Classification with Unbalanced Costs

Evan Heus and Vasco Villas-Boas

March 15, 2023

1 Introduction to Classification

Suppose we have a space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ where for $z = (x, y) \in \mathcal{Z}$, x is a set of characteristics and y is an associated class label for z. Moreover, suppose we have a probability distribution $(x, y) \sim P_{x,y}$ over the space[1]. We'll define notation for the remaining probability distributions and densities now as well:

$$
\begin{aligned}
&x \sim g_x \\
&x|y \sim g_{x|y} \\
&y|x \sim P_{y|x}(y|x) \\
&y \sim P_y
\end{aligned}
$$

In classification, we aim to construct a decision rule $f : \mathcal{X} \mapsto \mathcal{Y}$ that looks at the characteristics x of some element z and assigns it to a class y.

For a decision rule f, we define a loss function $l : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ so that $l(y, f(x))$ reports the penalty f receives for assigning class label $f(x)$ to some characteristics x instead of the correct label y. An interesting problem[2], and that of classification, is to find a $f \in \mathcal{F}$, where $\mathcal{F}$ is the global space of decision functions, that minimizes the expected loss of f according to $P_{x,y}$ and l. We call the quantity, $R(f, l) := \mathbb{E}_P[l(y, f(x))]$, the *risk* of the function f according to l. Mathematically, the optimal f^* satisfies,

$$
\begin{aligned}
f^* &:= \text{argmin}_{f\in\mathcal{F}} R(f, l) \\
&= \text{argmin}_{f\in\mathcal{F}} \mathbb{E}_P[l(y, f(x))] \qquad (1)
\end{aligned}
$$

1.1 Loss Function: 0-1 Loss

A typical loss function chosen is the 0-1 Loss function. In this case, $l(y, f(x)) = 1_{\{y \neq f(x)\}}$. We assign no penalty if f correctly classifies x and a penalty of 1 if f incorrectly classifies x. Mathematically,

[1] In the real world, this distribution $P_{x,y}$ is unknown and must be estimated. For now, for the next few sections, let's assume it is known to the statistician.

[2] There are many other problems that we could choose to solve. For example, we could pick the decision function that minimizes the 75th percentile of the loss function according to $P_{x,y}$. Minimizing the expected loss is arbitrary but nice in the sense we're finding the f that means we'll do the best on average according to our criteria l.

$$\begin{aligned} f^* &= \text{argmin}_{f\in\mathcal{F}} R(f,l) \\ &= \text{argmin}_{f\in\mathcal{F}} \mathbb{E}_P[l(y,f(x))] \\ &= \text{argmin}_{f\in\mathcal{F}} \mathbb{E}_P[1_{\{y\neq f(x)\}}] \\ &= \text{argmin}_{f\in\mathcal{F}} P_{x,y}(y\neq f(x)) \\ &= \text{argmin}_{f\in\mathcal{F}} \int_{x\in\mathcal{X}} \text{dx} \sum_{y\in\mathcal{Y}, y\neq f(x)} P_{x,y}(x,y) \\ &= \text{argmin}_{f\in\mathcal{F}} \int_{x\in\mathcal{X}} g_x(x)\text{dx} \sum_{y\in\mathcal{Y}, y\neq f(x)} P_{y|x}(y|x) \\ &= \text{argmin}_{f\in\mathcal{F}} \int_{x\in\mathcal{X}} g_x(x)\text{dx}[1-P_{y|x}(f(x)|x)] \end{aligned}$$

To minimize the argument in the last line, for any x in the integral, we take we pick $f(x)$ so as to maximize $P_{y|x}(f(x)|x)$. As a result, f^* can be defined as

$$f^*(x) = \text{argmax}_{y\in\mathcal{Y}} P_{y|x}(y|x) \tag{2}$$

This makes a lot of sense, for characteristics x, the optimal classifier assigns x to the class y that has the greatest likelihood given x.

