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?

?

one (1) two (2) three (3) four (4)

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

how to evaluate the **classification performance**?

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 Vector

cost vector **c**: a row of cost components

- absolute cost for digit 2: $\mathbf{c} = (1, 0, 1, 2)$
- **interval-insensitive** cost for **previous presentation** (*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:
special case of cost-sensitive classification

Cost-sensitive Classification Setup

Given

N examples, each
(input $\mathbf{x}_n$, label y_n, cost $\mathbf{c}_n$) $\in \mathcal{X} \times \{1, 2, \ldots, K\} \times R^K$

- $K = 2$: binary; $K > 2$: **multiclass**
- will assume $\mathbf{c}_n[y_n] = 0 = \min_{1 \le k \le K} \mathbf{c}_n[k]$

Goal

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

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

cost-sensitive classification:
can express any finite-loss supervised learning tasks

Our Contribution

	binary	multiclass
regular	well-studied	well-studied
cost-sensitive	known (Zadrozny, 2003)	ongoing (our work, among others)

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): $\begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix}$;

  regular evaluation matrix $\mathcal{C}_c$: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

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/ $\begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix}$

$(\mathbf{x}_1, \text{H1N1})$
$(\mathbf{x}_2, \text{NOH1N1})$
$(\mathbf{x}_3, \text{NOH1N1})$
$(\mathbf{x}_4, \text{NOH1N1})$
$(\mathbf{x}_5, \text{H1N1})$

equivalent problem

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

$(\mathbf{x}_1, \text{H1N1}), \cdots, (\mathbf{x}_1, \text{H1N1})$
$(\mathbf{x}_2, \text{NOH1N1})$
$(\mathbf{x}_3, \text{NOH1N1})$
$(\mathbf{x}_4, \text{NOH1N1})$
$(\mathbf{x}_5, \text{H1N1}), \cdots, (\mathbf{x}_5, \text{H1N1})$

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

$$\underbrace{\begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix}}_{\begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{pmatrix}} = \underbrace{\begin{pmatrix} 1000 & 0 \\ 0 & 1 \end{pmatrix}}_{\#\text{ of copies}} \cdot \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{\mathcal{C}_c}$$

$$\underbrace{\begin{pmatrix} 3 & 2 & 3 & 4 \end{pmatrix}}_{\mathbf{c}} = \underbrace{\begin{pmatrix} 1 & 2 & 1 & 0 \end{pmatrix}}_{\text{mixture weights } \mathbf{q}} \cdot \underbrace{\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}}_{\mathcal{C}_c}$$

- **split** the cost-sensitive example:

  $(\mathbf{x}, 2) \Longrightarrow$ a weighted mixture $\{(\mathbf{x}, 1, 1), (\mathbf{x}, 2, 2), (\mathbf{x}, 3, 1)\}$
- cost equivalence: for any classifier g,

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

	$\min_g$ expected LHS	(original cost-sensitive problem)
$=$	$\min_g$ expected RHS	(a regular problem when $\mathbf{q}[\ell] \geq 0$)

Cost Transformation Methodology: Preliminary

1. split each training example $(\mathbf{x}_n, y_n, \mathbf{c}_n)$ to a weighted mixture $\{(\mathbf{x}_n, \ell, \mathbf{q}_n[\ell])\}_{\ell=1}^{K}$
2. apply regular classification algorithm on the weighted mixtures $\bigcup_{n=1}^{N} \{(\mathbf{x}_n, \ell, \mathbf{q}_n[\ell])\}_{\ell=1}^{K}$

- by cost equivalence,

  good g for new regular classification problem

  $=$ good g for original cost-sensitive classification problem
- regular classification: needs $\mathbf{q}[\ell] \geq 0$

but what if $\mathbf{q}[\ell]$ negative?

