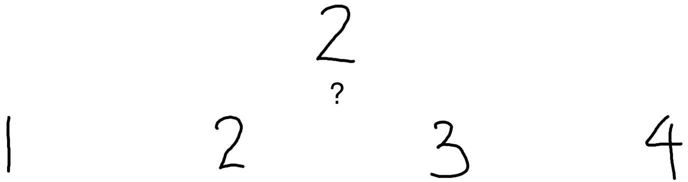
A Simple Algorithm for Cost-Sensitive Classification

Hsuan-Tien Lin

Dept. of CSIE, NTU

Department Seminar Talk, 09/19/2008

Which Digit Did You Write?

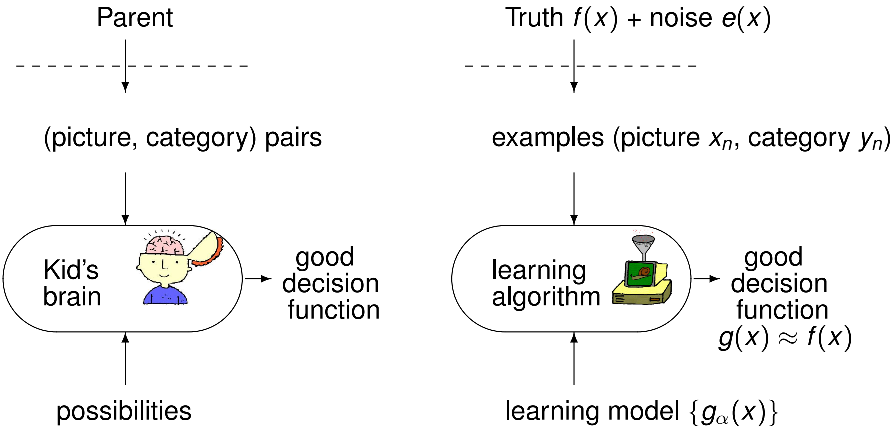


one (1) two (2) three (3) four (4)

- a **classification** problem
  —grouping "pictures" into different "categories"

How can machines learn to classify?

Supervised Machine Learning



challenge:

see only $\{(x_n, y_n)\}$ without knowing $f(x)$ or $e(x)$

$\overset{?}{\Longrightarrow}$ **generalize** to unseen (x, y) w.r.t. $f(x)$

Mis-prediction Costs ($g(x) \approx f(x)$?)

?

- ZIP code recognition:
  1: wrong; 2: right; 3: wrong; 4: wrong
- check value recognition:
  1: one-dollar mistake; 2: no mistake;
  3: one-dollar mistake; 4: **two**-dollar mistake
- evaluation by formation similarity:
  1: not very similar; 2: very similar;
  3: somewhat similar; 4: a **silly** prediction

different applications evaluate mis-predictions differently

ZIP Code Recognition

?

1: wrong; 2: right; 3: wrong; 4: right

- **regular** classification problem: only right or wrong
- wrong cost: 1; right cost: 0
- prediction error of g on some (x, y):

$$\text{classification cost} = [\![y \neq g(x)]\!]$$

regular classification: **well-studied**, many good algorithms

Check Value Recognition

?

1: one-dollar mistake; 2: no mistake;
3: one-dollar mistake; 4: **two**-dollar mistake

- **cost-sensitive** classification problem:
  different costs for different mis-predictions
- prediction error of g on some (x, y):

$$\text{absolute cost} = |y - g(x)|$$

cost-sensitive classification: **new**, need more research

Which Age-Group?

?

infant (1)

child (2)

teen (3)

adult (4)

- small mistake—classify a child as a teen; big mistake—classify an infant as an adult
- prediction error of g on some (x, y):

$$\mathcal{C}(y, g(x)), \text{ where } \mathcal{C} = \begin{pmatrix} 0 & 1 & 4 & 5 \\ 1 & 0 & 1 & 3 \\ 3 & 1 & 0 & 2 \\ 5 & 4 & 1 & 0 \end{pmatrix}$$

$\mathcal{C}$: **cost matrix**

Cost Matrix $\mathcal{C}$

regular classification

$\mathcal{C}$ = classification cost $\mathcal{C}_c$:

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

cost-sensitive classification

$\mathcal{C}$ = anything other than $\mathcal{C}_c$:

$$\begin{pmatrix} 0 & 1 & 4 & 5 \\ 1 & 0 & 1 & 3 \\ 3 & 1 & 0 & 2 \\ 5 & 4 & 1 & 0 \end{pmatrix}$$

regular classification:
special case of cost-sensitive classification

Cost-Sensitive Binary Classification (1/2)

medical profile x

?

medical profile x_1
SARS (1)

medical profile x_2
NOSARS (2)

- predicting SARS as NOSARS:
  serious consequences to public health
- predicting NOSARS as SARS:
  not good, but less serious
- cost-sensitive $\mathcal{C}$: $\begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix}$
- regular $\mathcal{C}_c$: $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

how to change the entry from 1 to 1000?