1.1.1 Binary Labeling

Restrict $\mathcal{Y} = \{0,1\}$ so that now we're trying to find the risk minimizing decision rule f^* for binary class labels according to 0-1 loss. We know that:

$$\begin{aligned} f^*(x) &= \text{argmax}_{y\in\mathcal{Y}} P_{y|x}(y|x) \\ &= \text{argmax}_{y\in\{0,1\}} P_{y|x}(y|x) \\ &= 1_{\{P_{y|x}(y=1|x)\geq P_{y|x}(y=0|x)\}} \\ &= 1_{\{\frac{P_{y|x}(y=1|x)}{P_{y|x}(y=0|x)}\geq 1\}} \\ &= 1_{\{\frac{P_{y|x}(y=1|x)}{1-P_{y|x}(y=1|x)}\geq 1\}} \qquad (3) \\ &= 1_{\{P_{y|x}(y=1|x)\geq 1/2\}} \qquad (4) \end{aligned}$$

In other words, in the case of binary classifier, the optimal classifier according to 0-1 loss picks a class if its likelihood is greater than 1/2 given the characteristics x. This result makes sense in many ways, if a class is more likely than not, we pick it. There are other nice ways to think about this classifier starting with the third line in the above derivation and using Bayes' rule:

$$\begin{aligned} f^*(x) &= 1_{\{P_{y|x}(y=1|x)\geq P_{y|x}(y=0|x)\}} \\ &= 1_{\{\frac{g_{x|y}(x|y=1)P(y=1)}{g_{x|y=1}(x|y=1)P(y=1)+g_{x|y}(x|y=0)P(y=0)}\geq \frac{g_{x|y}(x|y=0)P(y=0)}{g_{x|y=1}(x|y=1)P(y=1)+g_{x|y}(x|y=0)P(y=0)}\}} \\ &= 1_{\{g_{x|y}(x|y=1)P(y=1)\geq g_{x|y}(x|y=0)P(y=0)\}} \qquad (5) \end{aligned}$$

To understand Figure 1 below, suppose that $\mathcal{X} = \mathbb{R}$ so that our characteristic is a one dimensional value. In both plots, in red is the graph $g_{x|y=0} * P(y = 0)$ and in blue is the graph $g_{x|y=1} * P(y = 1)$. On the left, we say that the classes $Y = 1$ and $Y = 0$ have equal priors. Meanwhile on the right, we say that the prior $P(Y = 1) = 0.25$ and the prior $P(Y = 1) = 0.75$.

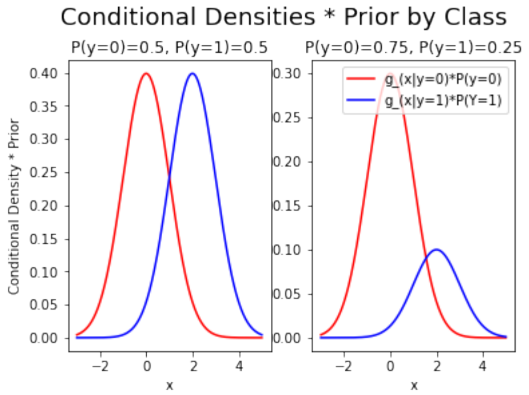

Figure 1. Interpreting 0-1 Loss Decision Rule

The optimal decision function f^*, as seen in equation 5, picks label 1 for x when the "density * prior" for label 1 is greater than the "density * prior" for label 0 (ie., the blue graph lies above the red graph at the given x).

1.1.2 An Example

Suppose that we're trying to predict if an individual will vote republican ($y = 1$) or democrat ($y = 0$). We observe x_1, the income of the individual in thousands, and we don't observe x_2, their parents' combined income in thousands when they were a child. Further, assume that $y = 1_{\{x_1 - x_2 \geq 50\}}$; in other words, if the individual makes at least $\$50,000$ more than their parents did when they were a kid[3], they vote republican, else democrat. Also, assume that x_1, x_2 are independent, $x_1 \sim \mathcal{N}(100, 36)$, and $x_2 \sim \mathcal{N}(110, 25)$.

We want to find the best decision rule f^* according to 0-1 loss to predict the voting decision of a person given their income x_1.

