Aggressive Double Sampling for Reducing Multi-class Classification to Binary Classification

Bikash Joshi (PhD Student)
AMA team, LIG

Supervised By:
Prof. Massih-Reza Amini and Dr. Franck Iutzeler

March 20, 2017
Outline

1. Introduction
2. Multiclass to Binary Reduction
3. Double-Sampled Multiclass to Binary Reduction
4. Experimental Results
5. Conclusion
Outline

1. Introduction
2. Multiclass to Binary Reduction
3. Double-Sampled Multiclass to Binary Reduction
4. Experimental Results
5. Conclusion
Multiclass Classification: Introduction

- Finite set of categories ($K > 2$)
- Popular applications: image and text classification.
Multiclass classification: Related Work

1 Combined approaches based on binary classification:
   - One-Vs-Rest
     - One binary problem for each class
     - \( K \) binary problems
     - \( O(K \times d) \)
   - One-Vs-One
     - One binary problem for each pair of classes
     - \( O(K^2 \times d) \)

2 Uncombined Approaches
   - for example: multiclass SVM, MLP
   - One scoring function per class

3 Logarithmic Time Algorithms
   - For example: logTree, Recall-Tree
   - Each leaf node represents a class
   - \( O(\log K) \)
Multiclass classification: Challenges

- The number of classes, $K$, in new emerging multiclass problems, for example in text and image classification, may reach $10^5$ to $10^6$ categories.

- For example:

  - $4 \times 10^6$ sites
  - $10^6$ categories
  - $10^5$ editors
  - Imbalanced nature of hierarchies
Multiclass classification: Challenges

- Class imbalance problem
- Majority of classes have few representative examples
- Long tailed distribution
Text Classification:

**Task:** Automatic classification of an example text to one of fixed set of categories.

**Feature Representation:**

- **Bag of Words:**
  - From training corpus extract vocabulary.
  - Represent each terms as 0 or 1
  - Highly sparse

- **Document-class joint feature representation:**
  - Inspired by learning to rank
  - Similarity features between an example and class of examples
  - For example:
    \[
    \sum_{t \in y \cap x} 1
    \]
  
  Where,
  - \(x \rightarrow\) One document
  - \(y \rightarrow\) Class of documents
Outline

1 Introduction

2 Multiclass to Binary Reduction

3 Double-Sampled Multiclass to Binary Reduction

4 Experimental Results

5 Conclusion
Motivation of our work

- Baselines: Model complexity increases with classes ($K$) and feature dimension ($d$).
- Algorithm that scales well for large scale data
- Does not suffer from class imbalance problem
- Less complex model
- Competitive with the state of the art approaches
Framework

- $\mathcal{X} \subseteq \mathbb{R}^d$ : Input Space
- $\mathcal{Y} = 1, \ldots, K$ : Output Space
- $S = (x_i^{y_i})_{i=1}^n$ : Training set of i.i.d. pairs
- $G = g : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ : Class of predictors

Instantaneous Loss

$$e(g, x^y) = \frac{1}{K - 1} \sum_{y' \in \mathcal{Y} \setminus y} \mathbb{1}_{g(x^y) \leq g(x^{y'})}$$  \hspace{1cm} (1)

- $\mathbb{1}_{\pi}$ is the indicator function (Value is 0 or 1)
- Average number of classes that get greater scoring by $g$ than true class
- Ranking loss used in Multiclass-SVM $^a$

$^a$Weston et. al. (1998)
Empirical Loss

Empirical error of \( g \in G \) over \( S \) is:

\[
L_m(g, S) = \frac{1}{m(K - 1)} \sum_{i=1}^{m} \sum_{y' \in Y \setminus y_i} \mathbb{1}_{g(x_{y_i}^{y_i}) \leq g(x_{y_i}^{y_i})}
\]

\[
= \frac{1}{m(K - 1)} \sum_{i=1}^{m} \sum_{y' \in Y \setminus y_i} \mathbb{1}_{\underbrace{h(x_{y_i}^{y_i}, x_{y_i}^{y_i}) \leq 0}_{g(x_{y_i}^{y_i}) - g(x_{y_i}^{y_i})}}
\]

- Resembles to binary-classification-loss based risk
- Selection of a hypothesis in \( G \) minimizing risk over \( S \) is equivalent to search a hypothesis in \( H \) minimizing risk over \( T(S) \) of size \( m \times (K - 1) \)
Multiclass to binary reduction example

