1 PECOC reduction

Let \( h: \mathcal{X} \rightarrow \mathbb{R}^d \) be a possibly randomized target function we wish to learn. In multiclass problems, we assume that \( h(x) \in \{e_1, \ldots, e_d\} \), so \( \mathbb{E}h(x) \in \Delta^d \); in multilabel problems, we assume that \( h(x) \in \{0, 1\}^d \), so \( \mathbb{E}h(x) \in [0,1]^d \). We are interested in reducing the problem of learning the vector-valued function \( h \) to the problem of learning a collection of scalar-valued functions.

Let \( C = [c_1 \ldots c_m]^\top \in \mathbb{R}^{m \times d} \) be our coding matrix (so each \( c_i \in \mathbb{R}^d \)), for some \( m \leq d \). We create \( m \) regression problems; the label of \( x \in \mathcal{X} \) in the \( i \)th problem is \( \langle c_i, h(x) \rangle \). The regressor \( b_i^* \) with least mean-squared-error for the \( i \)th problem is defined by \( b_i^*(x) = \mathbb{E}\langle c_i, h(x) \rangle = \langle c_i, \mathbb{E}h(x) \rangle \). Let \( b^* : \mathcal{X} \rightarrow \mathbb{R}^m \) be the vector-valued regressor defined by \( b^*(x) = [b_1^*(x), \ldots, b_m^*(x)]^\top = CEh(x) \).

2 Recovery for invertible \( C \)

If \( C \in [-1,+1]^{d \times d} \) is invertible, then we can recover \( \mathbb{E}h(x) \) from \( b^*(x) \) by taking

\[
C^{-1}b^*(x) = C^{-1}CEh(x) = \mathbb{E}h(x).
\]

Given regressors \( b_1, \ldots, b_d \), we are interested in the performance of the vector-valued regressor \( C^{-1}b \) defined by \( C^{-1}b(x) = C^{-1}[b_1(x), \ldots, b_d(x)]^\top \).

2.1 Generic regret statement for invertible \( C \)

Consider any regressor \( b: \mathcal{X} \rightarrow \mathbb{R}^d \). Then the regret of \( C^{-1}b \) is

\[
\mathbb{E}_x \|C^{-1}b(x) - \mathbb{E}h(x)\|^2 = \mathbb{E}_x \|C^{-1}(b(x) - b^*(x))\|^2.
\]

We would like to bound this quantity in terms of the average (scaled) regret of the \( b_i \), i.e. in terms of

\[
\bar{r} = \frac{1}{d} \sum_{i=1}^d r_i,
\]

where

\[
r_i = \mathbb{E}_x \left( \frac{b_i(x) - b_i^*(x)}{B_i} \right)^2 \quad \text{and} \quad B_i = \max_x \langle c_i, h(x) \rangle - \min_x \langle c_i, h(x) \rangle.
\]
2.2 One-against-all reduction

If $C = C^{-1} = I_d$ is the identity matrix, then the regret of $C^{-1}b$ is

$$E_x ||C^{-1}(b(x) - b^*(x))||_2^2 = E_x ||b(x) - b^*(x)||_2^2.$$ 

If $B_i \leq B$ for all $i$, then the regret is bounded by $dB^2 r$. If $||h(x)||_\infty \leq 1$ (e.g. in multiclass and multilabel problems), then $B \leq 1$.

2.3 Hadamard reduction

Let $C \in \{\pm 1\}^{d \times d}$ be the Hadamard matrix. The singular values of $C$ are all $\sqrt{d}$, so $||C^{-1}||_2^2 = 1/d$. Therefore the regret of $C^{-1}b$ is

$$E_x ||C^{-1}(b(x) - b^*(x))||_2^2 = ||C^{-1}||_2^2 E_x ||b(x) - b^*(x)||_2^2 = \frac{1}{d} E_x ||b(x) - b^*(x)||_2^2.$$ 

If $B_i \leq B$ for all $i$, then the regret is bounded by $B^2 r$. If $||h(x)||_1 \leq k$ (e.g. in $k$-sparse multilabel problems), then $B \leq 2k$.

2.4 Ridge regression

Suppose $X = \mathbb{R}^p$ and we use ridge regression to learn the $b_i = (\hat{E}[xx^\top] + \lambda I_p)^{-1}\hat{E}[x(h(x), c_i)]$. Then $b = (\hat{E}[xx^\top] + \lambda I_p)^{-1}\hat{E}[xh(x)^\top]C^\top$. The prediction on a new point $x_{new}$ is

$$C^{-1}b^\top x_{new} = C^{-1}C\hat{E}[xh(x)^\top]^\top(\hat{E}[xx^\top] + \lambda I_p)^{-1}x_{new} = \hat{E}[xh(x)^\top]^\top(\hat{E}[xx^\top] + \lambda I_p)^{-1}x_{new}.$$ 

Thus the predictor is the same as the one learned using ridge regression without the reduction. An implication of this is that the Hadamard problem is $d$ times as hard for ridge regression to solve as the one-against-all problem. That is, 

$$E_x \|b_{\text{Hadamard}}(x) - C\hat{E}h(x)\|_2^2 = d E_x \|b_{\text{OAA}}(x) - \hat{E}h(x)\|_2^2$$ 

where $C$ is the Hadamard matrix, where the same ridge regression estimator is used to learn both $b_{\text{Hadamard}}$ and $b_{\text{OAA}}$.

3 Multiclass experiments with regression trees

Are the Hadamard problems significantly harder than the one-against-all problems for other estimators? We evaluated the one-against-all and Hadamard reductions on eight multiclass tasks from the UCI repository. The base learner used is a simple regression tree learner (classregtree from the MATLAB statistics toolbox). We are interested in classification error, so we map a vector-valued prediction $y \in \mathbb{R}^d$ to the class $\arg \max_i y_i \in \{1, \ldots, d\}$. We report the test error in the following table (when a testing set was not given, we chose 10 random $(2/3, 1/3)$ splits of the data to form training and testing sets, and then averaged the resulting test errors).

<table>
<thead>
<tr>
<th>Data set</th>
<th>Number of classes</th>
<th>One-against-all</th>
<th>Hadamard</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecoli</td>
<td>8</td>
<td>0.0985</td>
<td>0.0517</td>
</tr>
<tr>
<td>glass</td>
<td>6</td>
<td>0.3874</td>
<td>0.3462</td>
</tr>
<tr>
<td>pendigits</td>
<td>10</td>
<td>0.0985</td>
<td>0.0517</td>
</tr>
<tr>
<td>satimage</td>
<td>6</td>
<td>0.1679</td>
<td>0.1376</td>
</tr>
<tr>
<td>soybean</td>
<td>19</td>
<td>0.6580</td>
<td>0.5993</td>
</tr>
<tr>
<td>splice</td>
<td>3</td>
<td><strong>0.0642</strong></td>
<td>0.0699</td>
</tr>
<tr>
<td>vowel</td>
<td>11</td>
<td>0.6356</td>
<td>0.5780</td>
</tr>
<tr>
<td>yeast</td>
<td>10</td>
<td>0.4893</td>
<td>0.4479</td>
</tr>
</tbody>
</table>
Acknowledgements

This note grew out of discussions with Sham M. Kakade, John Langford, and Tong Zhang.

References