By equation 4, if we can find the quantity $P_{y|x_1}(y = 1|x_1)$, we will have an expression for $f^*(x_1)$.

$$
\begin{aligned}
P_{y|x}(y = 1|x_1) &= P(x_1 - x_2 \geq 50|x_1) \\
&= P(x_1 - 50 \geq x_2|x_1) \\
&= P(x_2 \leq x_1 - 50|x_1) \\
&= P(\mathcal{N}(110, 25) \leq x_1 - 50|x_1) \\
&= P(\mathcal{N}(0, 1) \leq \frac{x_1 - 160}{5}|x_1) \\
&= \Phi(\frac{x_1 - 160}{5}|x_1)
\end{aligned}
$$

where $\Phi(\cdot)$ is the CDF of $\mathcal{N}(0, 1)$. Thus, the risk minimizing decision rule according to 0-1 loss for this problem is $f^*(x_1) = 1_{\{\Phi(\frac{x_1-160}{5}|x_1)\geq 1/2\}}$. We can simplify further: note that the event $\Phi(\frac{x_1-160}{5}|x_1) \geq 1/2$ happens precisely when $x_1 \geq 160$. Thus, we can simplify to say $f^*(x_1) = 1_{\{x_1 \geq 160\}}$. We suspect the reader could come up with the optimal decision rule without having done the derivation if they take a step back.

[3] Ignore inflation please!

2 Unbalanced Costs in Decisions

2.1 *Motivation for Unbalanced Costs*

In the 0-1 loss framework, we value all mis-classifications made by a decision rule f equally. That's not a good idea in some settings because certain errors in decisions may be much more costly than others. For example, consider the ternary case where we aim to decide whether a hospital patient has prostate cancer ($y = 2$), has colon cancer ($y = 1$), or has nothing ($y = 0$) based on characteristics x. If we predict $f(x) = 0$ when in fact $y = 1$, we've made a serious error because we've decided a patient was healthy and let them walk free when in fact they have a deadly sickness. If we say $f(x) = 1$ when in fact $y = 0$, we've made a less bad error because we will keep them under our supervision even though they're fine. If we say $f(x) = 2$ when in fact $y = 1$, the patient is unhealthy and we mis-attributed the reason but at least they'll stay under our supervision. We will expand on defining loss functions and give more examples in Section 3.

2.2 *Formulating the Loss Function for the Unbalanced Cost Model*

Suppose that we can enumerate the set $\mathcal{Y} = \{1, ..., n\}$ for some $n \in \mathbb{Z}^+$. Recall that $l(y, f(x))$ reports the penalty a decision rule f receives for assigning class label $f(x)$ to some characteristics x instead of the correct label y. We can show the loss function in tabular format:

$l(y, f(x))$	$f(x) = 1$	$f(x) = 2$	...	$f(x) = n$
$y = 1$	$l(1, 1)$	$l(1, 2)$	...	$l(1, n)$
$y = 2$	$l(2, 1)$	$l(2, 2)$	...	$l(2, n)$
$y = 3$	$l(3, 1)$	$l(3, 2)$	...	$l(3, n)$
...	...	...	..	...
$y = n$	$l(n, 1)$	$l(2, n)$	...	$l(n, n)$

Table 1. Unbalanced Loss Function

Observe that 0-1 loss is a special case of this unbalanced cost model where $l(i, j) = 1_{\{i \neq j\}}$. Again, we're interested in finding the optimal decision rule f^* that has minimum risk except this time according to the unbalanced cost rule l. Mathematically,

$$\begin{aligned}
f^* &:= \text{argmin}_{f \in \mathcal{F}} R(f, l) \\
&= \text{argmin}_{f \in \mathcal{F}} \mathbb{E}_P[l(y, f(x))] \\
&= \text{argmin}_{f \in \mathcal{F}} \int_{x \in \mathcal{X}} dx \sum_{y \in \mathcal{Y}} P_{x,y}(x, y) l(y, f(x)) \\
&= \text{argmin}_{f \in \mathcal{F}} \int_{x \in \mathcal{X}} g_x(x) dx \sum_{y' \in \mathcal{Y}} P_{y|x}(y|x) l(y, f(x))
\end{aligned}$$

Similar to the logic taken in the derivation for the optimal decision rule for 0-1 loss, to minimize the argument in the last line, for any x in the integral, we take we pick $f(x)$ so as to minimize $\sum_{y \in \mathcal{Y}} P_{y|x}(f(x)|x) l(y, f(x))$. As a result, f^* can be defined as

$$f^*(x) = \text{argmin}_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} P_{y|x}(y'|x) l(y', y) \tag{6}$$

This makes a lot of sense, for characteristics x, the optimal classifier assigns x to the class that will give the smallest expected loss given x[4].

