

Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers

Erin Allwein, Robert Schapire and Yoram Singer
Journal of Machine Learning Research,
1:113-141, 2000

CSE 254: Seminar on Learning Algorithms
Professor: Charles Elkan
Student: Aldebaro Klautau

April 23, 2001

1

Outline

- Motivation
- Definitions
 - *Margin, loss, output coding*
- Paper contributions
 - *Unified view of classifiers, "0" entries in coding matrix, bounds and simulations*
- Algorithm
- Bound on training error
- Experimental results
 - *Compare:* a) Hamming vs. loss-based decoding
 - b) different coding matrices
 - c) simple binary problems vs. robust matrix
- Conclusions

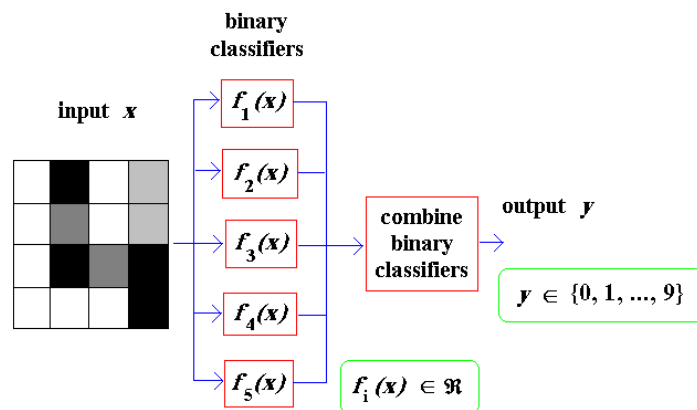
2

Motivation

- Many applications require **multiclass** categorization
- Algorithms that *easily* handle the multiclass case include C4.5 and CART
- ☹️ Some do not *easily* handle the multiclass case, as AdaBoost and SVM
- 😊 Alternative: reduce the multiclass problem to **multiple binary classifications**

3

Reducing multiclass to multiple binary problems



- Paper proposes general framework that unifies several methods, namely: **binary margin-based** algorithms coupled through a **coding matrix**

4

Binary margin-based learning algorithm

- Input: total of m binary labeled training examples

$$(x_1, y_1), \dots, (x_m, y_m) \text{ such that}$$

$$x_i \in X \text{ and } y_i \in \{-1, +1\}$$

- Output: real-valued function (hypothesis)

$$f : X \rightarrow \mathfrak{R}$$

- **Binary margin** of a training example (x, y) is defined as:

$$z = y f(x)$$



$z > 0$ if and only if f classifies x correctly

5

- Classification error: $\frac{1}{m} \sum_i^m \mathbb{I}[y_i f(x_i) \leq 0]$

☹ It is difficult to minimize classification error

☺ Instead, minimize some **loss function** of the **margin**
 $L(z) = L(y f(x))$ of each example (x, y)

- Algorithms that use margin-based loss: support-vector machines (SVM), AdaBoost, regression, decision trees, etc.

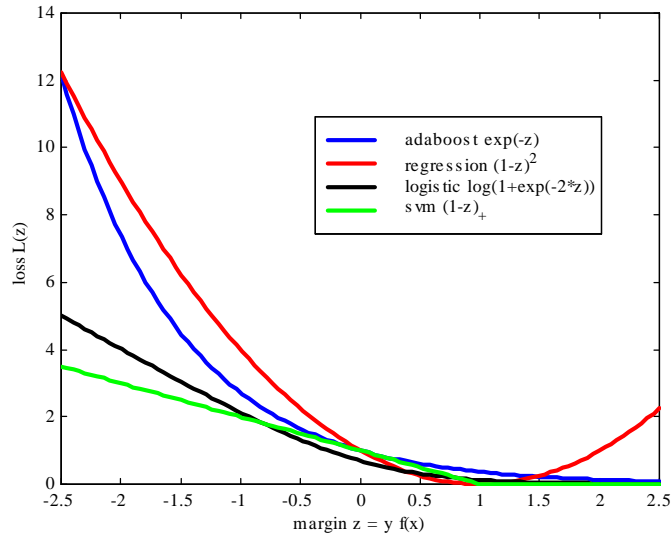
- Example: neural networks and least square regression attempt to minimize squared error