Cost-Sensitive Binary Classification (2/2)

copy each case labeled SARS 1000 times

original problem

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

(x_1,SARS)
(x_2,NOSARS)
(x_3,NOSARS)
(x_4,NOSARS)
(x_5,SARS)

equivalent problem

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

(x_1,SARS), $\cdots$, (x_1,SARS)
(x_2,NOSARS)
(x_3,NOSARS)
(x_4,NOSARS)
(x_5,SARS), $\cdots$, (x_5,SARS)

mathematically:

$$\begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1000 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

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 some promising algorithms with **superior experimental results**

will describe the methodology and a concrete algorithm

Key Idea: Cost Transformation

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

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

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

  $(x, 2)$

  $\Longrightarrow$ a mixture of regular examples $\{(x,1),(x,2),(x,2),(x,3)\}$

  or a weighted mixture $\{(x,1,1),(x,2,2),(x,3,1)\}$

why split?

Cost Equivalence by Splitting

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

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

$$\mathcal{C}(y, g(x)) = \sum_{\ell=1}^{K} Q(y, \ell) \, [\![\ell \neq g(x)]\!]$$

	$\min_g$ expected LHS	(original cost-sensitive problem)
$=$	$\min_g$ expected RHS	(a regular problem when $Q(y, \ell) \geq 0$)

Cost Transformation Methodology: Preliminary

1. split each training example (x_n, y_n) to a weighted mixture $\left\{(x_n, \ell, Q(y_n, \ell))\right\}_{\ell=1}^{K}$
2. apply regular classification algorithm on the weighted mixtures $\bigcup_{n=1}^{N} \left\{(x_n, \ell, Q(y_n, \ell))\right\}_{\ell=1}^{K}$

- by cost equivalence,

  good g for new regular classification problem
  $=$ good g for original cost-sensitive classification problem

- regular classification: needs $Q(y_n, \ell) \geq 0$

but what if $Q(y_n, \ell)$ negative?

Similar Cost Vectors

$$\underbrace{\begin{pmatrix} 1 & 0 & 1 & 2 \\ 3 & 2 & 3 & 4 \end{pmatrix}}_{\text{costs}} = \underbrace{\begin{pmatrix} 1/3 & 4/3 & 1/3 & -2/3 \\ 1 & 2 & 1 & 0 \end{pmatrix}}_{\text{mixture weights } Q(y,\ell)} \cdot \underbrace{\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}}_{\text{classification costs}}$$

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

$$\hat{\mathbf{c}}[g(x)] + \text{constant} = \mathbf{c}[g(x)] = \sum_{\ell=1}^{K} Q(y, \ell) \, [\![\ell \neq g(x)]\!]$$

- constant can be dropped during minimization

	$\min_g$ expected $\hat{\mathcal{C}}(y, g(x))$	(original cost-sensitive problem)
$=$	$\min_g$ expected RHS	(regular problem w/ $Q \geq 0$)

Cost Transformation Methodology: Revised

1. shift each row of original cost $\hat{\mathcal{C}}$ to a similar and "splittable" $\mathcal{C}(y, :)$
2. split (x_n, y_n) to a weighted mixture $\left\{(x_n, \ell, Q(y_n, \ell))\right\}_{\ell=1}^{K}$ with $\mathcal{C}$
3. apply regular classification algorithm on the weighted mixtures $\bigcup_{n=1}^{N} \left\{(x_n, \ell, Q(y_n, \ell))\right\}_{\ell=1}^{K}$

- **splittable**: $Q(y_n, \ell) \geq 0$
- by cost equivalence after shifting:

$$\begin{aligned} & \text{good } g \text{ for new regular classification problem} \\ = \quad & \text{good } g \text{ for original cost-sensitive classification problem} \end{aligned}$$

but infinitely many similar and splittable $\mathcal{C}$!

Uncertainty in Mixture

- a single example $\{(x, 2)\}$
  —**certain** that the desired label is 2
- a mixture $\{(x, 1, 1), (x, 2, 2), (x, 3, 1)\}$ sharing the same x
  —**uncertainty** in the desired label (25%: 1, 50%: 2, 25%: 3)
- over-shifting adds unnecessary mixture uncertainty:

$$\underbrace{\begin{pmatrix} 3 & 2 & 3 & 4 \\ 33 & 32 & 33 & 34 \end{pmatrix}}_{\text{costs}} = \underbrace{\begin{pmatrix} 1 & 2 & 1 & 0 \\ 11 & 12 & 11 & 10 \end{pmatrix}}_{\text{mixture weights}} \cdot \underbrace{\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}}_{\mathcal{C}_c}$$

should choose a similar and splittable **c**
with **minimum mixture uncertainty**

