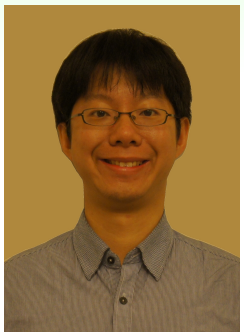
Cost-Sensitive Classification: Algorithm and Application

Hsuan-Tien Lin
htlin@csie.ntu.edu.tw

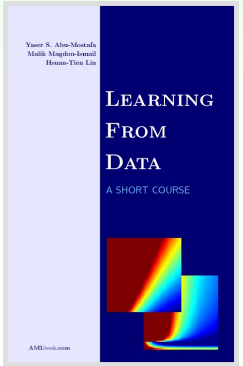
Department of Computer Science
& Information Engineering
National Taiwan University

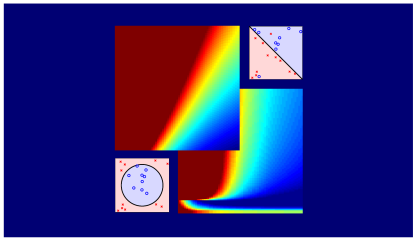
AI in Data Science Forum
December 20, 2017

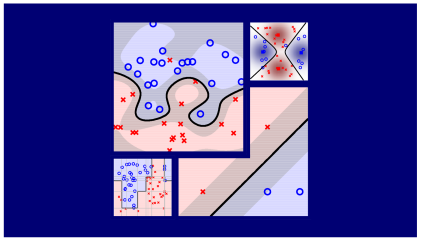
About Me



- Chief Data Scientist, Appier
- Professor, Dept. of CSIE, NTU
- Co-author of textbook "*Learning from Data: A Short Course*"
- Instructor of two NTU-Coursera Mandarin-teaching ML Massive Open Online Courses







goal: make machine learning **more realistic**

- multi-class cost-sensitive classification : in ICML '10, BIBM '11, KDD '12, ACML '14, IJCAI '16, etc.
- multi-label classification: in ACML '11, NIPS '12, ICML '14, AAAI '18, etc.
- online/active learning: in ICML '12, ACML '12, ICML '14, AAAI '15, etc.
- large-scale data mining (w/ Profs. S.-D. Lin & C.-J. Lin & students): KDDCup world champions of '10, '11 ($\times$2), '12, '13 ($\times$2)

Which Digit Did You Write?

?

one (1)

two (2)

three (3)

four (4)

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

C'mon, we know about multiclass classification all too well! :-)

Performance Evaluation ($g(\mathbf{x}) \approx f(\mathbf{x})$?)

?

- 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

different applications:
evaluate mis-predictions differently

ZIP Code Recognition

?

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

- **regular** multiclass classification: only right or wrong
- wrong cost: 1; right cost: 0
- prediction error of h on some $(\mathbf{x}, y)$:

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

regular multiclass classification:
well-studied, many good algorithms

Check Value Recognition

?

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

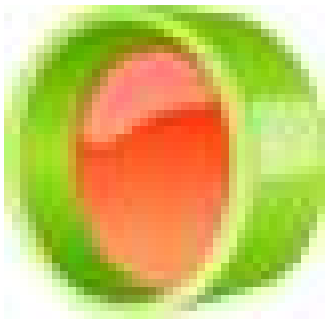
- **cost-sensitive** multiclass classification:
  different costs for different mis-predictions
- e.g. prediction error of h on some $(\mathbf{x}, y)$:

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

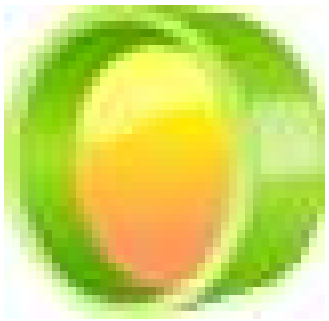
cost-sensitive multiclass classification:
relatively newer, need more research

What is the Status of the Patient?

?



H7N9-infected



cold-infected

healthy

- another **classification** problem
  —grouping "patients" into different "status"

are all mis-prediction costs equal?

Patient Status Prediction

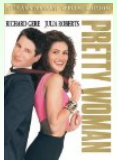
error measure = society cost

actual \ predicted	H7N9	cold	healthy
H7N9	0	1000	**100000**
cold	100	0	3000
healthy	100	30	0

- H7N9 mis-predicted as healthy: **very high cost**
- cold mis-predicted as healthy: high cost
- cold correctly predicted as cold: no cost

human doctors consider costs of decision;
can computer-aided diagnosis do the same?

What is the Type of the Movie?



