Ternary Decoding of Error Correcting Output Codes

Sergio Escalera
Oriol Pujol
Petia Radeva

Iberoamerican Congress on Pattern Recognition 2006
INDEX

1. ECOC

2. Ternary decoding

3. Results

4. Conclusions
Ternary Decoding of Error Correcting Output Codes

<table>
<thead>
<tr>
<th>ECOC</th>
<th>Ternary decoding</th>
<th>Results</th>
<th>Conclusions</th>
</tr>
</thead>
</table>

ERROR CORRECTING OUTPUT CODES

A general framework for solving multiclass categorization problems.

---

**Solving Multiclass Learning Problems via Error-Correcting Output Codes**

Thomas G. Dietterich  
*Department of Computer Science, 303 Dearborn Hall, Oregon State University, Corvallis, OR 97331 USA*

Ghulum Bakiri  
*Department of Computer Science, University of Bahrain, Isa Town, Bahrain*

---

**Ensemble strategy based on the reduction of the multi-class problem in different sets of binary problems.**

**How are the sets defined?**

**How are the classifiers combined?**

---

Iberoamerican Congress on Pattern Recognition 2006
It is a label perturbation technique that works in the following way:

**Coding step:** How many base classifiers? Which ones? Strategy to decompose a multiclass problem into complementary two “super-class” problems (a “super-class” a set of the original classes).

**Decoding step:** How do we decide the class of a new sample from the results of base classifiers? We expect that the decoding will be robust to error from learning algorithm, features and training samples.
## Ternary Decoding of Error Correcting Output Codes

### Example

**Classification Trees**

1. **Classifier 1**
   - **C1**: Sports
   - **C2**: Business
   - **C3**: Politics
   - **C4**: Arts

2. **Classifier 2**
3. **Classifier 3**

**Coding Matrix**

<table>
<thead>
<tr>
<th></th>
<th>C1 (Sports)</th>
<th>C2 (Business)</th>
<th>C3 (Politics)</th>
<th>C4 (Arts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>Class 2</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Class 3</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Class 4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Code for class C4**

Given a test sample we obtain a code according to the output of each classifier and find the “closest” code.

<table>
<thead>
<tr>
<th>X</th>
<th>-1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
</table>

### Conclusions

Iberoamerican Congress on Pattern Recognition 2006
## ECOC

<table>
<thead>
<tr>
<th>Ternary decoding</th>
<th>Results</th>
<th>Conclusions</th>
</tr>
</thead>
</table>

### Standard strategies

#### Coding

- **One-vs-one**
- **One-vs-all**
- **Dense Random**
- **Sparse Random**

#### 1 versus All

- **Code length: $N_c$**

#### Ternary codes

1 versus 1: “All pairs”

- **Code length: $N_c (N_c-1)/2$**

#### Random Dense ECOC

- **Code length: $10 \log N_c$**

#### Random Sparse ECOC

- **Code length: $15 \log N_c$**
Motivation

Many real problems involve a great number of classes.

- one-versus-all is the dominant strategy (e.g. shared boosting).

**Question:** how can we increase the technique performance while keeping the codeword length small?

**Answer:** problem dependent codification (the codeword length depends on the ensemble performance instead of being pre-fixed)
Traditional decoding

Hamming distance  
\[ d(x, y^i) = \sum_{j=1}^{n} |x_j - y_j^i| / 2 \]

Euclidean distance  
\[ d(x, y^i) = \sqrt{\sum_{j=1}^{n} (x_j - y_j^i)^2} \]

**Fig. 1.** Example of ternary matrix \( M \) for a 4-class problem. A new test codeword is misclassified due to the confusion of using the traditional decoding strategies.
### Inverse Hamming distance

\[ \Delta(i, j) = d(y^i, y^j) \]

\[ Q = [q_1, q_2, \ldots, q_{N_c}] \]

\[ Q = \Delta^{-1} D_T. \]

### Attenuated Euclidean distance

\[ d(x, y^i) = \sqrt{\sum_{j=1}^{n} |y_j^i| \ (x_j - y_j^i)^2} \]

### Loss-based decoding

\[ d^*(\ell, i) = \sum_{j=1}^{n} L(M(i, j) \cdot f(\ell, j)) \]


