A Simple Methodology for Soft Cost-sensitive Classification

Te-Kang Jan\textsuperscript{1} Da-Wei Wang\textsuperscript{1} Chi-Hung Lin\textsuperscript{2} Hsuan-Tien Lin\textsuperscript{3}

Academic Sinica

National Yang-Ming University

National Taiwan University

August 6, 2012
To classify bacteria which are very **dangerous** to human

These bacteria can be grouped into three classes:

1. Gram-positive (GP) bacteria 1
2. Gram-positive bacteria 2
3. Gram-negative (GN) bacteria

Different types of error should be charged for different cost

1. GP bacteria 1 classified as GP bacteria 1: **Correct**
2. GP bacteria 1 classified as GP bacteria 2: **Relatively OK; They share the same treatment.**
3. GP bacteria 1 classified as GN bacteria: **High cost; The treatment are different.**
Different types of error should be charged for different cost

1) GP bacteria 1 classified as GP bacteria 1: Correct
2) GP bacteria 1 classified as GP bacteria 2: Relatively OK; They share the same treatment.
3) GP bacteria 1 classified as GN bacteria: High cost; The treatment are different.

Error measure = society cost

<table>
<thead>
<tr>
<th>classify to</th>
<th>GP 1</th>
<th>GP 2</th>
<th>GN</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GP 1</td>
<td>Correct</td>
<td>Light Error</td>
<td>Serious Error</td>
</tr>
<tr>
<td>GP 2</td>
<td>Light Error</td>
<td>Correct</td>
<td>Serious Error</td>
</tr>
<tr>
<td>GN</td>
<td>Serious Error</td>
<td>Serous Error</td>
<td>Correct</td>
</tr>
</tbody>
</table>
Cost-Sensitive (CS) Classification

- CS classification charges different types of errors with different cost.

<table>
<thead>
<tr>
<th>error measure = society cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>classify to</td>
</tr>
<tr>
<td>Gram-positive</td>
</tr>
<tr>
<td>Another gram-positive</td>
</tr>
<tr>
<td>Gram-Negative</td>
</tr>
</tbody>
</table>

- **Known**, ongoing research topic
- Goal: minimized cost
Well-studied, many good algorithms

Only correct or error

error measure = society cost

<table>
<thead>
<tr>
<th>appropriate status</th>
<th>Gram-positive</th>
<th>Another gram-positive</th>
<th>Gram-Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gram-positive</td>
<td>Correct</td>
<td>Error</td>
<td>Error</td>
</tr>
<tr>
<td>Another gram-positive</td>
<td>Error</td>
<td>Correct</td>
<td>Error</td>
</tr>
<tr>
<td>Gram-Negative</td>
<td>Error</td>
<td>Error</td>
<td>Correct</td>
</tr>
</tbody>
</table>

Goal: minimized error rates

However

The minimized error rates is not close enough to this application need.
Cost Matrix → Cost Vector

Cost vector:
- A very common setting in cost-sensitive classification
- Use label and cost matrix to generate

<table>
<thead>
<tr>
<th>appropriate status</th>
<th>Gram-positive</th>
<th>Another gram-positive</th>
<th>Gram-Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gram-positive</td>
<td>0</td>
<td>30</td>
<td>100000</td>
</tr>
<tr>
<td>Another gram-positive</td>
<td>100</td>
<td>0</td>
<td>6000</td>
</tr>
<tr>
<td>Gram-Negative</td>
<td>7000</td>
<td>5000</td>
<td>0</td>
</tr>
</tbody>
</table>

Gram-positive bacteria: label 1, Another gram-positive bacteria: label 2, Gram-Negative bacteria: label 3
- cost-sensitive classification cost for label 3: \(c = (7000, 5000, 0)\)

Note
Regular classification is the special case of cost-sensitive classification.
Cost Matrix → Cost Vector

Cost vector:
- A very common setting in cost-sensitive classification
- Use label and cost matrix to generate

<table>
<thead>
<tr>
<th></th>
<th>Gram-positive</th>
<th>Another gram-positive</th>
<th>Gram-Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gram-positive</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Another gram-positive</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Gram-Negative</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Gram-positive bacteria: label 1, Another gram-positive bacteria: label 2, Gram-Negative bacteria: label 3