?

romance

fiction

terror

customer 1 who hates romance but likes terror

error measure = non-satisfaction

actual \ predicted	romance	fiction	terror
romance	0	5	100

customer 2 who likes terror and romance

actual \ predicted	romance	fiction	terror
romance	0	5	3

different customers:

evaluate mis-predictions differently

Cost-Sensitive Multiclass Classification Tasks

movie classification with non-satisfaction

actual \ predicted	romance	fiction	terror
customer 1, romance	0	5	100
customer 2, romance	0	5	3

patient diagnosis with society cost

actual \ predicted	H7N9	cold	healthy
H7N9	0	1000	**100000**
cold	100	0	3000
healthy	100	30	0

check digit recognition with absolute cost

$$\mathcal{C}(y, h(\mathbf{x})) = |y - h(\mathbf{x})|$$

Cost Vector

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

- customer 1 on a romance movie: $\mathbf{c} = (0, 5, 100)$
- an H7N9 patient: $\mathbf{c} = (0, 1000, 100000)$
- absolute cost for digit 2: $\mathbf{c} = (1, 0, 1, 2)$
- "regular" classification cost for label 2: $\mathbf{c}_c^{(2)} = (1, 0, 1, 1)$

regular classification:
special case of cost-sensitive classification

Setup: Vector-Based Cost-Sensitive Binary Classification

Given

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

$$\boxed{\text{and cost vectors } \mathbf{c}_n, \text{ each} \in \mathbb{R}^K}$$

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

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

cost-sensitive classification:

can express any finite-loss supervised learning tasks

Our Contribution (Tu and Lin, ICML 2010)

	binary	multiclass
regular	well-studied	well-studied
cost-sensitive	known (Zadrozny et al., 2003)	ongoing (our works, among others)

a theoretic and algorithmic study of cost-sensitive classification, which ...

- introduces a methodology to reduce cost-sensitive classification to **regression**
- provides **strong theoretical support** for the methodology
- leads to a promising algorithm with **superior experimental results**

will describe the methodology and an algorithm

Key Idea: Cost Estimator

Goal

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

if every $\mathbf{c}[k]$ known

optimal

$g^*(\mathbf{x}) = \text{argmin}_{1 \le k \le K} \mathbf{c}[k]$

if $r_k(\mathbf{x}) \approx \mathbf{c}[k]$ well

approximately good

$g_r(\mathbf{x}) = \text{argmin}_{1 \le k \le K} r_k(\mathbf{x})$

how to get cost estimator r_k? **regression**

Cost Estimator by Per-class Regression

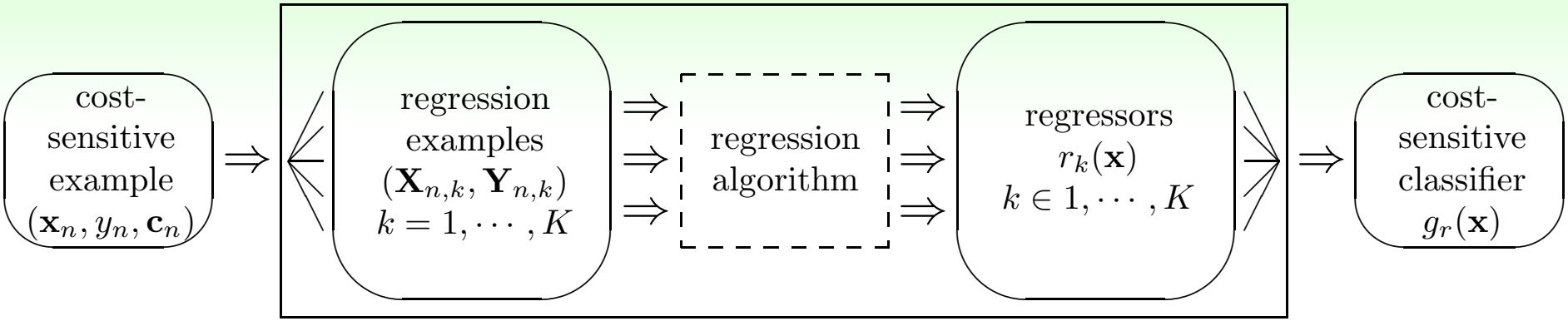
Given

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

input	$\mathbf{c}_n[1]$	input	$\mathbf{c}_n[2]$	$\ldots$	input	$\mathbf{c}_n[K]$
$\mathbf{x}_1$	0,	$\mathbf{x}_1$	2,		$\mathbf{x}_1$	1
$\mathbf{x}_2$	1,	$\mathbf{x}_2$	3,		$\mathbf{x}_2$	5
$\ldots$						
$\mathbf{x}_N$	6,	$\mathbf{x}_N$	1,		$\mathbf{x}_N$	0
	r_1		r_2			r_K

want: $r_k(\mathbf{x}) \approx \mathbf{c}[k]$ for all future $(\mathbf{x}, y, \mathbf{c})$ and k

The Reduction Framework



1. transform cost-sensitive examples $(\mathbf{x}_n, y_n, \mathbf{c}_n)$ to regression examples $(\mathbf{x}_{n,k}, Y_{n,k}) = (\mathbf{x}_n, \mathbf{c}_n[k])$
2. use your favorite algorithm on the regression examples and get estimators $r_k(\mathbf{x})$
3. for each new input $\mathbf{x}$, predict its class using $g_r(\mathbf{x}) = \text{argmin}_{1 \le k \le K} r_k(\mathbf{x})$

the reduction-to-regression framework:

systematic & easy to implement

Theoretical Guarantees (1/2)

$$g_r(\mathbf{x}) = \operatorname*{argmin}_{1 \le k \le K} r_k(\mathbf{x})$$

Theorem (Absolute Loss Bound)