$$(y - f(x))^2 = y^2(y - f(x))^2 = (yy - yf(x))^2 = (1 - yf(x))^2$$

$$L(z) = (1 - z)^2$$

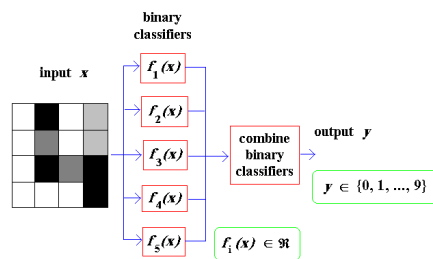
6

Loss function of popular algorithms $L : \mathfrak{R} \rightarrow [0, \infty)$



7

- Scenario: training sequence has labels drawn from set with cardinality $k > 2$ and the algorithm(s) assume $k = 2$
- Two popular alternatives: *one-against-all* and *all-pairs*



- For $k = 10$ classes

| Approach | Number of classifiers |
|-----------------|-----------------------|
| one-against-all | 10 |
| all-pairs | 45 |
| complete | 511 |

8

Output coding for solving multiclass problems

| Class | Code word (each column corresponds to a binary classifier) | | | | | |
|-------|------------------------------------------------------------|-----------------|---------------|--------------|--------------------|---------------------|
| | vertical line | horizontal line | diagonal line | closed curve | curve open to left | curve open to right |
| 0 | -1 | -1 | -1 | +1 | -1 | -1 |
| 1 | +1 | -1 | -1 | -1 | -1 | -1 |
| 2 | -1 | +1 | +1 | -1 | +1 | -1 |
| 3 | -1 | -1 | -1 | -1 | +1 | -1 |
| ... | | | ... | | | |
| 9 | -1 | -1 | +1 | +1 | -1 | -1 |

$k = 10$

$k =$ number of classes (rows) and $l =$ number of classifiers (columns)

9

Error correcting output code *ECOC*, proposed by Dietterich and Bakiri, 1995

| Class | Code word (each column corresponds to a binary classifier) | | | | | |
|-------|------------------------------------------------------------|-----------------|---------------|--------------|--------------------|---------------------|
| | vertical line | horizontal line | diagonal line | closed curve | curve open to left | curve open to right |
| 0 | -1 | -1 | -1 | +1 | -1 | -1 |
| 1 | +1 | -1 | -1 | -1 | -1 | -1 |
| 2 | -1 | +1 | +1 | -1 | +1 | -1 |
| 3 | -1 | -1 | -1 | -1 | +1 | -1 |
| ... | | | ... | | | |
| 9 | -1 | -1 | +1 | +1 | -1 | -1 |

- Associate each class r with a row of a “coding matrix”
- Train each binary classifier
- Each example labeled y is mapped to $M(y,s)$
- Obtain l hypotheses f_1 to f_l
- Given test example x , choose the row y of M that is “closest” to binary predictions $(f_1(x), \dots, f_l(x))$ according to some distance (e.g. Hamming)

10

Summary of paper contributions

- **Unified view** of margin classifiers
- Possibility of **"0" entry** in ECOC matrix
- **Decoding** when matrix has "0" entries
- **Bound on training** error (general)
- **Bound on testing** error (restricted to AdaBoost)
- **Experimental results**

11

Idea: allow entries "0"

- The coding matrix M is taken from the extended set $\{-1, 0, +1\}^{k \times l}$
- The entry $M(r, s) = 0$ indicates we do not care how f_s categorizes examples with label r

One-against-all $\rightarrow k$ by k

| | | | |
|----|----|----|----|
| +1 | -1 | -1 | -1 |
| -1 | +1 | -1 | -1 |
| -1 | -1 | +1 | -1 |
| -1 | -1 | -1 | +1 |

All-pairs $\rightarrow k$ by $\binom{k}{2}$

| | | | | | |
|----|----|----|----|----|----|
| +1 | +1 | +1 | 0 | 0 | 0 |
| -1 | 0 | 0 | +1 | +1 | 0 |
| 0 | -1 | 0 | -1 | 0 | +1 |
| 0 | 0 | -1 | 0 | -1 | -1 |