We consider the following transformation

\[ T(S) = \begin{cases} 
(z_1 = (x_1^{y_1}, x_1^{y_2}), +1) & (z_2 = (x_1^{y_1}, x_1^{y_3}), +1) & (z_3 = (x_1^{y_1}, x_1^{y_4}), +1) \\
(z_4 = (x_2^{y_1}, x_2^{y_2}), -1) & (z_5 = (x_2^{y_2}, x_2^{y_3}), +1) & (z_6 = (x_2^{y_2}, x_2^{y_4}), +1) \\
(z_7 = (x_3^{y_1}, x_3^{y_3}), -1) & (z_8 = (x_3^{y_2}, x_3^{y_3}), -1) & (z_9 = (x_3^{y_3}, x_3^{y_4}), +1) \\
(z_{10} = (x_4^{y_1}, x_4^{y_4}), -1) & (z_{11} = (x_4^{y_2}, x_4^{y_4}), -1) & (z_{12} = (x_4^{y_3}, x_4^{y_4}), -1) 
\end{cases} \]

- We consider the following transformation

\[ T(S) = \left\{ \begin{cases} 
(z_j = (x_i^k, x_i^{y_j}), \tilde{y}_j = -1) & \text{if } k < y_i \\
(z_j = (x_i^{y_i}, x_i^k), \tilde{y}_j = +1) & \text{elsewhere} 
\end{cases} \right\}_{j = (i-1)(K-1)+k}^{(i-1)(K-1)+K}

- \(|T(S)| = m \times (K - 1)|
Multiclass to binary reduction algorithm

[Bikash et al. 2015]

**Input:** Labeled training set $S = (x_i^{y_i})_{i=1}^m$; A binary classifier $A$;

**Initialize**

$T(S) \leftarrow \emptyset$;

for $i = 1..m$ do

  for $k = 1..K$ do

    if $y_i > k$ then
      $T(S) \leftarrow \{(\Phi(x_i^{y_i}) - \Phi(x_i^k), +1)\}$
    end

    if $y_i < k$ then
      $T(S) \leftarrow \{(\Phi(x_i^k) - \Phi(x_i^{y_i}), -1)\}$
    end

  end

end

Learn $A$ on $T(S)$ to get learned weight vector $w$

**Testing:** For test example, $x^*$, estimate $\Phi(x'^y)$ for all $x'^y$ pairs and predicted class is the one which maximizes $\langle w, \Phi(x'^y) \rangle$

---

**Fig. 2.** Multiclass reduced to binary classification (mRb)
Improvements and New challenges

**Improvements:**
- One parameter vector for all classes.
- Low-dimensional feature space.
- Overcome class imbalance.

**New Challenges:**
- Number of transformations huge for larger K
- Large computational overhead
- Large memory requirement
Outline

1 Introduction

2 Multiclass to Binary Reduction

3 Double-Sampled Multiclass to Binary Reduction

4 Experimental Results

5 Conclusion
Aggressive double sampling

1. Drawing uniformly $\mu$ examples per class, in order to form practical set $S_\mu$;
   - Reduce redundancy in examples
   - Emphasizing rare classes

2. For each example $x^y$ in $S_\mu$, drawing uniformly $\kappa$ adversarial classes in $\mathcal{Y}\setminus\{y\}$.
   - Reduces time complexity
   - Low memory requirement
Double Sampled Multi to Binary Reduction

**Input:** Labeled training set \( S = (x_i^{y_i})_{i=1}^m \)

**Initialize**

\[ T_\kappa(S_\mu) \leftarrow \emptyset \]

\[ S_\mu \leftarrow \emptyset \]

**for** \( k = 1..K \) **do**

- Draw randomly a set \( S \) of \( \mu \) examples of class \( k \) from \( S \) \( \triangleright \mu \ll |S_k| \)

\[ S_\mu \leftarrow S_\mu \cup S \]

**end**

**forall the** \( x^y \in S_\mu \) **do**

- Draw uniformly a set \( \mathcal{K} \) of \( \kappa \) classes from \( \mathcal{Y} \setminus \{y\} \) \( \triangleright \kappa \ll |K| \)