For any set of estimators (cost estimators) $\{r_k\}_{k=1}^K$ *and for any example* $(\mathbf{x}, y, \mathbf{c})$ *with* $\mathbf{c}[y] = 0$,

$$\mathbf{c}[g_r(\mathbf{x})] \le \sum_{k=1}^{K} \left| r_k(\mathbf{x}) - \mathbf{c}[k] \right|.$$

low-cost classifier $\Longleftarrow$ accurate estimator

Theoretical Guarantees (2/2)

$$g_r(\mathbf{x}) = \operatorname*{argmin}_{1 \le k \le K} r_k(\mathbf{x})$$

Theorem (Squared Loss Bound)

For any set of estimators (cost estimators) $\{r_k\}_{k=1}^K$ *and for any example* $(\mathbf{x}, y, \mathbf{c})$ *with* $\mathbf{c}[y] = 0$,

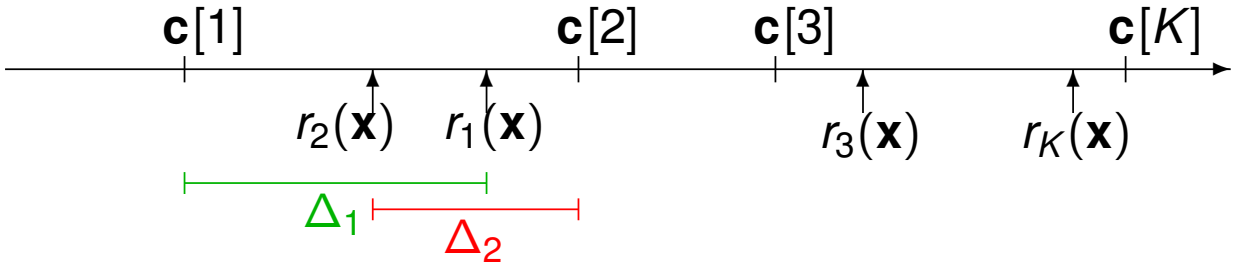
$$\mathbf{c}[g_r(\mathbf{x})] \le \sqrt{2 \sum_{k=1}^{K} (r_k(\mathbf{x}) - \mathbf{c}[k])^2}.$$

applies to common **least-square regression**

A Pictorial Proof

$$\mathbf{c}[g_r(\mathbf{x})] \leq \sum_{k=1}^{K} \left| r_k(\mathbf{x}) - \mathbf{c}[k] \right|$$

- assume $\mathbf{c}$ ordered and not degenerate:
  $y = 1; 0 = \mathbf{c}[1] < \mathbf{c}[2] \leq \cdots \leq \mathbf{c}[K]$
- assume mis-prediction $g_r(\mathbf{x}) = 2$:
  $r_2(\mathbf{x}) = \min_{1 \leq k \leq K} r_k(\mathbf{x}) \leq r_1(\mathbf{x})$



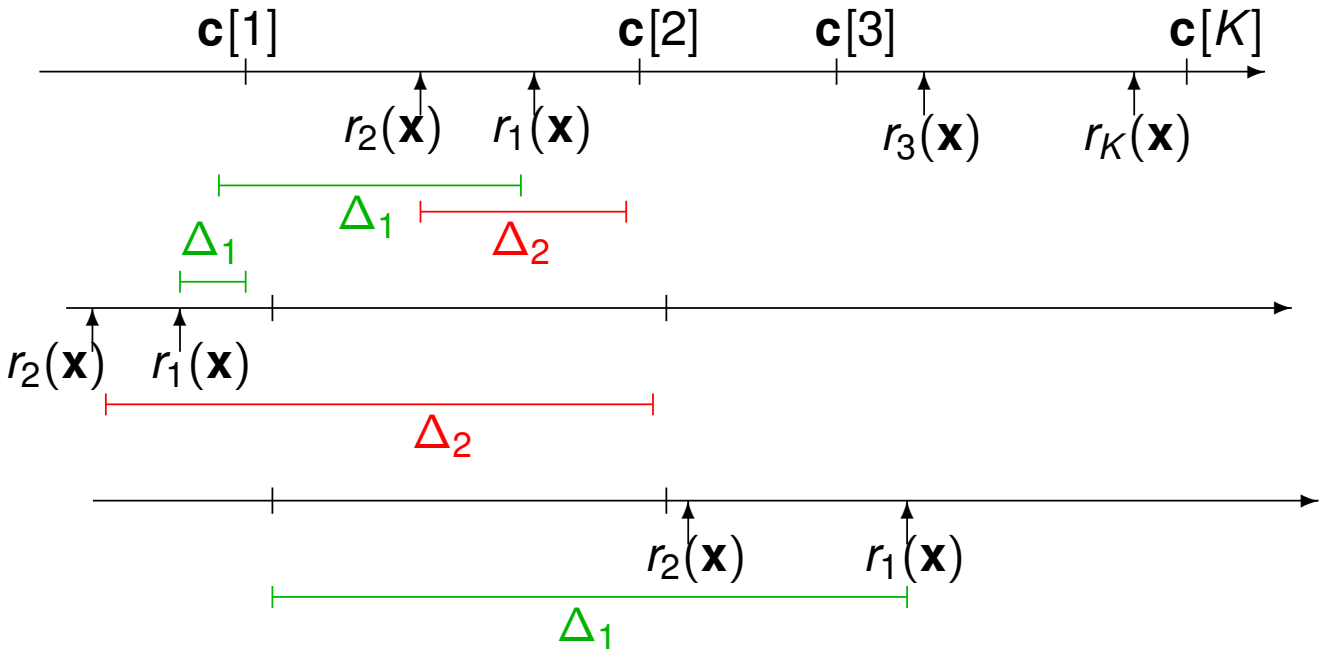
$$\mathbf{c}[2] - \underbrace{\mathbf{c}[1]}_{0} \leq \left|\Delta_1\right| + \left|\Delta_2\right| \leq \sum_{k=1}^{K} \left| r_k(\mathbf{x}) - \mathbf{c}[k] \right|$$