12

Algorithm

- Associate each class r with a row of a coding matrix $M \in \{-1, 0, +1\}^{k \times l}$
- Train the binary classifier for each column $s=1, \dots, l$.
- For training classifier s , example labeled y is mapped to $M(y,s)$. **Omit examples** for which $M(y,s) = 0$
- Obtain l classifiers
- Given test example x , choose the row y of M that is “closest” to binary predictions $(f_1(x), \dots, f_l(x))$ according to some distance (e.g. **modified Hamming**)

13

Distance

- Two intuitive options
 - a) “Quantize” predictions and then use a generalized **Hamming** distance

$$\Delta(\mathbf{u}, \mathbf{v}) = \sum_{s=1}^l \frac{1 - u_s v_s}{2}$$

| | | | | | |
|------------------|---|----|-----|-----|----|
| String A | 1 | -1 | 0 | 0 | -1 |
| String B | 1 | 1 | 1 | 0 | -1 |
| Distance parcels | 0 | 1 | 0.5 | 0.5 | 0 |

- b) Loss-based decoding: for each row, calculate the margin z_s of each classifier s and sum their losses $L(z_s)$ adopting the same L used by the binary classifier. **Sounds better because the magnitude of predictions are an indication of a level of “confidence”**

14

Hamming vs. loss-based decoding

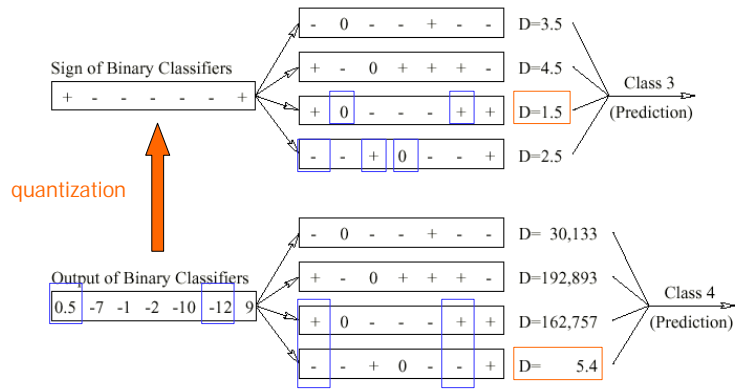
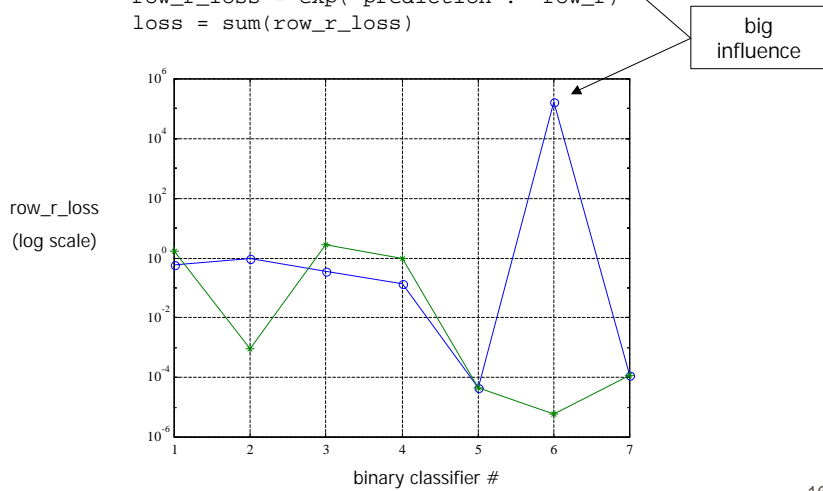


Figure 2: An illustration of the multiclass prediction procedure for Hamming decoding (top) and loss-based decoding (bottom) for a 4-class problem using a code of length 7. The exponential function was used for the loss-based decoding.

15

Losses for classes 3 and 4 in fig. 2

```
prediction = [0.5 -7 -1 -2 -10 -12 9]
row_3 = [+1 0 -1 -1 -1 +1 +1]
row_4 = [-1 -1 +1 0 -1 -1 +1]
row_r_loss = exp(-prediction .* row_r)
loss = sum(row_r_loss)
```



16

Bound on training error

- Training error

$$E \leq \frac{l\varepsilon}{\rho L(0)}$$

- Average binary loss

$$\varepsilon = \frac{1}{ml} \sum_{i=1}^m \sum_{s=1}^l L(M(y_i, s) f_s(x_i))$$

- Minimum distance between pair of rows

$$\rho = \min\{\Delta(\mathbf{M}(r_1), \mathbf{M}(r_2)) : r_1 \neq r_2\}$$

One-against-all $\rightarrow \rho = 2$

| | | | |
|----|----|----|----|
| +1 | -1 | -1 | -1 |
| -1 | +1 | -1 | -1 |
| -1 | -1 | +1 | -1 |
| -1 | -1 | -1 | +1 |

