REDUCING MULTICLASS COST SENSITIVE CLASSIFICATION TO BINARY COST SENSITIVE CLASSIFICATION

1. Goals

The goal here is to find a good reduction from multiclass cost sensitive classification to binary cost sensitive classification. These two problems are summarized next, then a method is discussed.

1.1. Multiclass Cost Sensitive Classification. In multiclass cost sensitive classification, labeled examples have the form $(x, c_1, c_2, ..., c_k)$ where $x \in X$ is an arbitrary feature and $c_l \in [0, \infty)$ is the "cost" of choosing label l. Since the value of a good classifier is independent of a global offset, we'll assume that there is at least one label l which satisfies $c_l = 0$.

To solve this problem, we must find a hypothesis $h : X \to \{1, ..., k\}$ which produces low cost labels given the input feature. With respect to some measure D producing these examples we want to do the optimization:

$$\min_h E_{(x,c_1,c_2,...,c_k)\sim D} c_{h(x)}$$

Here the optimization (or heuristic optimization) is over some unspecified and algorithm dependent set of classifiers.

1.2. Binary Cost Sensitive Classification. In binary cost sensitive classification, labeled examples have the form (x, y, i) where $x \in X$ is an arbitrary feature, $y \in \{0, 1\}$ is a binary label and $i \in [0, \infty)$ is the "importance" of classifying correctly. To solve this problem, we must find a hypothesis $h : X \to \{0, 1\}$ which is correct on high importance examples. With respect to some measure D producing these examples, we want to solve the optimization:

$$\min_h E_{x,y,i\sim D} i I(h(x) \neq y)$$

Again, this (heuristic) optimization is according to some learning algorithm.

2. A Reduction

Let's attempt a reduction of Multiclass Cost Sensitive Classification to a single instance of Binary cost sensitive classification.

The method we use depends on a free parameter w_x for every value of x. This parameter, roughly speaking, controls how optimistic we are about the effectiveness of our classifier. If we believe the classifier can choose the correct label from a large set, then a small w_x makes sense.

We map the example $(x, c_1, c_2, ..., c_k)$ to a binary example with feature $x_l = (x, l)$, label $y_l = \text{sign}(w - c_l)$ (where w is a free parameter we worry about specifying later), and an importance i_l specified later. In order to use a binary classifier for multiclass classification, we must specify a method for producing the multiclass label. The following simple procedure seems reasonable.

Predict(h,x)

(1) $P = \{\}$
(2) for every $l = 1$ to k do
  (a) if $h(x) = 1$ then $P \leftarrow P \cup \{l\}$
(3) if $|P| > 0$ then return a uniform draw from P
(4) else return a uniform draw from $\{1, ..., k\}$

This procedure implicitly specifies how errors in the binary classification transform into errors for the multiclass classification. This transformation, in turn, can tell us what the good values for the importance i in the binary classification problem are.

One fundamental quantity for any example $(x, c_1, ..., c_k)$ and weight w_x is the size of the set $n_w = |\{l : c_l < w\}|$. Another is the expected cost given that no errors occur:

$$c_w = \frac{1}{n_w} \sum_{l:c_l<w} c_l$$

. Notice that in this notation $c_\infty = \frac{1}{k}\sum_{l=1}^{k} c_l$ is the average cost.

There are two basic failure modes: either we have a false positive or a false negative. We want to analyze these failure modes in terms of the parameters c_w (cost when everything works out) and c_l (the cost of the label l that we err on).

False positive: For some label l which should have $h(x, l) = 0$ we instead get $h(x, l) = 1$. In this case, the expected cost is:

$$\frac{n_w c_w + c_l}{n_w + 1} = c_w + \frac{c_l - c_w}{n_w + 1}$$

This analysis is misleading because it only deals with the one error case. If we have n_w false negatives, then the cost of a false positive might be as high as $c_l - c_\infty$ (notice that this might be negative). A simple upper bound which always holds is $c_l - c_w$.

False, negative: For some l which should have $h(x, l) = 1$ we instead get $h(x, l) = 0$. In this case, the expected cost breaks down into two possibilities.