Cost Transformation Methodology: Final

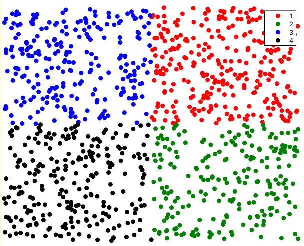
1. shift original cost $\hat{\mathcal{C}}$ to a similar and splittable $\mathcal{C}$ with minimum "mixture uncertainty"
2. split (x_n, y_n) to a weighted mixture $\left\{(x_n, \ell, Q(y_n, \ell))\right\}_{\ell=1}^{K}$ with $\mathcal{C}$
3. apply regular classification algorithm on the weighted mixtures $\bigcup_{n=1}^{N} \left\{(x_n, \ell, Q(y_n, \ell))\right\}_{\ell=1}^{K}$

- mixture uncertainty: entropy of each normalized $Q(y, :)$
- a simple and unique optimal shifting exists for every $\hat{\mathcal{C}}$

good g for new regular classification problem
$=$ good g for original cost-sensitive classification problem

Unavoidable (Minimum) Uncertainty

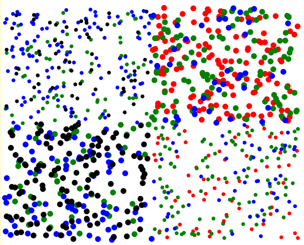
Original Cost-Sensitive Classification Problem



individual examples with certainty

+ absolute cost =

New Regular Classification Problem



mixtures with unavoidable uncertainty

- new problem usually **harder** than original one

need robust regular classification algorithm to deal with uncertainty

Making a Classification Algorithm Robust

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 **(original one-versus-one)**
2. for a pair (i, j), from each weighted mixture $\{(x_n, \ell, q_\ell)\}$ with $q_i > q_j$, keep (x_n, i) with weight $(q_i - q_j)$; vice versa **(robust one-versus-one)**
3. train a binary classifier $g^{(i,j)}$ using those examples
4. repeat the previous two steps for all different (i, j)
5. predict using the votes from $g^{(i,j)}$

- un-shifting inside the meta-method to remove uncertainty
- robust step makes it suitable for cost transformation methodology

cost-sensitive one-versus-one:
cost transformation + robust one-versus-one

Cost-Sensitive One-Versus-One (CSOVO)

1. for a pair (i, j), transform all examples (x_n, y_n) to $\left(x_n, \operatorname*{argmin}_{k \in \{i,j\}} \mathcal{C}(y_n, k)\right)$ with weight $\left|\mathcal{C}(y_n, i) - \mathcal{C}(y_n, j)\right|$
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**:

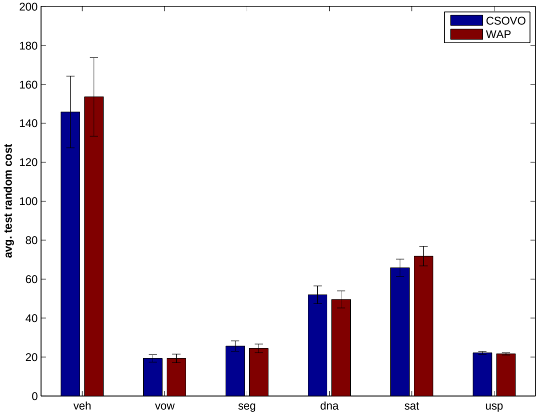
$$\text{test cost of final classifier} \leq 2 \sum_{i<j} \text{test cost of } g^{(i,j)}$$

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

another success:

cost-sensitive one-versus-all (Lin, 2008)

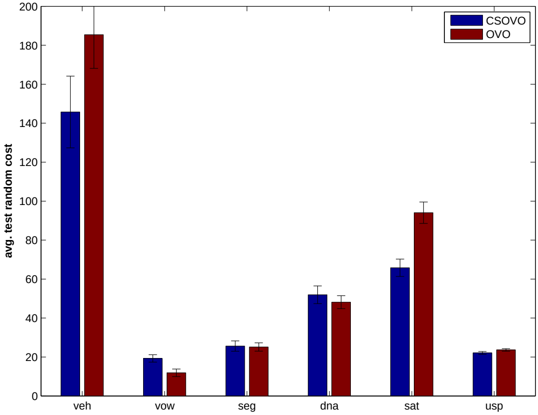
CSOVO v.s. WAP



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

CSOVO simpler, faster, with similar performance —a preferable choice

CSOVO v.s. OVO



- OVO: popular regular classification meta-method, NOT cost-sensitive
- couple both meta-methods with SVM

CSOVO often better suited for cost-sensitive classification

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: **superior**

many more cost-sensitive algorithms can be designed similarly