An Even Closer Look

let $\Delta_1 \equiv r_1(\mathbf{x}) - \mathbf{c}[1]$ and $\Delta_2 \equiv \mathbf{c}[2] - r_2(\mathbf{x})$

1. $\Delta_1 \geq 0$ and $\Delta_2 \geq 0$: $\mathbf{c}[2] \leq \Delta_1 + \Delta_2$
2. $\Delta_1 \leq 0$ and $\Delta_2 \geq 0$: $\mathbf{c}[2] \leq \Delta_2$
3. $\Delta_1 \geq 0$ and $\Delta_2 \leq 0$: $\mathbf{c}[2] \leq \Delta_1$

$$\mathbf{c}[2] \leq \max(\Delta_1, 0) + \max(\Delta_2, 0) \leq |\Delta_1| + |\Delta_2|$$



Tighter Bound with One-sided Loss

Define **one-sided loss** $\xi_k \equiv \max(\Delta_k, 0)$

with $\Delta_k \equiv \left(r_k(\mathbf{x}) - \mathbf{c}[k]\right)$ if $\mathbf{c}[k] = c_{\min}$

$\Delta_k \equiv \left(\mathbf{c}[k] - r_k(\mathbf{x})\right)$ if $\mathbf{c}[k] \neq c_{\min}$

Intuition

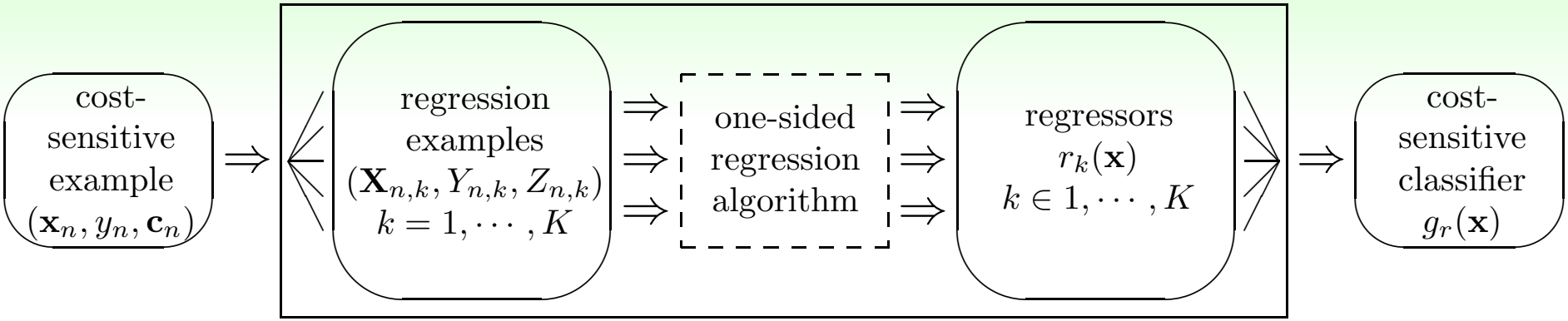
- $\mathbf{c}[k] = c_{\min}$: wish to have $r_k(\mathbf{x}) \leq \mathbf{c}[k]$
- $\mathbf{c}[k] \neq c_{\min}$: wish to have $r_k(\mathbf{x}) \geq \mathbf{c}[k]$

—**both wishes same as** $\Delta_k \leq 0$ **and hence** $\xi_k = 0$

One-sided Loss Bound:

$$\mathbf{c}[g_r(\mathbf{x})] \leq \sum_{k=1}^{K} \xi_k \leq \sum_{k=1}^{K} \left|\Delta_k\right|$$