All-pairs $\rightarrow \rho = 1 + 1/2 (l - 1)$

| | | | | | |
|----|----|----|----|----|----|
| +1 | +1 | +1 | 0 | 0 | 0 |
| -1 | 0 | 0 | +1 | +1 | 0 |
| 0 | -1 | 0 | -1 | 0 | +1 |
| 0 | 0 | -1 | 0 | -1 | -1 |

17

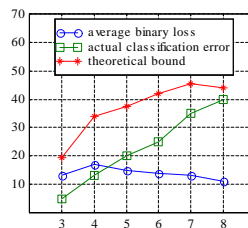
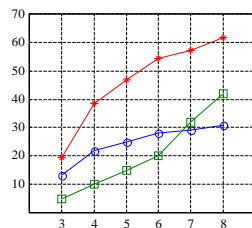
Bound on training error

$$E \leq \frac{l\varepsilon}{\rho L(0)}$$

Simulation: AdaBoost with $L(0) = 1$

complete code $E \leq \frac{(2^{k-1} - 1)\varepsilon}{2^{k-2}} \approx 2\varepsilon$

one-against-all $E \leq \frac{k\varepsilon}{2}$



classes

18

Bound on training error

Requirement:

$$\frac{L(z) + L(-z)}{2} \geq L(0) > 0$$

Do not need to be convex:
 $\sin(z) + 1$

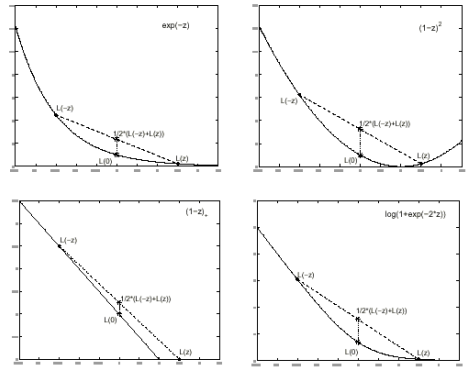


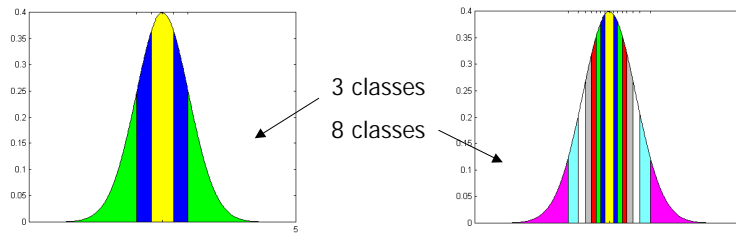
Figure 1: Some of the margin-based loss functions discussed in the paper: the exponential loss used by AdaBoost (top left); the square loss used in least-squares regression (top right); the “hinge” loss used by support-vector machines (bottom left); and the logistic loss used in logistic regression (bottom right).

Experimental results

- Tradeoff between simple binary problems and robust coding matrix
- Hamming versus loss-based decoding (AdaBoost and SVM)
- Comparison among different output codes (AdaBoost and SVM)

Experiment 1 - synthetic data

- Set thresholds to have exactly 100 examples per class
- Label test examples using these thresholds
- Use AdaBoost
 - Weak hypotheses: set of thresholds
 - threshold t would label x as $+1$ if $|x| < t$ and -1 otherwise
 - 10 rounds

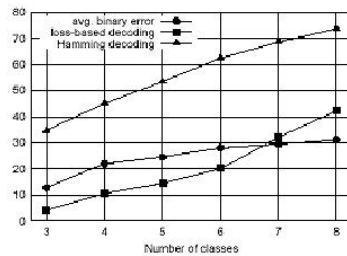


21

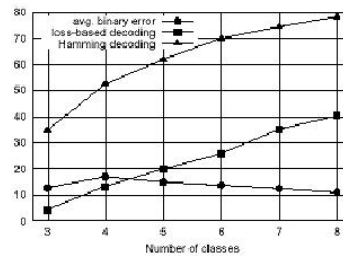
Comparisons:

- a) Hamming vs. loss-based decoding
- b) simple binary problems vs. robust matrix

complete code

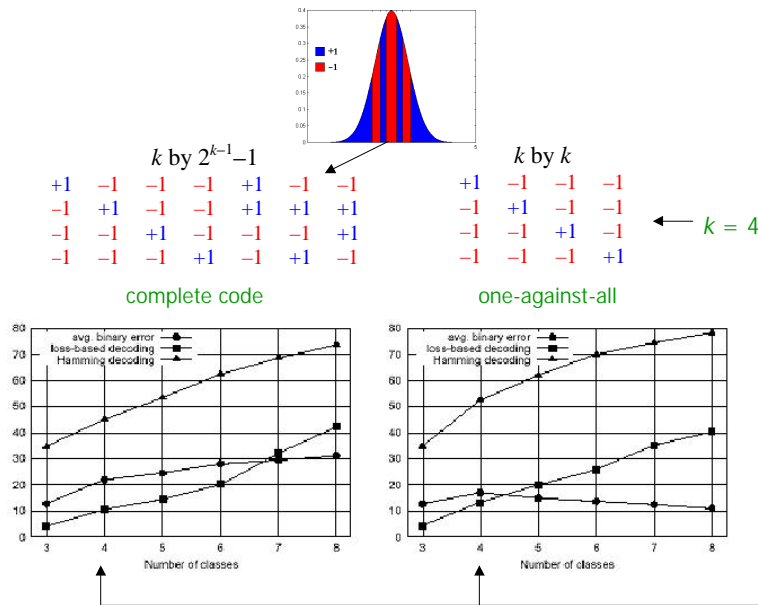


one-against-all



AdaBoost used in both simulations

22



23

Comparison: different output codes

- Experiments with UCI databases
- SVM: polynomial kernels of degree 4
- AdaBoost: decision stumps for base hypotheses
- 5 output codes: *one-against-all*, *complete*, *all-pairs*, *dense* and *sparse*
 - design of *dense* and *sparse*: try 10,000 random codes, pick a code with high r

| Problem | #Examples | | #Attributes | #Classes |
|--------------|-----------|------|-------------|----------|
| | Train | Test | | |
| dermatology | 366 | - | 34 | 6 |
| satimage | 4435 | 2000 | 36 | 6 |
| glass | 214 | - | 9 | 7 |
| segmentation | 2310 | - | 19 | 7 |
| ecoli | 336 | - | 8 | 8 |
| pendigits | 7494 | 3498 | 16 | 10 |
| yeast | 1484 | - | 8 | 10 |
| vowel | 528 | 462 | 10 | 11 |
| soybean | 307 | 376 | 35 | 19 |
| thyroid | 9172 | - | 29 | 20 |
| audiology | 226 | - | 69 | 24 |
| isolet | 6238 | 1559 | 617 | 26 |
| letter | 16000 | 4000 | 16 | 26 |

Table 1: Description of the datasets used in the experiments.

24

| Hamming Decoding | | | | | |
|------------------|------------|----------|-----------|-------|--------|
| Problem | One-vs-all | Complete | All-Pairs | Dense | Sparse |
| dermatology | 5.0 | 4.2 | 3.1 | 3.9 | 3.6 |
| satimage | 14.9 | 12.3 | 11.7 | 12.3 | 13.2 |
| glass | 31.0 | 31.0 | 28.6 | 28.6 | 27.1 |
| segmentation | 0.0 | 0.1 | 0.0 | 0.1 | 0.1 |
| ecoli | 21.5 | 18.5 | 19.1 | 17.6 | 19.7 |
| pendigits | 8.9 | 8.6 | 3.0 | 9.3 | 6.2 |
| yeast | 44.7 | 41.9 | 42.5 | 43.9 | 49.5 |
| vowel | 67.3 | 59.3 | 50.2 | 62.6 | 54.5 |
| soybean | 8.2 | - | 9.0 | 5.6 | 8.0 |
| thyroid | 7.8 | - | - | 12.3 | 8.1 |
| audiology | 26.9 | - | 23.1 | 23.1 | 23.1 |
| isolet | 9.2 | - | - | 10.8 | 10.1 |
| letter | 27.7 | - | 7.8 | 30.9 | 27.1 |

