

Large Scale Image Classification Based on a Multi-Class Logistic Regression Model

Presented By: Tianlong Song

Department of Electrical & Computer Engineering

Michigan State University

East Lansing, Michigan 48824, USA.

Email: songtia6@msu.edu

April 28, 2015

Outline

- Motivation
- Multi-Class Logistic Regression
- Training Example Sampling
- Experimental Results
- Extra Efforts

Motivation

- **Challenges**

- Massive training examples
- High feature dimension
- High class diversity

- **Why Logistic?**

- Flexibility in No. of examples involved in each iteration
- No. of parameters increases with feature dimension and class diversity

Multi-Class Logistic Regression

- Binary Case

$$\ln \frac{p(y_i = 1|\mathbf{x}_i)}{p(y_i = -1|\mathbf{x}_i)} = \mathbf{w}^T \mathbf{x}_i \quad (1)$$

- Multi-Class Case

$$\ln p(y_i|\mathbf{x}_i) \propto \mathbf{w}_{y_i}^T \mathbf{x}_i \quad (2)$$

- Probability Constrain

$$\sum_{y_i=1}^K p(y_i|\mathbf{x}_i) = 1 \quad (3)$$

Multi-Class Logistic Regression–cont'

- Formal Representation

$$p(y_i|\mathbf{x}_i) = \frac{1}{Z} e^{\mathbf{w}_{y_i}^T \mathbf{x}_i} \quad (4)$$

with

$$Z = \sum_{j=1}^K e^{\mathbf{w}_j^T \mathbf{x}_i} \quad (5)$$

Multi-Class Logistic Regression–cont'

- MLE Optimization

$$\begin{aligned}\max_{\mathbf{W}} \mathcal{L}(\mathbf{W}) &= \max_{\mathbf{W}} \sum_{i=1}^N \ln p(y_i | \mathbf{x}_i) \\ &= \max_{\mathbf{W}} \sum_{i=1}^N \ln \frac{e^{\mathbf{w}_{y_i}^T \mathbf{x}_i}}{\sum_{j=1}^K e^{\mathbf{w}_j^T \mathbf{x}_i}}\end{aligned}\tag{6}$$

Multi-Class Logistic Regression–cont'

- Gradient Calculation

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{w}_k} &= \sum_{i=1}^N \mathbf{x}_i \left(I(y_i = k) - \frac{e^{\mathbf{w}_k^T \mathbf{x}}}{\sum_{j=1}^K e^{\mathbf{w}_j^T \mathbf{x}}} \right) \\ &= \sum_{i=1}^N \mathbf{x}_i (I(y_i = k) - p(y_i = k | \mathbf{x}_i))\end{aligned}\tag{7}$$

- Iteration Formula

$$\mathbf{w}_{k,t+1} = \mathbf{w}_{k,t} + \eta_t \frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{w}_k}\tag{8}$$

Training Example Sampling

- **Involve all the training examples in each iteration?**
 - No!!! Computationally expensive.
- **Sampling** Only a very small amount of memory needed to accommodate these sampled examples.
 - How many training examples in each iteration?
Experimentally $N_s = 250$ is chosen.
 - How many iterations needed?
 - How to implement fast sampling?
Text documents \rightarrow Binary documents: replace the time-consuming sequential access with random access

Experimental Results

- Cross Validation Accuracy and Mean Average Precision**

Table 1: Cross Validation Report on the Proposed Approach Under Different Numbers of Iterations

NO. OF ITERATION	ACCURACY /MAP	TRAINING TIME	TESTING TIME
100	29.11%/22.40%	4MIN57S	30S
200	30.29%/25.23%	9MIN9S	38S
500	31.53%/30.53%	18MIN57S	35S
1000	32.40%/35.56%	37MIN46S	32S
2000	32.99%/39.37%	1H7MIN31S	31S
5000	33.74%/44.31%	1H54MIN28S	30S
10000	34.03%/47.98%	3H58MIN20S	31S
20000	34.40%/49.94%	6H37MIN15S	30S

Experimental Results–cont'

- Time and Memory Cost of the Final Training and Prediction

Table 2: Time and Memory Cost of the Final Training and Prediction

NO. OF ITERATIONS	TRAINING TIME	PREDICTION TIME	PEAK MEMORY
5000	3H27MIN43S	49s	1.764MB

- MATLAB Profile Report

Profile Summary

Generated 16-Apr-2013 10:28:53 using cpu time.

<u>Function Name</u>	<u>Calls</u>	<u>Total Time</u>	<u>Self Time*</u>	<u>Allocated Memory</u>	<u>Freed Memory</u>	<u>Self Memory</u>	<u>Peak Memory</u>
<u>main</u>	1	415.952 s	0.015 s	58992112.00 Kb	58991952.00 Kb	-1156.00 Kb	1764.00 Kb

Extra Efforts

- **Smart Sampling** Untested.
 - Unbalanced class distribution.
 - Equal chance of being sampled for each class.
- **Principal Component Analysis (PCA)** Failed.
- **Bad Data Elimination** Failed.
 - Assumption: Some data are abnormal and misleading.
 - Methodology: Judge by their deviation from the centroid.
- **Feature Vector Expansion** Failed.
 - Assumption: Some data are undistinguishable under a low degree.
 - Methodology: Expand the feature vectors in a polynomial way.



Thank you!

Questions?