- regular classification cost for label 2: $\bar{c}_2 = (1, 0, 1)$

Note
Regular classification do not use the cost information.
## Training
- given a labeled data set: \( S = \{(x_n, y_n, \bar{c}_{y_n})\}, \ n = 1, \ldots, N \)
- use a learning algorithm (decision trees, neural networks, etc.)
- get a learning model

## Goal
- Try to use the learning model to get the \textbf{expected error rates} as \textbf{small} as possible

\[
\min_g E = \mathbb{E}_{(x,y,\bar{c}_y) \sim \mathcal{D}} \bar{c}_y[g(x)]
\]
A More Realistic Concept: Cost-Sensitive (CS) Classification Set Up

**Training**
- given a cost-sensitive labeled data set: \( S = \{(x_n, y_n, c_n)\} \)
- use a cost-sensitive learning algorithm (MetaCost, One-sided regression)
- get a cost-sensitive learning model

**Goal**
- try to use the learning model to get the expected cost as small as possible

\[
\min_g E = \frac{\mathbb{E}}{(x, y, c) \sim D_c} c[g(x)]
\]
To Summary Until Now

An application: different types of errors should be charged with different cost

Regular classification
- minimized error rates
- do not match the application need

Cost-sensitive classification
- minimized cost
- "looks like" match the application need?
In many cost-sensitive classification research, cost-sensitive algorithms sometimes reveal some problems:

1. **trade off to high error rates**

CS algorithms usually have trade-off between error rates and cost.
2. overfitting to cost

Some cost-sensitive experiments indicate that cost-sensitive algorithms sometimes have the higher cost than regular classification algorithms.
Regular and Cost-Sensitive Classification

Examples
A cost-sensitive artificial data sets, and use many linear classifiers to classify it.

![Diagram](image)

- **Green area**: low error rates but high cost:
  - Regular classification algorithms seek for

- **Red area**: low cost but high error rates:
  - CS classification algorithms seek for

These two types of classifiers take the trade-off to the extreme.

Our Goal:
Find the algorithms with **low cost AND low error rates**
Our Contribution

1) Provide a new type of classification which combines the cost-sensitive and regular classification

2) Provide a platform to compare cost-sensitive and regular algorithms using 22 data sets

3) To apply our algorithms on real world applications
Regular and Cost-Sensitive Classification

Regular algorithms goal

$$\min_g E = \mathcal{E}_{(x,y,\tilde{c}) \sim \mathcal{D}} \tilde{c}_y [g(x)]$$

CS algorithms goal:

$$\min_g E_c = \mathcal{E}_{(x,y,c) \sim \mathcal{D}_c} c[g(x)]$$

Our goal

$$\min_g \mathbf{E}(g) = [E_c(g), E(g)] \text{ subject to all feasible } g.$$

We have two criteria of interest.

Such a problem belongs to Multicriteria optimization problem.
Multicriteria Optimization Problem

Multicriteria optimization problem

\[
\min_g \mathbf{E}(g) = [E_1(g), E_2(g), \ldots, E_M(g)]
\]

subject to all feasible \( g \),

Solutions

- Evolutionary algorithms
- Strength Pareto Evolutionary
- Weighted Sum Approach
- .........................

Note

In this paper, we would focus on weighted sum approach.
Why Weighted Sum Approach?

- It transfers the problem to a single-criterion optimization one.
  - We are more familiar.
- It can directly reuse the modern CS algorithms.
  - Other multicriteria methods need to redesign CS algorithms.
- Simple and popular.
  - Compared to other multicriteria methods

In fact, the weighted sum approach has also been implicitly taken by other algorithms in data mining. (Sculley, KDD, 2010) combines the pairwise ranking criterion and squared regression criterion.
Weighted Sum Approach

Key Idea
Give the weight to each criterion

An Optimal solution in Multicriteria Optimization
- Often there is no global optimal solution that is the best in terms of every criterion.
- Instead, try to seek the better solutions where there exists no other feasible solution that improves the value of at least one objective function without deteriorating any other objective.