| Loss-based Decoding (L_1) | | | | | |
|-------------------------------|------------|----------|-----------|-------|--------|
| Problem | One-vs-all | Complete | All-Pairs | Dense | Sparse |
| dermatology | 4.2 | 4.2 | 3.1 | 3.9 | 3.6 |
| satimage | 12.1 | 12.4 | 11.2 | 11.9 | 11.9 |
| glass | 26.7 | 31.0 | 27.1 | 27.1 | 26.2 |
| segmentation | 0.0 | 0.1 | 0.0 | 0.1 | 0.7 |
| ecoli | 17.3 | 17.6 | 18.8 | 18.5 | 19.1 |
| pendigits | 4.6 | 8.6 | 2.9 | 8.8 | 6.8 |
| yeast | 41.6 | 42.0 | 42.6 | 43.2 | 49.8 |
| vowel | 56.9 | 59.1 | 50.9 | 61.9 | 54.1 |
| soybean | 7.2 | - | 8.8 | 4.8 | 8.2 |
| thyroid | 6.5 | - | - | 12.0 | 8.0 |
| audiology | 19.2 | - | 23.1 | 19.2 | 23.1 |
| isolet | 5.3 | - | - | 10.1 | 9.8 |
| letter | 14.6 | - | 7.4 | 29.0 | 26.6 |

| Loss-based Decoding (Exp.) | | | | | |
|----------------------------|------------|----------|-----------|-------|--------|
| Problem | One-vs-all | Complete | All-Pairs | Dense | Sparse |
| dermatology | 4.2 | 3.9 | 3.1 | 4.2 | 3.1 |
| satimage | 12.1 | 12.3 | 11.4 | 12.0 | 12.0 |
| glass | 26.7 | 28.6 | 27.6 | 25.2 | 29.0 |
| segmentation | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ecoli | 15.8 | 16.4 | 18.2 | 17.0 | 17.9 |
| pendigits | 4.6 | 7.2 | 2.9 | 8.1 | 4.8 |
| yeast | 41.6 | 42.1 | 42.3 | 43.0 | 49.3 |
| vowel | 56.9 | 54.1 | 51.7 | 60.0 | 49.8 |
| soybean | 7.2 | - | 8.8 | 4.8 | 5.6 |
| thyroid | 6.5 | - | - | 11.4 | 7.2 |
| audiology | 19.2 | - | 23.1 | 19.2 | 19.2 |
| isolet | 5.3 | - | - | 9.4 | 9.7 |
| letter | 14.6 | - | 7.1 | 28.3 | 22.3 |

Table 2: Results of experiments with output codes with datasets from the UCI repository using AdaBoost as the base binary learner. For each problem five output codes were used and then evaluated (see text) with three decoding methods: Hamming decoding, loss-based decoding using AdaBoost with randomized predictions (denoted L_1), and loss-based decoding using the exponential loss function (denoted $Exp.$).

| Hamming Decoding | | | | | |
|------------------|------------|----------|-----------|-------|--------|
| Problem | One-vs-all | Complete | All-Pairs | Dense | Sparse |
| dermatology | 4.2 | 3.6 | 3.1 | 3.6 | 2.5 |
| satimage | 40.9 | 14.3 | 50.4 | 15.0 | 27.4 |
| glass | 37.6 | 34.3 | 29.5 | 34.8 | 32.4 |
| ecoli | 15.8 | 14.2 | 13.9 | 15.2 | 14.2 |
| pendigits | 3.9 | 2.0 | 26.2 | 2.5 | 2.6 |
| yeast | 73.9 | 42.4 | 40.8 | 42.5 | 48.1 |
| vowel | 60.4 | 53.0 | 39.2 | 53.5 | 50.2 |
| soybean | 20.5 | - | 9.6 | 9.0 | 9.0 |

| Loss-based Decoding | | | | | |
|---------------------|------------|----------|-----------|-------|--------|
| Problem | One-vs-all | Complete | All-Pairs | Dense | Sparse |
| dermatology | 3.3 | 3.6 | 3.6 | 3.9 | 3.1 |
| satimage | 40.9 | 13.9 | 27.8 | 14.3 | 13.3 |
| glass | 38.6 | 34.8 | 31.0 | 34.8 | 32.4 |
| ecoli | 16.1 | 13.6 | 13.3 | 14.8 | 14.8 |
| pendigits | 2.5 | 1.9 | 3.1 | 2.1 | 2.7 |
| yeast | 72.9 | 40.5 | 40.9 | 39.7 | 47.2 |
| vowel | 50.9 | 51.3 | 39.0 | 51.7 | 47.0 |
| soybean | 21.0 | - | 10.4 | 8.8 | 9.0 |