The Improved Reduction Framework



1. transform cost-sensitive examples $(\mathbf{x}_n, y_n, \mathbf{c}_n)$ to regression examples
2. use a one-sided regression algorithm to get estimators $r_k(\mathbf{x})$
3. for each new input $\mathbf{x}$, predict its class using $g_r(\mathbf{x}) = \text{argmin}_{1 \le k \le K} r_k(\mathbf{x})$

the reduction-to-OSR framework:
need a good OSR algorithm

Regularized One-sided Hyper-linear Regression

Given

$$\left(\mathbf{x}_{n,k}, Y_{n,k}, Z_{n,k}\right) = \left(\mathbf{x}_n, \mathbf{c}_n[k], 2\left[\!\left[\mathbf{c}_n[k] = \mathbf{c}_n[y_n]\right]\!\right] - 1\right)$$

Training Goal

all training $\xi_{n,k} = \max\left(\underbrace{Z_{n,k}\left(r_k(\mathbf{x}_{n,k}) - Y_{n,k}\right)}_{\Delta_{n,k}}, 0\right)$ small

—**will drop** k

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{\lambda}{2}\langle \mathbf{w}, \mathbf{w}\rangle + \sum_{n=1}^{N} \xi_n \\ \text{to get} \quad & r_k(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x})\rangle + b \end{aligned}$$

One-sided Support Vector Regression

Regularized One-sided Hyper-linear Regression

$$\min_{\mathbf{w},b} \quad \frac{\lambda}{2}\langle \mathbf{w}, \mathbf{w} \rangle + \sum_{n=1}^{N} \xi_n$$
$$\xi_n = \max\left(Z_n \cdot \left(r_k(\mathbf{x}_n) - Y_n\right), 0\right)$$

Standard Support Vector Regression

$$\min_{\mathbf{w},b} \quad \frac{1}{2C}\langle \mathbf{w}, \mathbf{w} \rangle + \sum_{n=1}^{N} (\xi_n + \xi_n^*)$$
$$\xi_n = \max\left(+1 \cdot \left(r_k(\mathbf{x}_n) - Y_n - \epsilon\right), 0\right)$$
$$\xi_n^* = \max\left(-1 \cdot \left(r_k(\mathbf{x}_n) - Y_n + \epsilon\right), 0\right)$$

OSR-SVM = SVR + $(0 \to \epsilon)$ + (keep ξ_n or ξ_n^* by Z_n)

OSR versus Other Reductions

OSR: K regressors

How unlikely (costly) does the example belong to class k?

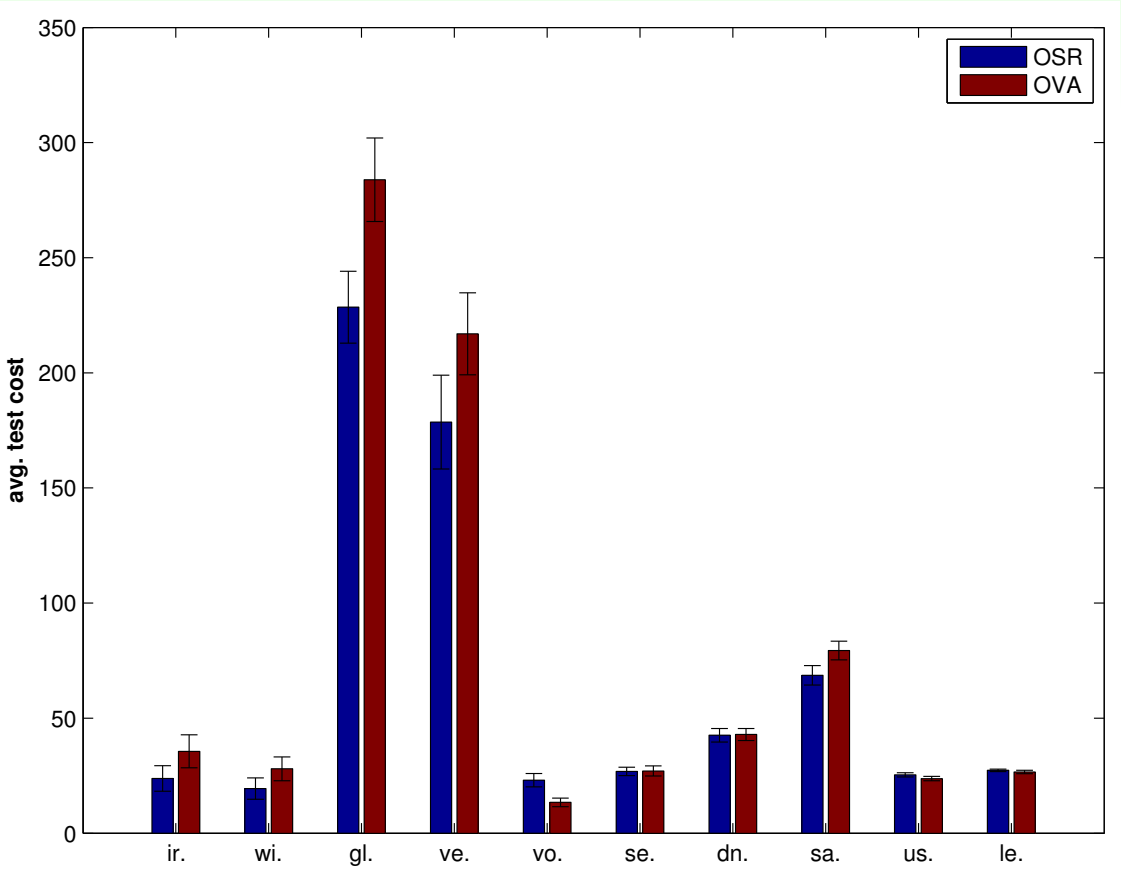
Filter Tree (FT): $K-1$ binary classifiers

