PARALLEL CLASSIFICATION ALGORITHMS

By:
Faiz Quraishi
Riti Sharma

9th May, 2013
OVERVIEW

- Introduction
- Types of Classification
- Linear Classification
- Support Vector Machines
- Parallel SVM Approach
- Decision Trees
- Parallel Implementation of Decision Trees
- Results of Decision Trees
- Conclusion
INTRODUCTION

- Data comes in a wide variety of diverse forms.
- Understanding and making sense of this vast and diverse collection of data i.e. identifying patterns, trends, etc. is required.
- Example: People classify email messages (spam/not spam).
BASIC MODEL

- **Sensor**: Collect the data.
- **Feature Extractor**: Selecting or extracting some of the features of the data that are important for classification.
- **Classifier**
TYPES OF CLASSIFICATION

- **Supervised**: A training set is available to train the classifier.

- **Unsupervised**: No training set is available and also the number of classes is unknown.
CLASSIFICATION STEPS

- **Training**
  - Each sample in the training set is assumed to belong to a predefined class.
  - These sets are used to construct a model.

- **Testing**
  - The unknown test samples are considered.
  - These are classified using the model constructed using the training set.
TRAINING VS TESTING

Training

Testing

<table>
<thead>
<tr>
<th>NAME</th>
<th>RANK</th>
<th>YEARS</th>
<th>TENURED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>Assistant Prof</td>
<td>3</td>
<td>no</td>
</tr>
<tr>
<td>Mary</td>
<td>Assistant Prof</td>
<td>7</td>
<td>yes</td>
</tr>
<tr>
<td>Bill</td>
<td>Professor</td>
<td>2</td>
<td>yes</td>
</tr>
<tr>
<td>Jim</td>
<td>Associate Prof</td>
<td>7</td>
<td>yes</td>
</tr>
<tr>
<td>Dave</td>
<td>Assistant Prof</td>
<td>6</td>
<td>no</td>
</tr>
<tr>
<td>Anne</td>
<td>Associate Prof</td>
<td>3</td>
<td>no</td>
</tr>
</tbody>
</table>

Training Data

Classifier

Testing Data

Classifier (Model)

Unseen Data

Tenured?

IF rank = ‘professor’ OR years > 6 THEN tenured = ‘yes’
DIFFERENT CLASSIFIERS

- Supervised Classifiers
  - Support Vector Machines
  - Linear Discriminants
  - K-Nearest Neighbors
  - Neural Networks
  - Decision Trees

- Unsupervised
  - K Means
  - Hierarchical Clustering
LINEAR CLASSIFICATION

- Find a linear function
  
  \[ g(x) = w'x + b \]

- A hyper-plane in the feature space
LINEAR CLASSIFICATION

- How would you classify these in order to minimize the error rate?
- Many answers...!!
- Which is the best one??
SUPPORT VECTOR MACHINES

- Classifier with the maximum margin is the best one.

- Boundary
  \[ w^T x^+ + b = 1 \]
  \[ w^T x^- + b = -1 \]

- Margin Width
  \[ M = \frac{2}{||w||} \]
TRAINING SVM

- In order to maximize the margin, we need to minimize $|w|$ subject to the constraints

$$y_i(w^T x_i + w_0) \geq 1 \quad i = 1, \ldots, n$$

- A standard approach to optimization problem with equality and inequality constraints is the Lagrange formalism.

$$L_p = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i(y_i(w^T x_i + w_0) - 1)$$

- where $\alpha$ are the LaGrange multipliers.
PARALLEL SVM

- To solve for support vectors is time consuming.
- As the training set is increased, the time complexity increases.
- Therefore, need for parallelization arises.
- We need to solve for $\lambda$ and $z$ and $H$ is calculated using the training data.
H MATRIX INTO BLOCKS

- Hi and Ai result from partitioning the data evenly among "p" processors.
- This can be used in solving the system of equations by back substitution.

\[
H = \begin{bmatrix}
H_1 & H_2 & \cdots & A_1^T \\
H_2 & H_2 & \cdots & A_2^T \\
\vdots & \vdots & \ddots & \cdots \\
A_1 & A_2 & \cdots & H_p \\
A_1 & A_2 & \cdots & A_p \\
A_1^T & A_2^T & \cdots & 0
\end{bmatrix},
\]

\[
H \begin{bmatrix}
\Delta z \\
\Delta \lambda
\end{bmatrix} = \begin{bmatrix}
 r_c \\
r_b
\end{bmatrix}
\]
PARALLEL SVM (GENERAL)

- The training dataset is divided among the processors.
- Each processor calculates the support vectors according to the dataset provided to him.
- The slaves send their support vectors to the master processor.
- The master processor finds the actual support vectors using the message from slaves.
- Master computes the final decision boundary using the support vectors.
DECISION TREES

• The goal is to create a model that predicts the value of a target variable based on several input variables.

• Each interior node corresponds to one of the input variables.

• A tree can be learned by splitting the source set into subsets based on an attribute value test.

• This process is repeated in a recursive manner.
DECISION TREES

• The leaves represent class labels.

• Branches represent conjunction of features that lead to those class labels.

• These are simple to interpret and understand.
PARALLEL IMPLEMENTATION OF TREE LEARNING

- Decision Trees require each node to possess the "optimal" question.

- Optimal questions require a search over all data, and can be defined by different metrics

- Analyzing every possibility is intensive

- Here parallelization benefits large data sets
Google's PLANET system
  • Stopping criteria $|D| \leq 10$
  • $|D|$ is the amount of data that goes to the node
GOOGLE'S PLANET SYSTEM

- Three Construction Components
  - ModelFile - the model of the tree
  - MapReduceQueue
    - Data too large for memory (not considered)
  - InMemoryQueue - Nodes and their rules
INTERACTIONS

• Controller
  o Has all training data and an empty tree
  o Finds the initial optimal decision question, and places the nodes onto the queue

• Slaves
  o Each slave is assigned a branch node to work on from the queue
  o Returns the optimal question
WALKTHROUGH

- **Controller (Master)**
  - Decides initial split
  - Hands branches to slaves

- **Slaves**
  - Receive a splitting task, and return optimal split to Controller

- **Controller**
  - Updates tree model and adds new branches to queue (if there are any)
  - Repeat until stopping criterion
EXAMPLE

\[
\begin{align*}
\text{A: } X_1 &< v_1 \\
|D| &= 10, |D| &= 90 \\
&= 0.42266
\end{align*}
\]

\[
\begin{align*}
\text{B: } X_2 &\in \{v_2, v_3\} \\
|D| &= 45, |D| &= 45
\end{align*}
\]

\[
\begin{align*}
\text{C: } |D| &= 20, |D| &= 25, |D| &= 15 \\
\text{D: } |D| &= 30
\end{align*}
\]

\[
\begin{align*}
\text{E, F, G, H}
\end{align*}
\]
RESULTS FROM PLANET
CONCLUSIONS

- Classification is computationally expensive

- The time complexity increases as the size of the training data increases.

- Parallelization significantly reduces the computation time.

- Mutually exclusive dataset make parallelization easy
QUESTIONS