Table 3: Results of experiments with output codes with datasets from the UCI repository using the support-vector machine (SVM) algorithm as the base binary learner. For each problem five different classes of output codes were tested were used and then evaluated with Hamming decoding and the appropriate loss-based decoding for SVM.

25



Show error obtained with Hamming minus error with loss-based decoding
 Negative height indicates loss-based outperformed Hamming decoding

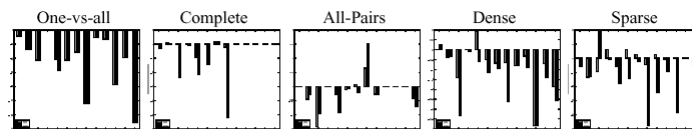


Figure 4: Comparison of the test error using Hamming decoding and loss-based decoding when the binary learners are trained using AdaBoost. Two loss functions for decoding are plotted: the exponential loss ("Exp", in black) and $1/(1 + e^{2yf(x)})$ when using AdaBoost with randomized predictions (" L_1 ", in gray).

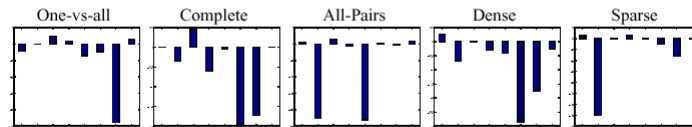


Figure 5: Comparison of the test error using Hamming decoding and loss-based decoding when the binary learner is support vector machines.

26



Entry (r, c) shows error of row r classifier minus error of column c classifier
 Positive height indicates classifier c outperformed classifier r

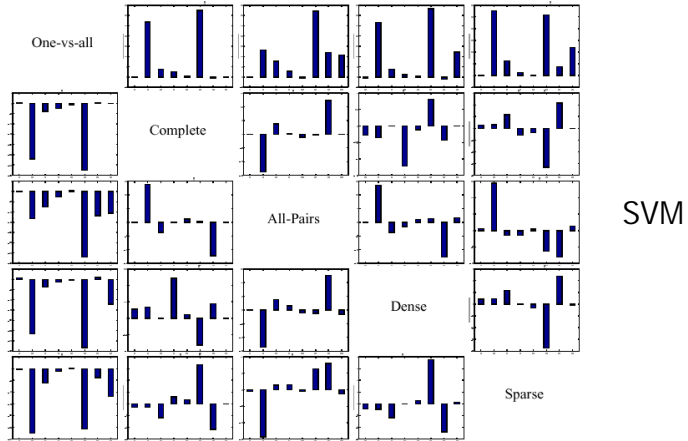
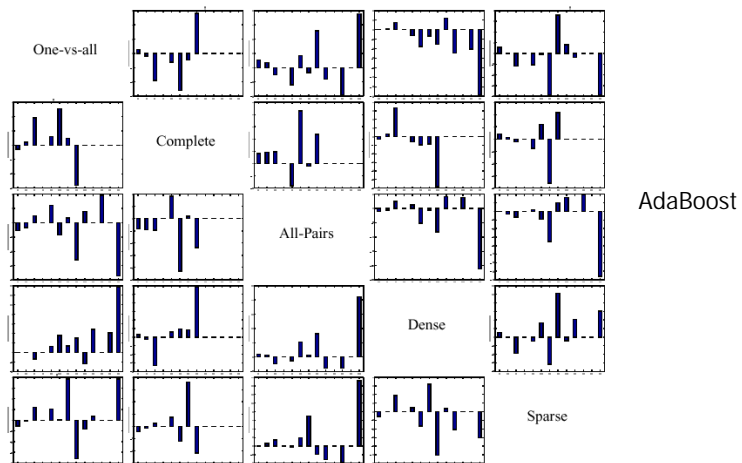


Figure 6: The difference between the test errors for pairs of error correcting matrices using support vector machines as the binary learners.