Is the lowest cost within labels $\{1,4\}$ or $\{2,3\}$?
Is the lowest cost within label $\{1\}$ or $\{4\}$?
Is the lowest cost within label $\{2\}$ or $\{3\}$?

Weighted All Pairs (WAP): $\frac{K(K-1)}{2}$ binary classifiers

is $\mathbf{c}[1]$ or $\mathbf{c}[4]$ lower?

OSR-SVM on Semi-Real Data

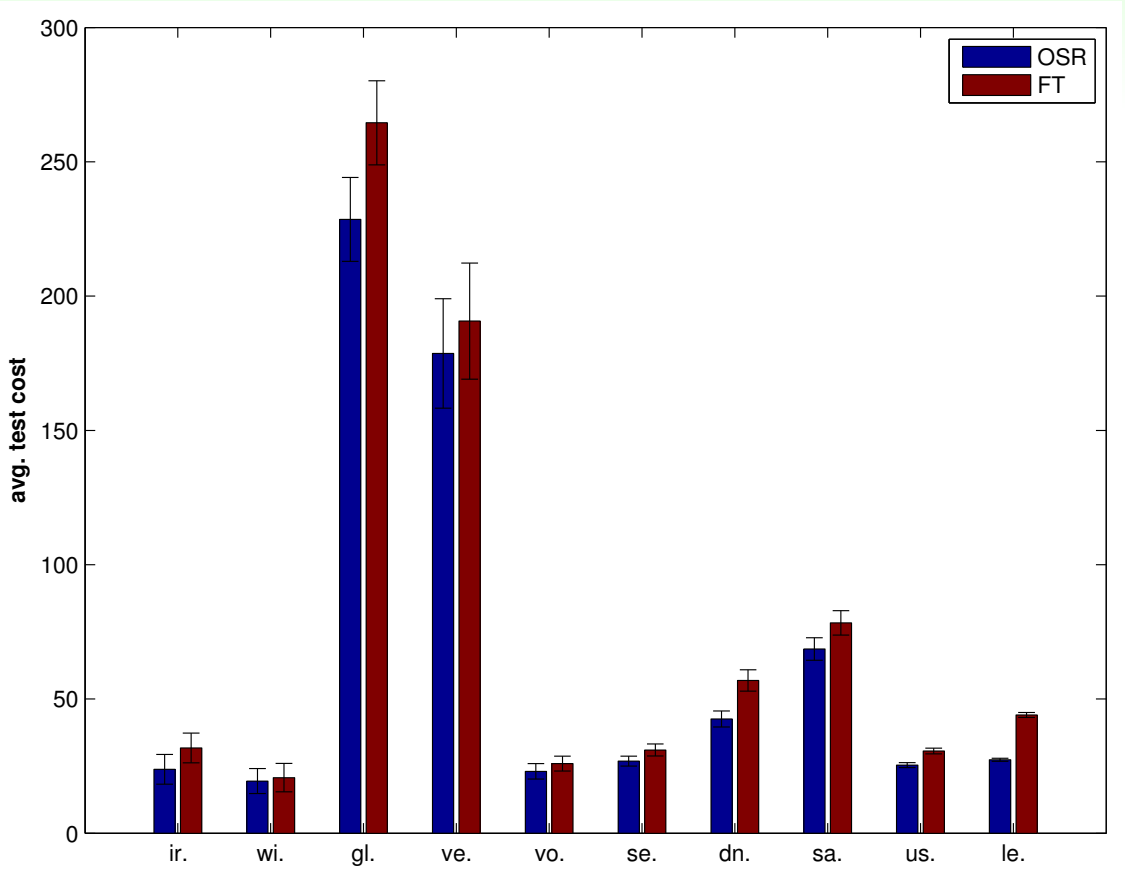


(Tu and Lin, ICML 2010)

