Cost-aware Pre-training for Multiclass Cost-sensitive Deep Learning

Yu-An Chung\textsuperscript{1}  Hsuan-Tien Lin\textsuperscript{1}  Shao-Wen Yang\textsuperscript{2}

\textsuperscript{1} Dept. of Computer Science and Information Engineering
National Taiwan University, Taiwan

\textsuperscript{2} Intel Labs
Intel Corporation, USA

IJCAI 2016
Outline

1. Cost-sensitive Classification Setup
2. Estimate the costs - Regression Network
3. A novel Cost-aware Pre-training Technique
4. Conclusions
Outline

1. Cost-sensitive Classification Setup
2. Estimate the costs - Regression Network
3. A novel Cost-aware Pre-training Technique
4. Conclusions
What is the status of the patient?

- H1N1-infected
- Cold-infected
- Healthy

- A classification problem
  - grouping patients into different status.

Which mistake is more serious? Predicting ...

- H1N1 as Healthy vs. Cold as Healthy
Cost-sensitive Classification

Measuring the Mis-classification Costs by **Cost Matrix**

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th>H1N1</th>
<th>Cold</th>
<th>Healthy</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1N1</td>
<td>0</td>
<td>1000</td>
<td>100000</td>
<td></td>
</tr>
<tr>
<td>Cold</td>
<td>100</td>
<td>0</td>
<td>3000</td>
<td></td>
</tr>
<tr>
<td>Healthy</td>
<td>100</td>
<td>30</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- \( C(i, j) \): cost of classifying a class \( i \) example as class \( j \)
- Regular classification: special case of cost-sensitive classification

Cost-sensitive Classification Setup

- **Input:** A training set \( S = \{(x_n, y_n)\}_{n=1}^{N} \) and a **cost matrix** \( C \), where \( x_n \in \mathcal{X} \), \( y_n \in \mathcal{Y} = \{1, 2, ..., K\} \)
- **Goal:** Use \( S \) and \( C \) to train a classifier \( g : \mathcal{X} \to \mathcal{Y} \) such that the expected cost \( C(y, g(x)) \) on test example \( (x, y) \) is minimal
## Our Contributions

**Where are we?**

<table>
<thead>
<tr>
<th>Shallow Models (e.g., SVM)</th>
<th>Deep Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular (Cost-insensitive) Classification</td>
<td>Well-studied</td>
</tr>
<tr>
<td>Cost-sensitive Classification</td>
<td>Well-studied</td>
</tr>
</tbody>
</table>

- First work that studies Cost-sensitive Deep Learning
  1. a novel Cost-sensitive Loss (**CSL**) for training any deep models (**end-to-end**)
  2. a Cost-sensitive Autoencoder (**CAE**) equipped with **CSL** for pre-training deep models (**layer-wise**)
  3. a combination of 1) and 2) as a complete Cost-sensitive Deep Neural Network (**CSDNN**) solution
  4. extensive experimental results have shown that deep models indeed outperformed shallow ones (potential to study more!)
Outline

1. Cost-sensitive Classification Setup

2. Estimate the costs - Regression Network

3. A novel Cost-aware Pre-training Technique

4. Conclusions
Regression Network

Network: to estimate the per-class costs

Training:
- motivated by an earlier cost-sensitive SVM work, a **Cost-sensitive Loss (CSL)** that trains the network cost-sensitively is derived in this work (see paper or poster for details)

Prediction:  
\[
g(x) \equiv \arg\min_{1 \leq k \leq K} r_k(x)
\]
Outline

1. Cost-sensitive Classification Setup
2. Estimate the costs - Regression Network
3. A novel Cost-aware Pre-training Technique
4. Conclusions
Recap on Unsupervised Pre-training

A classical way of training DNNs

- Two steps
  - Unsupervised layer-wise pre-training
    - Autoencoder, Restricted Boltzmann Machine (RBM)
    - Several Autoencoders or RBMs can then be stacked to form a DNN.
  - End-to-end supervised fine-tuning

Cost-aware Pre-training

- Embed the proposed Cost-sensitive Loss (CSL) into Autoencoder
  - a cost-sensitive version of Autoencoder (CAE)
  - conduct cost-related features extraction
Autoencoder (AE):

Let $L_{CE}$ denotes the reconstruction errors of the AE to be minimized (CE stands for cross-entropy).
A novel Cost-aware Pre-training Technique

Cost-sensitive Autoencoder (CAE) for cost-aware pre-training

Cost-sensitive Autoencoder (CAE):

- **Objective function for CAE:**
  \[(1 - \beta) \times L_{CE} + \beta \times L_{CSL} \]
  \[\beta \in [0, 1]\]

- **When** \(\beta = 0\), **CAE \equiv AE**

- **CAE:** Reconstruct \(x\) and estimate \(C\) simultaneously
Experimental Results (Selected)

3 methods were compared to show the validity of CSL and CAE:

<table>
<thead>
<tr>
<th>Method</th>
<th>Cost-sensitive pre-training?</th>
<th>Cost-sensitive training?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DNN + CSL</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>CSDNN</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

![Average Test Costs on 8 Datasets (with DNN linearly scaled to 1)](chart.png)
Outline

1. Cost-sensitive Classification Setup
2. Estimate the costs - Regression Network
3. A novel Cost-aware Pre-training Technique
4. Conclusions
Conclusions

- CSL: make any deep model cost-sensitive (see paper for details)

- CSDNN = CAE pre-training + CSL fine-tuning: both techniques lead to significant improvements

- Extensive experimental results showed the superiority of CSDNN (see paper or poster)
Thank you!
Supplementary Materials

$\beta$ vs. Test Costs

- **MNIST**$\_\text{imb}$
  - $x$-axis: $0$ to $1$
  - $y$-axis: $0.16$ to $0.24$

- **bg–img–rot**$\_\text{imb}$
  - $x$-axis: $0$ to $1$
  - $y$-axis: $4$ to $5.2$

- **SVHN**$\_\text{imb}$
  - $x$-axis: $0$ to $1$
  - $y$-axis: $0.24$ to $0.34$

- **CIFAR–10**$\_\text{imb}$
  - $x$-axis: $0$ to $1$
  - $y$-axis: $6.4$ to $7.6$