Similar Cost Vectors

$$\underbrace{\begin{pmatrix} 1 & 0 & 1 & 2 \\ 3 & 2 & 3 & 4 \end{pmatrix}}_{\text{costs}} = \underbrace{\begin{pmatrix} 1/3 & 4/3 & 1/3 & -2/3 \\ 1 & 2 & 1 & 0 \end{pmatrix}}_{\text{mixture weights } \mathbf{q}} \cdot \underbrace{\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}}_{\text{classification costs}}$$

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

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

- constant can be dropped during minimization

	$\min_g$ expected $\mathbf{c}$	(original cost-sensitive problem)
$=$	$\min_g$ expected $\hat{\mathbf{c}}$	(shifted cost-sensitive problem)
$=$	$\min_g$ expected RHS	(regular problem w/ $\mathbf{q}[\ell] \geq 0$)

Cost Transformation Methodology: Revised

1. (minimum-)shift each cost $\mathbf{c}$ to a similar and "splittable" $\hat{\mathbf{c}}$
2. split each training example $(\mathbf{x}_n, y_n, \hat{\mathbf{c}}_n)$ to a weighted mixture $\{(\mathbf{x}_n, \ell, \mathbf{q}_n[\ell])\}_{\ell=1}^{K}$
3. apply regular classification algorithm on the weighted mixtures $\bigcup_{n=1}^{N} \{(\mathbf{x}_n, \ell, \mathbf{q}_n[\ell])\}_{\ell=1}^{K}$

- **splittable**: $\mathbf{q}_n[\ell] \geq 0$
- **minimum**: *see paper*

next: **OVO** to find good g for new regular classification problem

From OVO to CSOVO

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

1. for a pair (i, j), take all examples $(\mathbf{x}_n, y_n)$ 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)

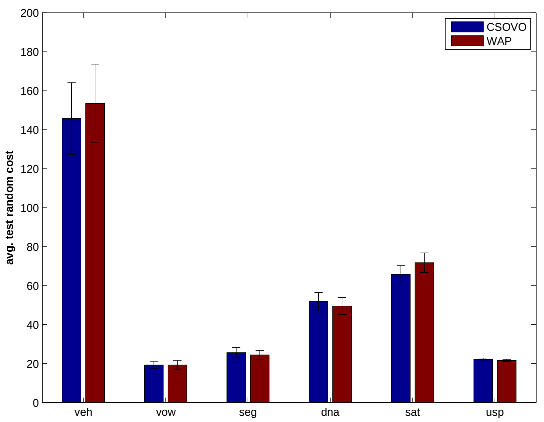
1. for a pair (i, j), transform all examples $(\mathbf{x}_n, y_n, \mathbf{c}_n)$ to
$$\left(\mathbf{x}_n, \operatorname*{argmin}_{k \in \{i,j\}} \mathbf{c}_n[k]\right) \text{ with weight } \left|\mathbf{c}_n[i] - \mathbf{c}_n[j]\right|$$
2. train a binary classifier $g^{(i,j)}$ using those examples
3. repeat the previous two steps for all different (i, j)
4. predict using the votes from $g^{(i,j)}$

- comes with **good theoretical guarantee**:

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

simple, **efficient**, and takes original OVO as **special case**

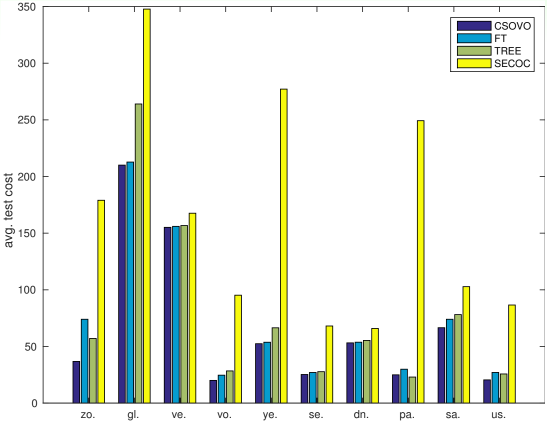
CSOVO v.s. WAP



- 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

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?