- OSR: a cost-sensitive extension of OVA
- OVA: regular SVM

OSR often significantly better than OVA

OSR versus FT on Semi-Real Data

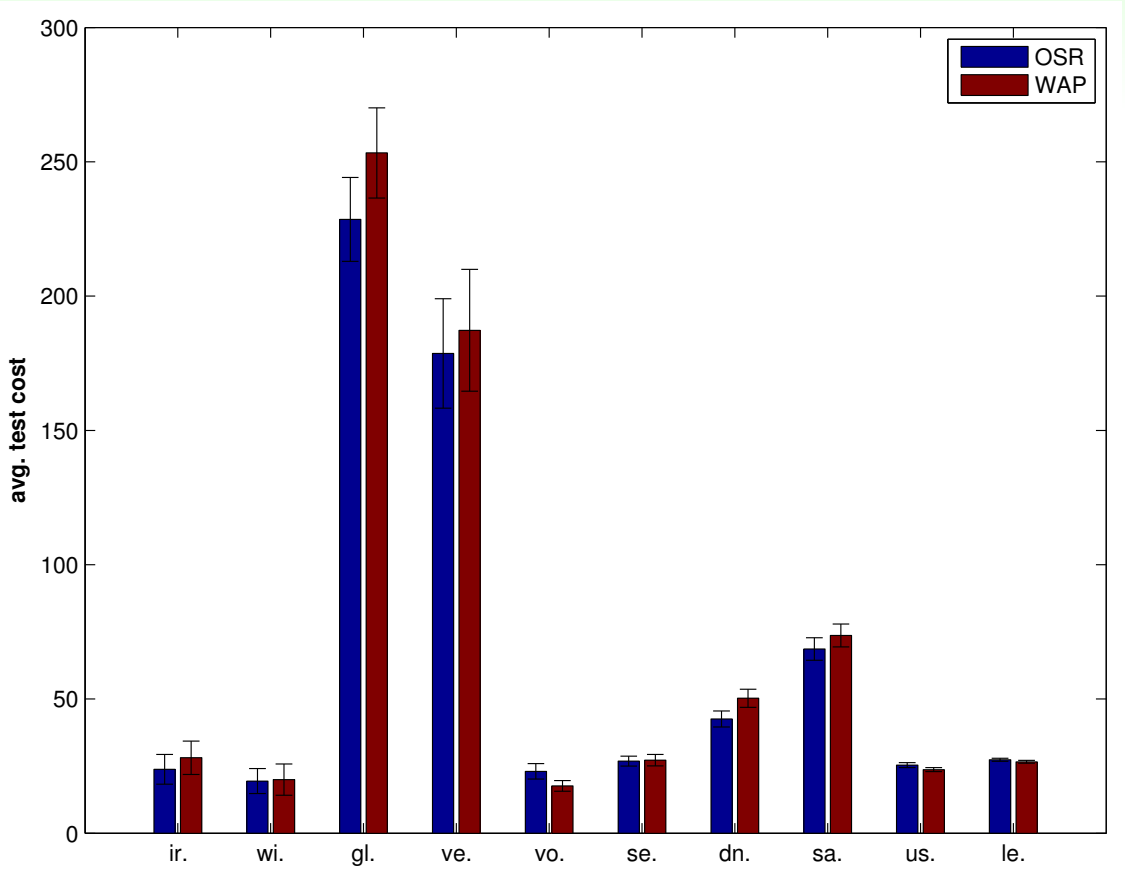


(Tu and Lin, ICML 2010)

- OSR (per-class): $O(K)$ training, $O(K)$ prediction
- FT (tournament): $O(K)$ training, $O(\log_2 K)$ prediction

FT faster, but OSR better performing

OSR versus WAP on Semi-Real Data



(Tu and Lin, ICML 2010)

- OSR (per-class): $O(K)$ training, $O(K)$ prediction
- WAP (pairwise): $O(K^2)$ training, $O(K^2)$ prediction

OSR faster and comparable performance

Six Years after OSR-SVM (Chung, Lin and Yang, IJCAI 2016)

OSR-SVM

$$\min_{\mathbf{w},b} \quad \frac{\lambda}{2}\langle \mathbf{w}, \mathbf{w}\rangle + \sum_{n=1}^{N} \xi_n$$

with

$$r_k(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x})\rangle + b$$

$$\xi_n = \max\left(Z_n \cdot \left(r_k(\mathbf{x}_n) - Y_n\right), 0\right)$$

CS Deep NNet (CSDNN)

$$\min_{\text{NNet}} \quad \text{regularizer} + \sum_{n=1}^{N} \delta_n$$

with

$$r_k(\mathbf{x}) = \text{NNet}(\mathbf{x})$$

$$\delta_n = \ln\left(1 + \exp\left(Z_n \cdot \left(r_k(\mathbf{x}_n) - Y_n\right)\right)\right)$$

- CSDNN: world's first cost-sensitive deep model via a smoother upper bound—$\delta_n \geq \xi_n$ because $\ln(1 + \exp(\bullet)) \geq \max(\bullet, 0)$
- δ_n used in both pretraining & training for better NNet feature extraction

