Reduction from Cost-sensitive Multiclass Classification to One-versus-one Binary Classification

Hsuan-Tien Lin

Department of Computer Science and Information Engineering, National Taiwan University
(initiated and partly done at Caltech)

ACML 2014, 11/28/2014
Which Digit Did You Write?

- a **classification** problem
  — grouping “pictures” into different “categories”

**How to evaluate the classification performance?**
Cost-sensitive Classification

Mis-prediction Costs

- ZIP code recognition (regular classification):
  1: wrong; 2: right; 3: wrong; 4: wrong
  —only right or wrong

- Check value recognition (cost-sensitive classification):
  1: one-dollar mistake; 2: no mistake;
  3: one-dollar mistake; 4: two-dollar mistake
  —different costs for different mis-predictions

Cost-sensitive classification: embed application needs
Cost-sensitive Classification

Cost Vector

cost vector \( \mathbf{c} \): a row of cost components

- absolute cost for digit 2: \( \mathbf{c} = (1, 0, 1, 2) \)
- \textit{interval-insensitive} cost for previous presentation (\textit{interval insensitive loss for ordinal classification}): \( \mathbf{c} = (1, 0, 0, 0, 2, 3) \)
- “regular” classification cost for label 2: \( \mathbf{c}_{c}^{(2)} = (1, 0, 1, 1) \)

regular classification: 
\textbf{special case} of cost-sensitive classification
Cost-sensitive Classification Setup

**Given**

Given $N$ examples, each (input $x_n$, label $y_n$, cost $c_n) \in \mathcal{X} \times \{1, 2, \ldots, K\} \times \mathbb{R}^K$

- $K = 2$: binary; $K > 2$: **multiclass**
- will assume $c_n[y_n] = 0 = \min_{1 \leq k \leq K} c_n[k]$

**Goal**

A classifier $g(x)$ that pays a small cost $c[g(x)]$ on future **unseen** example $(x, y, c)$

- will assume $c[y] = 0 = c_{\text{min}} = \min_{1 \leq k \leq K} c[k]$
- note: $y$ not really needed in evaluation

**cost-sensitive classification:**

*can express any finite-loss supervised learning tasks*
## Our Contribution

<table>
<thead>
<tr>
<th></th>
<th>Binary</th>
<th>Multiclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>Well-studied</td>
<td>Well-studied</td>
</tr>
<tr>
<td>Cost-sensitive</td>
<td>Known (Zadrozny, 2003)</td>
<td><strong>Ongoing</strong> (our work, among others)</td>
</tr>
</tbody>
</table>

A theoretical and algorithmic study of cost-sensitive classification, which...

- introduces a methodology for extending regular classification algorithms to cost-sensitive ones with **any cost**
- provides **strong theoretical support** for the methodology
- leads to a simple algorithm with **promising experimental results**

will describe the methodology and a concrete algorithm
Cost-sensitive Binary Classification (1/2)

- patient status (?)
- H1N1 (1)
- NOH1N1 (2)
- predicting H1N1 as NOH1N1: serious to public health
- predicting NOH1N1 as H1N1: not good, but less serious
- cost-sensitive matrix (each row as a vector): 
  \[
  \begin{pmatrix}
  c_1 \\
  c_2 
  \end{pmatrix} = \begin{pmatrix}
  0 & 1000 \\
  1 & 0
  \end{pmatrix};
  \]
- regular evaluation matrix \( C_c \):
  \[
  \begin{pmatrix}
  0 & 1 \\
  1 & 0
  \end{pmatrix}
  \]

how to change the entry from 1 to 1000?
Cost-sensitive Binary Classification (2/2)

**copy each case labeled H1N1 1000 times**

**original problem**

evaluate w/ \[
\begin{pmatrix}
0 & 1000 \\
1 & 0 \\
\end{pmatrix}
\]

- \((x_1, \text{H1N1})\)
- \((x_2, \text{NOH1N1})\)
- \((x_3, \text{NOH1N1})\)
- \((x_4, \text{NOH1N1})\)
- \((x_5, \text{H1N1})\)

**equivalent problem**

evaluate w/ \[
\begin{pmatrix}
0 & 1 \\
1 & 0 \\
\end{pmatrix}
\]

- \((x_1, \text{H1N1}), \ldots, (x_1, \text{H1N1})\)
- \((x_2, \text{NOH1N1})\)
- \((x_3, \text{NOH1N1})\)
- \((x_4, \text{NOH1N1})\)
- \((x_5, \text{H1N1}), \ldots, (x_5, \text{H1N1})\)

mathematically:

\[
\begin{pmatrix}
0 & 1000 \\
1 & 0 \\
\end{pmatrix} = \begin{pmatrix}
1000 & 0 \\
0 & 1 \\
\end{pmatrix} \cdot \begin{pmatrix}
0 & 1 \\
1 & 0 \\
\end{pmatrix}
\]
Key Idea: Cost Transformation

\[
\begin{pmatrix}
0 & 1000 \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
c_1 \\ c_2
\end{pmatrix} =
\begin{pmatrix}
1000 & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
0 & 1 \\
1 & 0
\end{pmatrix}
\begin{pmatrix}
c_1 \\ c_2
\end{pmatrix}
\]

