\[ \vec{\beta} \]
Single Feature Optimization

\[ \vec{\beta} \]

\[ \beta'_d \]
Newton’s Method

\[
p_{id} = \frac{e^{\beta \cdot \bar{x}_i + x'_i \beta'_d}}{1 + e^{\beta \cdot \bar{x}_i + x'_i \beta'_d}}
\]

\[
\beta'_d = \arg\max_{\beta'_d} \sum_{i=1}^{N} \left( y_i \ln p_{id} + (1 - y_i) \ln(1 - p_{id}) \right)
\]
Newton’s Method

\[
p_{id} = \frac{e^{\bar{\beta} \cdot \tilde{x}_i + x'_i \beta'_d}}{1 + e^{\bar{\beta} \cdot \tilde{x}_i + x'_i \beta'_d}}
\]

\[
\beta'_d = \arg\max_{\beta'_d} \sum_{i=1}^{N} \left( y_i \ln p_{id} + (1 - y_i) \ln(1 - p_{id}) \right)
\]
Newton’s Method

\[ p_{id} = \frac{e^{\bar{\beta} \cdot \bar{x}_i + x'_{id} \beta'_d}}{1 + e^{\bar{\beta} \cdot \bar{x}_i + x'_{id} \beta'_d}} \]

\[ \frac{\partial L}{\partial \beta'_d} = \sum_{i=1}^{N} x'_{id} (y_i - p_{id}) \]

\[ \frac{\partial^2 L}{\partial \beta'_d^2} = -\sum_{i=1}^{N} p_{id} (1 - p_{id}) x'_{id}^2 \]
As $N$ grows, Newton’s method slows down considerably
- $B$ bins, based on predicted probability of base model
  - using only $\hat{\beta}$ and $\hat{x}$
- Newton’s method dependent on $B$ instead of $N$
  - $N \gg B$
Map Reduce implementation

*Map*: Parallel over *records*

- **Input**: Base features $\tilde{x}_i$, class $y_i$, new features $\tilde{x}_i'$
- Predict using the base model $p_i$
- **Output**: $(x'_{id}, \langle y_i, p_i \rangle)$ for every feature $x'_{id}$ in $\tilde{x}_i'$

*Reduce*: Parallel over *features*

- **Input**: $x'_{d}, \langle y_i, p_i \rangle^n$
- Use Newton’s method to find $\beta'_d$ that maximizes scoring measure
- With or without histogram approximation
- **Output**: Estimated coefficient $\beta'_d$

Evaluate the coefficients on test dataset to evaluate utility
Map Reduce implementation

- **Map**: Parallel over records
  - **Input**: Base features $\vec{x}_i$, class $y_i$, new features $\vec{x}'_i$
  - Predict using the base model $p_i$
  - **Output**: $(x'_{id}, \langle y_i, p_i \rangle)$ for every feature $x'_{id}$ in $\vec{x}'_i$

- **Reduce**: Parallel over features
  - **Input**: $x'_d, \langle y_i, p_i \rangle^n$
  - Use Newton’s method to find $\beta'_d$ that maximizes scoring measure
  - With or without histogram approximation
  - **Output**: Estimated coefficient $\beta'_d$

Evaluate the coefficients on test dataset to evaluate utility
Map Reduce implementation

- **Map**: Parallel over records
  - **Input**: Base features $\vec{x}_i$, class $y_i$, new features $\vec{x}_i'$
  - Predict using the base model $p_i$
  - **Output**: $(x_{id}', \langle y_i, p_i \rangle)$ for every feature $x_{id}'$ in $\vec{x}_i'$

- **Reduce**: Parallel over features
  - **Input**: $x_d', \langle y_i, p_i \rangle^n$
  - Use Newton’s method to find $\beta_d'$ that maximizes scoring measure
  - With or without histogram approximation
  - **Output**: Estimated coefficient $\beta_d'$

Evaluate the coefficients on test dataset to evaluate utility
Outline

1 Motivation
   - Logistic Regression
   - Feature Selection

2 Single Feature Optimization
   - Method
   - Histogram Approximation
   - Parallelization

3 Experiments
   - UCI Datasets
   - RCV1
   - Parallelization
Methods

- **IRLS**: Iteratively Re-weighted Least Squares
  - P. Komarek and A. Moore, *ICDM 2005*
  - Fast, efficient single machine implementation of Logistic Regression
  - Retrain classifier for each candidate feature

- **SFO**: Single Feature Optimization
  - Use IRLS to train the “base” model

- **GD**: Gradient Method
  - S. Perkins and J. Theiler, *ICML 2003*
  - Ranks features according to their gradient on training data
  - Parallelize it same way as SFO