**forall the** \( k \in \mathcal{K} \) **do**

  - if \( k < y \) **then**
    
    \[ T_\kappa(S_\mu) \leftarrow T_\kappa(S_\mu) \cup (z = (\phi(x^k), \phi(x^y)), \tilde{y} = -1) \]
    
  - **end**

  - else
    
    \[ T_\kappa(S_\mu) \leftarrow T_\kappa(S_\mu) \cup (z = (\phi(x^y), \phi(x^k)), \tilde{y} = +1) \]
    
  - **end**

**end**

**return** \( T_\kappa(S_\mu) \)
Outline

1. Introduction
2. Multiclass to Binary Reduction
3. Double-Sampled Multiclass to Binary Reduction
4. Experimental Results
5. Conclusion
Experimental Setup

Datasets:
- Application: Text Classification
- DMOZ and Wikipedia datasets. (LSHTC challenge)
- Pre-processed with stop word removal and stemming.
- Random samples of 1000, 2000, 3000, 4000, 5000, 7500, 10000, 20000.

Comparison:
- DS-m2b: Proposed double sampled multiclass to binary algorithm
- OVA: One-Vs-All algorithm
- M-SVM: Crammar-Singer implementation of multiclass SVM
- Recall Tree: Hierarchical One-Vs-Some algorithm
### Feature representation $\Phi(x^y)$

<table>
<thead>
<tr>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $\sum_{t \in y \cap x} \ln(1 + y_t)$</td>
</tr>
<tr>
<td>2. $\sum_{t \in y \cap x} \ln(1 + \frac{l_S}{S_t})$</td>
</tr>
<tr>
<td>3. $\sum_{t \in y \cap x} l_t$</td>
</tr>
<tr>
<td>4. $\sum_{t \in y \cap x} \ln(1 + \frac{y_t}{</td>
</tr>
<tr>
<td>5. $\sum_{t \in y \cap x} \ln(1 + \frac{y_t}{</td>
</tr>
<tr>
<td>6. $\sum_{t \in y \cap x} \ln(1 + \frac{y_t}{</td>
</tr>
<tr>
<td>7. $\sum_{t \in y \cap x} 1$</td>
</tr>
<tr>
<td>8. $\sum_{t \in y \cap x} \frac{y_t}{</td>
</tr>
<tr>
<td>9. BM25</td>
</tr>
<tr>
<td>10. $d(x^y, \text{centroid}(y))$</td>
</tr>
</tbody>
</table>

- $x_t$: number of occurrences of term $t$ in document $x$,
- $\mathcal{V}$: Number of distinct terms in $S$,
- $y_t = \sum_{x \in y} x_t$, $|y| = \sum_{t \in \mathcal{V}} y_t$, $S_t = \sum_{x \in S} x_t$, $l_S = \sum_{t \in \mathcal{V}} S_t$.
- $l_t$: idf of the term $t$,
Results: Runtime Comparison

<table>
<thead>
<tr>
<th># of classes</th>
<th>Total runtime (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10^1</td>
</tr>
<tr>
<td>1000</td>
<td>10^2</td>
</tr>
<tr>
<td>3000</td>
<td>10^3</td>
</tr>
<tr>
<td>5000</td>
<td>10^4</td>
</tr>
<tr>
<td>7500</td>
<td>10^5</td>
</tr>
<tr>
<td>10000</td>
<td>10^6</td>
</tr>
<tr>
<td>20000</td>
<td></td>
</tr>
</tbody>
</table>

- **OVA**
- **MSVM**
- **Recall Tree**
- **DS-m2b**
Results: Memory Comparison

![Graph showing memory usage comparison for different methods and class numbers. The x-axis represents the number of classes, ranging from 1000 to 20000, and the y-axis represents total memory usage in GB, with limits of 16GB and 32GB. Different methods are plotted: OVA, MSVM, Recall Tree, and DS-m2b. The graph illustrates how memory usage increases with the number of classes for each method.]
Results: Prediction Performance Comparison

![Graph showing prediction performance comparison with different algorithms.](image)
Outline

1. Introduction
2. Multiclass to Binary Reduction
3. Double-Sampled Multiclass to Binary Reduction
4. Experimental Results
5. Conclusion
Conclusion:

- Multiclass to binary reduction to handle large-class scenario and overcome class imbalance problem.
- Use of double sampling to further improve computational complexity and memory usage.
Questions?