27



Entry (r, c) shows error of row r classifier minus error of column c classifier
 Positive height indicates classifier c outperformed classifier r



28

Conclusions

- Bounds give insight about the tradeoffs but can be of limited use in practice
- Experiments show that in most cases:
 - Loss-based is better than Hamming decoding
 - One-against-all is outperformed (SVM)
- Choosing / designing the coding matrix is an open problem and the best approach is possibly task-dependent
 - (Crammer & Singer, 2000)

29

1. Proof of training error bound

We want to relate the error ϵ of the final (multi-class) classifier to the average loss ϵ of the binary classifiers. Notice that ϵ is the fraction of examples wrongly classified, while ϵ is an average loss. The key observation to relate them is that whenever an example x_i leads to an error (increases ϵ), its contribution to the total loss (which average is ϵ) is at least $\rho L(0)$. The total loss T of the binary classifiers is $T = \sum_{i=1}^m \sum_{s=1}^k L(M(y_i, s) f_s(x_i))$, and by definition $\epsilon = T/ml$. Let $d(\mathbf{M}(r), \mathbf{f}(x)) = \sum_{s=1}^k L(M(r, s) f_s(x))$, such that $T = \sum_{i=1}^m d(\mathbf{M}(y_i), \mathbf{f}(x_i))$. Define E as the set of examples wrongly classified, with $\epsilon = |E|/m$. From now on, let us consider $x \in E$. In this case there is $r \neq y$ such that $d(\mathbf{M}(r), \mathbf{f}(x)) \leq d(\mathbf{M}(y), \mathbf{f}(x))$. Notice that when we compute ϵ (and T consequently) we always use the loss correspondent to the correct row $d(\mathbf{M}(y), \mathbf{f}(x))$ even if $x \in E$. Let us use y and r to split the columns of M in three subsets. Let

$$S_\Delta = \{s : M(r, s) \neq M(y, s) \wedge M(r, s) \neq 0 \wedge M(y, s) \neq 0\}$$

be the set of columns of M in which rows r and y differ and are both non-zero. Let

$$S_0 = \{s : M(r, s) = 0 \vee M(y, s) = 0\}$$

be the set of columns in which either row r or y is zero. Define $z_r^* = M(r, s) f_s(x)$ and $z_y^* = M(y, s) f_s(x)$. We know that $d(\mathbf{M}(r), \mathbf{f}(x)) \leq d(\mathbf{M}(y), \mathbf{f}(x))$ and we assumed $\frac{d(\mathbf{M}(r), \mathbf{f}(x))}{2} \geq L(0)$, so, when an error occurs, $d(\mathbf{M}(y), \mathbf{f}(x))$ can be lower bounded as follows.

$$\begin{aligned} d(\mathbf{M}(y), \mathbf{f}(x)) &\geq \sum_{s \in S_\Delta \cup S_0} L(z_y^*) \\ &\text{because } S_\Delta \cup S_0 \text{ is a subset and loss is non-negative} \\ &\geq \frac{1}{2} \sum_{s \in S_\Delta \cup S_0} L(z_r^*) + \frac{1}{2} \sum_{s \in S_\Delta \cup S_0} L(z_y^*) \\ &\text{because } \sum_{s \in S_\Delta \cup S_0} L(z_r^*) \leq \sum_{s \in S_\Delta \cup S_0} L(z_y^*) \\ &\geq \frac{1}{2} \sum_{s \in S_\Delta \cup S_0} (L(z_r^*) + L(z_y^*)) \\ &\text{by linearity} \\ &= \sum_{s \in S_\Delta} (L(z_r^*) + L(z_y^*)) + \frac{1}{2} \sum_{s \in S_0} (L(z_r^*) + L(z_y^*)) \\ &\text{because sets are disjoint} \\ &\geq (|S_\Delta| + \frac{|S_0|}{2}) L(0) = \Delta(M(r), M(y)) L(0) \geq \rho L(0) \end{aligned}$$

Given that we know each error "costs" a loss of at least $\rho L(0)$ and the total loss is T , we can find the maximum number of errors as follows.

$$T = ml\epsilon \geq \sum_{x \in E} d(\mathbf{M}(y), \mathbf{f}(x)) \geq |E| \rho L(0)$$

$$\epsilon = \frac{|E|}{m} \leq \frac{L\epsilon}{\rho L(0)} \tag{1.1}$$

30