concept of reduction-to-OSR still useful after 6 years

A Real Medical Application: Classifying Bacteria

The Problem

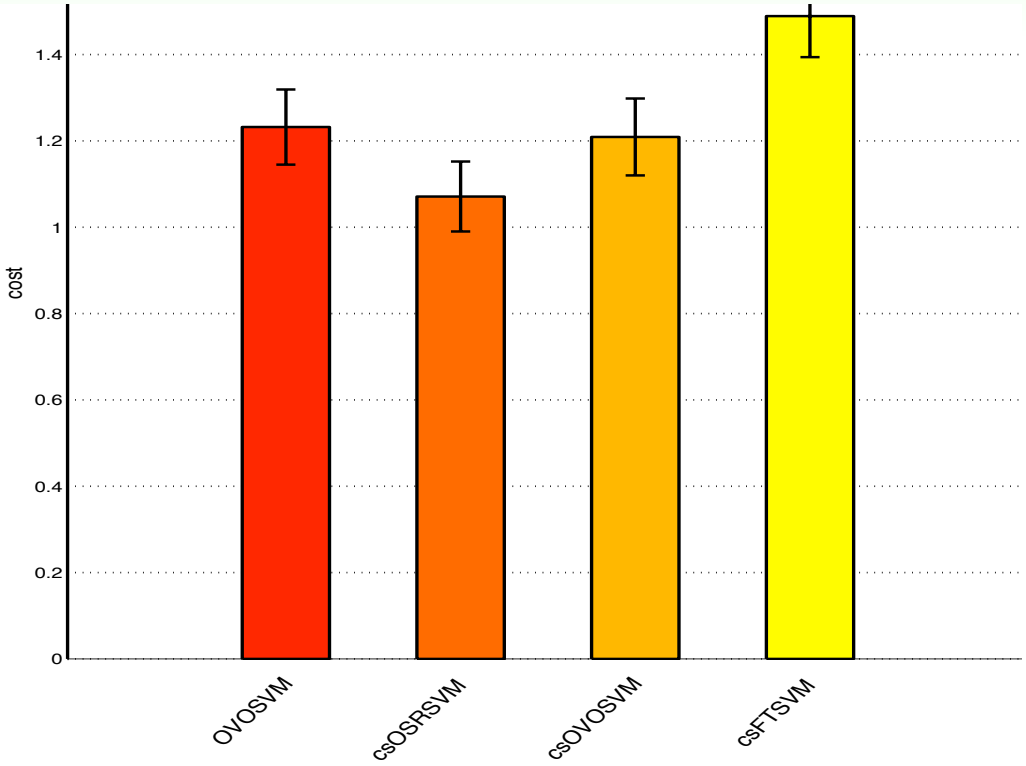
- by human doctors: **different treatments** $\Longleftrightarrow$ serious costs
- cost matrix averaged from two doctors:

	Ab	Ecoli	HI	KP	LM	Nm	Psa	Spn	Sa	GBS
Ab	0	1	10	7	9	9	5	8	9	1
Ecoli	3	0	10	8	10	10	5	10	10	2
HI	10	10	0	3	2	2	10	1	2	10
KP	7	7	3	0	4	4	6	3	3	8
LM	8	8	2	4	0	5	8	2	1	8
Nm	3	10	9	8	6	0	8	3	6	7
Psa	7	8	10	9	9	7	0	8	9	5
Spn	6	10	7	7	4	4	9	0	4	7
Sa	7	10	6	5	1	3	9	2	0	7
GBS	2	5	10	9	8	6	5	6	8	0

is cost-sensitive classification **realistic**?

OSR versus OVO/CSOVO(WAP)/FT on Bacteria Data

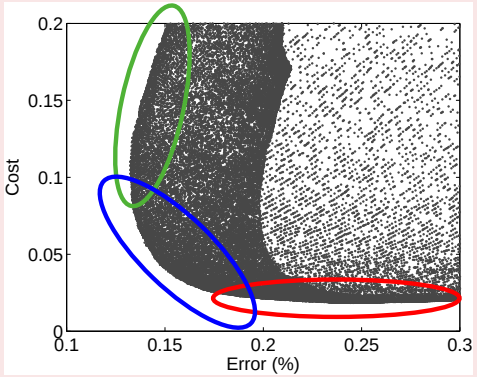
(Jan et al., BIBM 2011)



OSR best: cost-sensitive classification is helpful

Soft Cost-sensitive Classification

The Problem



- cost-sensitive classifier: low cost but high error
- traditional classifier: low error but high cost
- how can we get the blue classifiers?: low error and low cost

cost-and-error-sensitive: more suitable for medical needs

Improved OSR for Cost and Error on Semi-Real Data

key idea (Jan et al., KDD 2012): consider a 'modified' cost that mixes original cost and 'regular cost'

Cost

iris	≈
wine	≈
glass	≈
vehicle	≈
vowel	◯
segment	◯
dna	◯
satimage	≈
usps	◯
zoo	◯
splice	≈
ecoli	≈
soybean	≈

Error

iris	◯
wine	◯
glass	◯
vehicle	◯
vowel	◯
segment	◯
dna	◯
satimage	◯
usps	◯
zoo	◯
splice	◯
ecoli	◯
soybean	◯

improves other cost-sensitive classification algorithms, too

Conclusion

- reduction from cost-sensitive classification to regression: via cost estimation
- one-sided regression with solid theoretical guarantee
- superior experimental results with OSR-SVM
- OSR for deep learning: OSR-SVM $\rightarrow$ CSDNN
- OSR for medical application: towards cost-and-error-sensitive

Thank you. Questions?