**Laplacian strategy** We propose a Laplacian decoding strategy to give to each class the distance according to the number of coincidences between the input codeword and the class codeword, normalized by the errors without considering the zero symbol. In this way, the coded positions of the codewords with more zero symbols attain more importance. The distance is estimated by:

\[
d(x, y^i) = \frac{C_i + 1}{C_i + E_i + K}
\]

where \(C_i\) is the number of coincidences from the test codeword and the codeword for class \(i\), \(E_i\) is the number of failures from the test codeword and the codeword for class \(i\), and \(K\) is an integer value that codifies the number of classes considered by the classifier, in this case 2, due to the binary partitions of the base classifiers. The offset \(1/K\) is the default value (bias) in case that the coincidences and failures tend to zero. Note that when the number of \(C\) and \(E\) are sufficiently high, the factor \(1/K\) does not contribute:

\[
\lim_{C \to 0, E \to 0} d(x, y^i) = \frac{1}{K} \quad \lim_{C \gg K} d(x, y^i) = \frac{C}{C + E}
\]
Beta Density Distribution Pessimistic Strategy

$$
\psi(z, \alpha, \beta) = \frac{1}{K} z^\alpha (1 - z)^\beta
$$

$$
a_i : \int_{Z_i - a_i}^{Z_i} \psi_i(z) = \frac{1}{3}
$$

**Fig. 2.** Pessimistic Density Probability estimations for the test codeword $x$ and the matrix $M$ for the four classes of fig. 1. The probability for the second class allows a successful classification in this case.
<table>
<thead>
<tr>
<th>Problem</th>
<th>#Train</th>
<th>#Test</th>
<th>#Attributes</th>
<th>#Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dermatology</td>
<td>366</td>
<td>-</td>
<td>34</td>
<td>6</td>
</tr>
<tr>
<td>Ecoli</td>
<td>336</td>
<td>-</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Glass</td>
<td>214</td>
<td>-</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Vowel</td>
<td>990</td>
<td>-</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>Yeast</td>
<td>1484</td>
<td>-</td>
<td>8</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 1. UCI repository databases characteristics.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>0% zeros</th>
<th>10% zeros</th>
<th>20% zeros</th>
<th>30% zeros</th>
<th>40% zeros</th>
<th>50% zeros</th>
<th>Global rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD</td>
<td>3.2</td>
<td>3.2</td>
<td>4.4</td>
<td>4.2</td>
<td>4.6</td>
<td>4.0</td>
<td>3.9</td>
</tr>
<tr>
<td>ED</td>
<td>3.2</td>
<td>3.2</td>
<td>2.4</td>
<td>2.2</td>
<td>2.6</td>
<td>3.2</td>
<td>2.8</td>
</tr>
<tr>
<td>AED</td>
<td>3.2</td>
<td>3.6</td>
<td>4.6</td>
<td>3.8</td>
<td>2.4</td>
<td>4.0</td>
<td>3.6</td>
</tr>
<tr>
<td>IHD</td>
<td>3.4</td>
<td>4.0</td>
<td>5.8</td>
<td>4.0</td>
<td>6.0</td>
<td>5.2</td>
<td>4.7</td>
</tr>
<tr>
<td>LLB</td>
<td>1.6</td>
<td>6.8</td>
<td>7.0</td>
<td>6.8</td>
<td>6.6</td>
<td>7.2</td>
<td>6.0</td>
</tr>
<tr>
<td>ELB</td>
<td>1.6</td>
<td>4.2</td>
<td>6.8</td>
<td>5.2</td>
<td>5.8</td>
<td>5.6</td>
<td>4.9</td>
</tr>
<tr>
<td>LAP</td>
<td>2.4</td>
<td>2.2</td>
<td>2.2</td>
<td>2.0</td>
<td>1.8</td>
<td>1.6</td>
<td>2.0</td>
</tr>
<tr>
<td>β-DEN</td>
<td>2.4</td>
<td>2.4</td>
<td>1.8</td>
<td>1.0</td>
<td>2.4</td>
<td>1.2</td>
<td>1.9</td>
</tr>
</tbody>
</table>

Table 2. Mean ranking evolution for the methods on the UCI databases tests when the number of zeros is increased.
Ternary Decoding of Error Correcting Output Codes

Iberoamerican Congress on Pattern Recognition 2006
Conclusions

- Special treatment of the zero symbol
- Laplian and Beta decoding improve state-of-the-art decoding strategies
Thank you!