Multicriteria optimization problem using weighted sum approach

\[
\begin{align*}
\min_{g} & \quad E(g) = [E_1(g), E_2(g), \ldots, E_M(g)] \\
\min_{g} & \quad \sum_{m=1}^{M} \alpha_m E_m(g) \text{ subject to all feasible } g,
\end{align*}
\]
## Soft Cost-Sensitive Classification

### Regular algorithms goal

\[
\min_g E = \mathbb{E}_{(x,y,c) \sim \mathcal{D}} \bar{c}_y[g(x)]
\]

### CS algorithms goal:

\[
\min_g E_c = \mathbb{E}_{(x,y,c) \sim \mathcal{D}_c} c[g(x)]
\]

### Our goal using weighted sum approach

\[
\min_g \alpha_1 E_c(g) + \alpha_2 E(g)
\]

Without loss of generality, let \( \alpha_1 = 1 - \alpha \) and \( \alpha_2 = \alpha \) for \( \alpha \in [0, 1] \)

### Our goal using weighted sum approach (2)

\[
\min_g \mathbb{E}_{(x,y,c) \sim \mathcal{D}_c} \left( (1 - \alpha) \left( c[g(x)] \right) + \alpha \left( \bar{c}_y[g(x)] \right) \right)
\]

---

Jan et al. (Academic Sinica)  
A Simple Method for Soft Cost-Sensitive  
August 6, 2012  
19 / 33
Our goal using weighted sum approach (2)

\[
\min_{g} \mathbb{E}_{(x,y,c) \sim D_c} (1 - \alpha) \left( c[g(x)] \right) + \alpha \left( \bar{c}_y[g(x)] \right)
\]

Note

For any given \( \alpha \), such an optimization problem is exactly the cost-sensitive classification problem with modified cost. The new cost vector is the linear combination from regular classification cost vector and cost vector.
Soft Cost-Sensitive Classification (2)

Our goal using weighted sum approach

\[
\min_{g} \mathcal{E}_{(x,y,c) \sim \mathcal{D}_c} \left( 1 - \alpha \right) \left( c[g(x)] \right) + \alpha \left( \bar{c}_y[g(x)] \right)
\]

Regular (Non cost-sensitive): \( \alpha = 1 \)

\[
\min E = \mathcal{E}_{(x,y,c) \sim \mathcal{D}} \bar{c}_y[g(x)]
\]

Cost-sensitive (Hard cost-sensitive): \( \alpha = 0 \)

\[
\min E_c = \mathcal{E}_{(x,y,c) \sim \mathcal{D}_c} c[g(x)]
\]

non-zero values of \( \alpha \) (soft cost-sensitive)

\[
\min_{g} \mathcal{E}_{(x,y,c) \sim \mathcal{D}_c} \left( 1 - \alpha \right) \left( c[g(x)] \right) + \alpha \left( \bar{c}_y[g(x)] \right)
\]
Regularization for Cost-sensitive Classification

\[
\min_{g \sim D_c} \mathcal{E} \left( 1 - \alpha \right) \left( c[g(x)] \right) + \alpha \left( \bar{c}_y[g(x)] \right)
\]

- It may be overfitting if we only use the limited information in the training set.
- The added term \( \alpha \left( \bar{c}_y[g(x)] \right) \) can be treated as restricting the number of low-cost classifiers.
- The restriction is similar to common regularization schemes.
For soft cost-sensitive algorithms, we treat $\alpha$ as the parameter.

- The intermediate value $\alpha$ leads to the better performance.
- Different data sets need different $\alpha$. 

Note
Couple all the algorithms with the support vector machine (SVM) with the perceptron kernel as the internal learner.

**Cross Validation**

- For regular algorithms, regularization parameter $\lambda$ chosen by minimized 5-fold CV error rates.
- For cost-sensitive algorithms, regularization parameter $\lambda$ chosen by minimized 5-fold CV cost.
- For soft CS, regularization parameter $\lambda$ and $\alpha$ chosen by minimized 5-fold CV cost.
## Compared Methods

### OVO-type

1. **OVOSVM**: a well-known regular multi-class SVM
2. **CSOVOSVM**: Cost-sensitive version of OVOSVM (Lin, 2010)
3. **soft-CSOVO**: proposed method; adapting CSOVOSVM

### OVA-type

1. **OVASVM**: a well-known regular multi-class SVM
2. **OSRSVM**: Cost-sensitive version of OVASVM (Tu et al, 2010)
3. **soft-OSR**: proposed method; adapting OSRSVM