\# of copies

\[
\begin{pmatrix}
3 & 2 & 3 & 4
\end{pmatrix} =
\begin{pmatrix}
1 & 2 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0
\end{pmatrix}
\]
mixture weights \( q \)

- **split** the cost-sensitive example:
  \((x, 2) \implies \) a weighted mixture \{ \((x, 1, 1), (x, 2, 2), (x, 3, 1)\) \}

- **cost equivalence**: for any classifier \( g \),

\[
c[g(x)] = \sum_{\ell=1}^{K} q[\ell] [\ell \neq g(x)]
\]

\[
\min_g \text{ expected LHS} = \min_g \text{ expected RHS}
\]

(original cost-sensitive problem)

(a regular problem when \( q[\ell] \geq 0 \))
Cost Transformation Methodology: Preliminary

1. split each training example \((x_n, y_n, c_n)\) to a weighted mixture
   \[
   \left\{(x_n, \ell, q_n[\ell])\right\}_{\ell=1}^K
   \]

2. apply regular classification algorithm on the weighted mixtures
   \[
   \bigcup_{n=1}^N \left\{(x_n, \ell, q_n[\ell])\right\}_{\ell=1}^K
   \]

- by cost equivalence,
  
  \[\text{good } g \text{ for new regular classification problem} = \text{good } g \text{ for original cost-sensitive classification problem}\]

- regular classification: needs \(q[\ell] \geq 0\)

**but what if \(q[\ell]\) negative?**
Cost Transformation Methodology

Similar Cost Vectors

\[
\begin{pmatrix}
1 & 0 & 1 & 2 \\
3 & 2 & 3 & 4 \\
\end{pmatrix}
\]

\[
\begin{pmatrix}
1/3 & 4/3 & 1/3 & -2/3 \\
1 & 2 & 1 & 0 \\
\end{pmatrix}
\cdot
\begin{pmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
\end{pmatrix}
\]

- negative \( q[\ell] \): cannot split
- but \( c = (1, 0, 1, 2) \) is similar to \( \hat{c} = (3, 2, 3, 4) \):
  for any classifier \( g \),

\[
c[g(x)] + \text{constant} = \hat{c}[g(x)]
\]

- constant can be dropped during minimization

\[
\begin{align*}
\min_g \text{expected } c &= \text{(original cost-sensitive problem)} \\
= \min_g \text{expected } \hat{c} &= \text{(shifted cost-sensitive problem)} \\
= \min_g \text{expected RHS} &= \text{(regular problem w/ } q[\ell] \geq 0) \\
\end{align*}
\]
Cost Transformation Methodology

Cost Transformation Methodology: Revised

1. (minimum-)shift each cost $c$ to a similar and "splittable" $\hat{c}$

2. split each training example $(x_n, y_n, \hat{c}_n)$ to a weighted mixture
   \[
   \{(x_n, \ell, q_n[\ell])\}_{\ell=1}^K
   \]

3. apply regular classification algorithm on the weighted mixtures
   \[
   \bigcup_{n=1}^N \{(x_n, \ell, q_n[\ell])\}_{\ell=1}^K
   \]

- **splittable**: $q_n[\ell] \geq 0$
- **minimum**: see paper

next: **OVO** to find good $g$ for new regular classification problem
Cost-sensitive One-Versus-One

From OVO to CSOVO

### One-Versus-One: A Popular Classification Meta-Method

1. for a pair \((i, j)\), take all examples \((x_n, y_n)\) that \(y_n = i\) or \(j\)
2. train a binary classifier \(g^{(i,j)}\) using those examples
3. repeat the previous two steps for all different \((i, j)\)
4. predict using the votes from \(g^{(i,j)}\)

---

**cost-sensitive one-versus-one:**

- cost transformation + one-versus-one
Cost-sensitive One-Versus-One (CSOVO)

1. for a pair \((i, j)\), transform all examples \((x_n, y_n, c_n)\) to 
\[
\left(x_n, \arg\min_{k \in \{i, j\}} c_n[k]\right)
\] with weight \(|c_n[i] - c_n[j]|\)

2. train a binary classifier \(g^{(i,j)}\) using those examples

3. repeat the previous two steps for all different \((i, j)\)

4. predict using the votes from \(g^{(i,j)}\)

- comes with good theoretical guarantee:

\[
test\ cost\ of\ final\ classifier \leq 2 \sum_{i<j} test\ cost\ of\ g^{(i,j)}
\]

**simple, efficient**, and takes original OVO as **special case**
CSOVO v.s. WAP

- a general cost-sensitive setup with “random” cost
- WAP (Abe et al., 2004): related to CSOVO, but a bit more complicated
- couple both meta-methods with SVM

CSOVO simpler with similar performance — a preferable choice
Experimental Performance

CSOVO v.s. Others

- other meta-methods to binary classification: tree-based (FT, TREE) and error-correcting-code (SECOC)
- couple all meta-methods with SVM

CSOVO often among the best
Conclusion

- **cost transformation** methodology: makes any (robust) regular classification algorithm cost-sensitive.
- Theoretical guarantee: **cost equivalence**
- Algorithmic use: a **novel and simple** algorithm CSOVO
- Experimental performance of CSOVO: **promising**

Thank you! Questions?