(1) $n_w \geq 2$. The expected cost is $\frac{n_w c_w - c_l}{n_w - 1} = c_w + \frac{c_w - c_l}{n_w - 1}$ ($c_w - c_l$ can be positive or negative: some false negatives can improve performance).
(2) $n_w = 1$. The expected cost is $c_w + c_\infty - c_l$ (note that $c_l = c_w$ in this case).

When multiple false negatives occur the "bad failure mode" of $n_w = 1$ can occur even when there are many possible "right" answers, $n_w > 1$. One efficient way to account for this effect is to divide the bad failure mode across all of the "right" answers.

A good importance weight is proportional to the deviation of our final cost from c_w. When our example is of the form $x_l = (x, l)$, label $y_l = 0$ we set $i_l = c_l - c_w$. When our example is of the form feature $= x_l = (x, l)$, label $= y_l = 1$ we set $i_l = \max\left\{\frac{c_\infty - c_l}{n_w}, c_w - c_l\right\}$. We can prove that these importances are sufficient to upper bound the loss of our multiclass classifier.

Lemma 2.1. *The importances of errors upper bounds the loss:*

$$\forall (x, c_1, c_2, ..., c_k),\ \ h : X \times Y \to \{0, 1\} \quad \sum_{l=1}^{k} i_l I(h(x, l) \neq y_l) \geq E_{l \sim Predict(h,x)} c_l - c_w$$

Proof. Each side is 0 when there are no errors so the inequality is satisfied so we need only consider the case of errors.

There are two error cases to consider. In the first, there are no positives, and the expected cost is c_∞ and the difference is $E_{l\sim\text{Predict}(h,x)}c_l - c_w = c_\infty - c_w$. In this case, we must have "false negative" errors for every example which must be positive implying:

$$\sum_{l=1}^{k} i_l I(h(x,l) \neq y_l) \geq \sum_{l:y_l=1} \frac{c_\infty - c_l}{n_w} = c_\infty - c_w$$

The other case is when there are l for which $h(x,l) = 1$. In this case, $E_{l\sim\text{Predict}(h,x)}c_l - c_w = \frac{1}{\{l:h(x,l)=1\}}\sum_{l:h(x,l)=1} c_l - c_w$. In this setting, we can have false positives (where $y_l = 0$ but $h(x,l) = 1$). Each false positive has a term in the left-side sum which dominates it's contribution to the expectation on the right side: $i_l = c_l - c_w \geq \frac{c_l - c_w}{\{l:h(x,l)=1\}}$. All that remains is proving the following when there is at least one label l for which $h(x,l) = 1$ and $y_l = 1$.

$$\sum_{l:h(x,l)=0\wedge y_l=1} i_l \geq \frac{1}{\{l : h(x,l) = 1\}} \sum_{l:h(x,l)=1\wedge y_l=1} c_l - c_w$$

The basic argument is that $\frac{1}{\{l:h(x,l)=1\}}\sum_{l:h(x,l)=1\wedge y_l=1} c_l - c_w$ changes by less than the importance for every error. Changes are maximized when $\{l : h(x,l) = 1\} = 1$, so we will actually prove:

$$\sum_{l:h(x,l)=0\wedge y_l=1} i_l \geq \sum_{l:h(x,l)=1\wedge y_l=1} c_l - c_w$$

If $c_l > c_w$, then a false negative error actually decreases the right side sum (i.e. lowers this portion of the expected cost). Since $i \geq 0$, we are done. When $c_l > c_w$, we know that $i_l \geq c_w - c_l$ which upper bounds the change in the right-hand side sum. $\square$

2.1. **Practical Use.** To put this to practical use, we must handle two things. One of these is coping with the lack of independence (we produce one example for every label) associated with the map. This is can be handled by drawing from the weighted distribution of importances over the labels on each example.

The other issue is w_x. It is not clear in advance how optimistic we can reasonably be that the classifier can distinguish good classes. Updating methods similar to the ones that Naoki is playing with seem like a plausibly right answer.