### Filter Ttee-type

1. **FTSVM**: a well-known regular multi-class SVM
2. **CSFTSVM**: Cost-sensitive version of FTSVM (Beygelzimer et al, 2010)
3. **soft-CSFT**: proposed method; adapting CSFTSVM.
Comparisons soft CS algorithms with the CS algorithms and their sibling

<table>
<thead>
<tr>
<th>data set</th>
<th>OVA</th>
<th>OSR</th>
<th>OVO</th>
<th>CSOVO</th>
<th>FT</th>
<th>CSFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>iris</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>○</td>
<td></td>
</tr>
<tr>
<td>wine</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>○</td>
<td></td>
</tr>
<tr>
<td>glass</td>
<td>○</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>vehicle</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>vowel</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>segment</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>dna</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>satimage</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>usps</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>zoo</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>yeast</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>pagblock</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>anneal</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>solar</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>splice</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>ecoli</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>nursery</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>soybean</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>arrhythmia</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>○</td>
<td></td>
</tr>
<tr>
<td>optdigits</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>○</td>
<td></td>
</tr>
<tr>
<td>mfeat</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>pendigit</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>#○</td>
<td>15</td>
<td>8</td>
<td>9</td>
<td>6</td>
<td>9</td>
<td>4</td>
</tr>
</tbody>
</table>

- 22 benchmark datasets
- ○: soft CS algorithms significantly better
- ≈: otherwise
- Red: regular classification algorithms
- Blue: cost-sensitive classification algorithms

**Note**

Soft CS algorithms can achieve better and similar performance than CS.
Comparisons soft CS algorithms with the CS algorithms

<table>
<thead>
<tr>
<th>data set</th>
<th>OSR</th>
<th>CSOVO</th>
<th>CSFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>iris</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>wine</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>glass</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>vehicle</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>vowel</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>segment</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>dna</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>satimage</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>usps</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>zoo</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>yeast</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>pagblock</td>
<td>○</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>anneal</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>solar</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>splice</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>ecoli</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>nursery</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>soybean</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>arrhythmia</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>optdigits</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>mfeat</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>pendigit</td>
<td>≈</td>
<td>≈</td>
<td>≈</td>
</tr>
<tr>
<td>#○</td>
<td>16</td>
<td>13</td>
<td>14</td>
</tr>
</tbody>
</table>

- 22 benchmark datasets
- ○: soft CS algorithms significantly better
- ≈: otherwise
- Red: regular classification algorithms
- Blue: cost-sensitive classification algorithms

**Note**

Soft CS algorithms can achieve **better** performance than CS.
Real World Biomedical Application

- Classifying the bacterial meningitis:
  - a serious and often life-threatening form of the meningitis infection.
- Collected the clinical sample from National Taiwan University Hospital and ATCC
- Cost matrix is given by the senior doctor

<table>
<thead>
<tr>
<th>real class</th>
<th>Ab</th>
<th>Ecoli</th>
<th>HI</th>
<th>KP</th>
<th>LM</th>
<th>Nm</th>
<th>Psa</th>
<th>Spn</th>
<th>Sa</th>
<th>GBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ab</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Ecoli</td>
<td>3</td>
<td>0</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>HI</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>KP</td>
<td>7</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>LM</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Nm</td>
<td>3</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>Psa</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>8</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>Spn</td>
<td>6</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>Sa</td>
<td>7</td>
<td>10</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>GBS</td>
<td>2</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>
Real World Biomedical Application

Figure: Our goal

Figure: OVA-type Results

Note
CS algorithm get the higher error rates than regular sibling.
Real World Biomedical Application

Figure: Our goal

Note
Soft CS algorithms match our goal: lower error rates
Real World Biomedical Application

Figure: Our goal

Figure: OVO-type Results

Note

This application is the typical cost overfitting problem: CS algorithms get the worse cost performance than their regular sibling.
Real World Biomedical Application

Figure: Our goal

Note
But soft CS algorithm can avoid the overfitting: soft CS algorithms get the better cost and error rates performance than CS ones
Conclusion

- Trade-off between the cost and the error rate in cost-sensitive classification tasks:
  - propose a simple methodology to take both criteria into account
- Call Multicriteria problem
  - using the weighted sum approach
- Feeding the exact cost information may not be the best approach
  - a new insight for cost-sensitive classification

Thank you for your attention!