---

1 [http://www.autonlab.org/autonweb/10538.html](http://www.autonlab.org/autonweb/10538.html)
## Mushroom Dataset

<table>
<thead>
<tr>
<th>Base Features</th>
<th>Feature Class</th>
<th>IRLS -LL</th>
<th>SFO -LL</th>
<th>Rank</th>
<th>GD Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>odor</td>
<td>0.111</td>
<td>0.076</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>spore-print-color</td>
<td>0.558</td>
<td>0.543</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>gill-color</td>
<td>0.623</td>
<td>0.604</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>stalk-surface-above</td>
<td>0.696</td>
<td>0.692</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>ring-type</td>
<td>0.711</td>
<td>0.687</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>bias, odor</td>
<td>spore-print-color</td>
<td>0.074</td>
<td>0.069</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>stalk-surface-above</td>
<td>0.098</td>
<td>0.090</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>population</td>
<td>0.099</td>
<td>0.092</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>gill-color</td>
<td>0.099</td>
<td>0.091</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>stalk-color-below</td>
<td>0.100</td>
<td>0.086</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

**Table:** The negative test set log-likelihood for the top features in the Mushroom data set as selected by IRLS, the corresponding SFO scores, and rankings from SFO and the gradient method.
InternetAds Dataset

Figure: Coverage of the IRLS ranking by SFO and the Gradient method for the Internet Ads data. The features were ranked by test set log-likelihood.
### Table: Top 5 features & estimated improvement on training set loglikelihood.

<table>
<thead>
<tr>
<th>Round 1</th>
<th>Round 2</th>
<th>Round 3</th>
<th>Round 4</th>
<th>Round 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>bias</td>
<td>bias</td>
<td>bias</td>
<td>bias</td>
<td>bias</td>
</tr>
<tr>
<td>econ 283.7</td>
<td>econ 204.3</td>
<td>defi 110.2</td>
<td>shar 106.7</td>
<td>infl 79.5</td>
</tr>
<tr>
<td>defi 213.7</td>
<td>defi</td>
<td>defi</td>
<td>defi</td>
<td>defi</td>
</tr>
<tr>
<td>infl 190.1</td>
<td>infl 139.3</td>
<td>shar 106.8</td>
<td>shar 82.1</td>
<td>stat 76.5</td>
</tr>
<tr>
<td>gdp 182.9</td>
<td>prof</td>
<td>infl</td>
<td>prof</td>
<td>mood</td>
</tr>
<tr>
<td>muni 176.3</td>
<td>muni</td>
<td>gdp</td>
<td>infl</td>
<td>dig</td>
</tr>
</tbody>
</table>
Experiments
Parallelization

Timing Results

Figure: Timing (10,000,000 records / 100,000 features)
Figure: Speedup (10,000,000 records / 100,000 features)
Summary

- Introduce **Single Feature Optimization (SFO)**
  - *approximation to Forward Feature Selection*
- To scale to large datasets, utilize **MapReduce** for parallelism
- **Histogram** Approximation is used to further scalability

**Future Work:**
- Multiple Feature Optimization
  - *pairs of features*
- Optimize on metrics other than LogLikelihood
Parallel Large Scale Feature Selection for Logistic Regression

Sameer Singh$^1$  Jeremy Kubica$^2$  Scott Larsen$^2$
Daria Sorokina$^3$

$^1$Department of Computer Science
University of Massachusetts, Amherst

$^2$Google Inc, Pittsburgh

$^3$Department of Computer Science
Carnegie Mellon University

SIAM Data Mining, 2009
Histogram Approximation

- For each bin $b$
  - Mean probability $p_{id}$ of the bin $\hat{p}_b$
  - Total number of records in the bin $N_b$
  - Number of records in which $x_d = 1$, $N_b^+$

- Calculate $p'_b$ using $\hat{p}_b$ and $\beta_d$

\[
\frac{\partial L}{\partial \beta'_d} = \sum_{b=1}^{B} N_b^+ - p'_b \cdot N_b
\]

\[
\frac{\partial L}{\partial \beta'^2_d} = - \sum_{b=1}^{B} N_b \cdot p'_b \cdot (1 - p'_b)
\]
Figure: **Map:** operate on training data \((\mathbf{x}_i, y_i, \mathbf{x}'_i)\) to produce intermediate records \((y_i, p_i)\) for each new feature in the record \(\mathbf{x}'_i\). **Reduce:** operate on intermediate records, computing coefficients for the new features \(\beta'_d\).