[4] When discussing unbalanced losses, one possible objection is that expected value of loss isn't an ideal quantity to minimize. For example, it's possible the individual constructing the optimal classifier is risk averse and fearful of taking very large costs. That's actually not a problem in this model- we can properly adjust the costs to account for the optimizer's risk preferences.

2.3 *Binary Labeling*

Restrict $\mathcal{Y} = \{0,1\}$ so that we're now trying to find the risk minimizing decision rule f^* for binary class labels with unbalanced costs. We know that:

$$\begin{aligned}
f^*(x) &= \text{argmin}_{y\in\mathcal{Y}} \sum_{y'\in\mathcal{Y}} P_{y|x}(y'|x)l(y',y) \\
&= \text{argmax}_{y\in\{0,1\}} \sum_{y'\in\mathcal{Y}} P_{y|x}(y'|x)l(y',y) \\
&= 1_{\{P_{y|x}(y=1|x)l(1,1)+P_{y|x}(y=0|x)l(0,1)\geq P_{y|x}(y=1|x)l(1,0)+P_{y|x}(y=0|x)l(0,0)\}} \\
&= 1_{\{P_{y|x}(y=1|x)(l(1,1)-l(1,0))\geq P_{y|x}(y=0|x)(l(0,0)-l(0,1))\}} \\
&= 1_{\{\frac{P_{y|x}(y=1|x)}{1-P_{y|x}(y=1|x)}\geq\frac{l(0,0)-l(0,1)}{l(1,1)-l(1,0)}\}} \\
&= 1_{\{\frac{P_{y|x}(y=1|x)}{1-P_{y|x}(y=1|x)}\geq\frac{l(0,1)-l(0,0)}{l(1,0)-l(1,1)}\}} && (7) \\
&= 1_{\{P_{y|x}(y=1|x)\geq\frac{l(0,1)-l(0,0)}{l(0,1)-l(0,0)+l(1,0)-l(1,1)}\}} && (8)
\end{aligned}$$

This has many nice interpretations. First, since 0-1 loss is a special case of the unbalanced cost model, if we let l be the 0-1 loss, we should derive the optimal decision rule for 0-1 loss. That's true: $l(1,0) = 1, l(0,1) = 1, l(0,0) = 0, l(1,1) = 0$ and so equation 7 reduces to equation 3, which is one way to write the optimal 0-1 loss decision rule.

Next, an observation: the optimal decision rule depends only on the differences $l(0,1) - l(0,0), l(1,0) - l(1,1)$[5]. This makes sense because for any class label y, we should be able to select one possible classification to serve as a numeraire. We then re-define the other losses in terms of the numeraire. Let's set $l'(0,0) := 0$ and $l'(1,1) := 0$ to serve as numeraires and consequently $l'(0,1) = l(0,1) - l(0,0)$ and $l'(1,0) = l(1,0) - l(1,1)$. Tabularly, we can convert the loss functions as follows:

$l(y, f(x))$	$f(x) = 0$	$f(x) = 1$	$\longrightarrow$	$l'(y, f(x))$	$f(x) = 0$	$f(x) = 1$
$y = 0$	$l(0,0)$	$l(0,1)$	$\longrightarrow$	$y = 0$	0	$l(0,1) - l(0,0)$
$y = 1$	$l(1,0)$	$l(1,1)$	$\longrightarrow$	$y = 1$	$l(1,0) - l(1,1)$	0

Table 2. Redefining Unbalanced Costs

This finding that we can fix one entry from each row to serve as a numeraire, actually generalizes to any number of class labels. Thus, we can focus on solving problems where we force $l(y,y) = 0\ \forall y \in \mathcal{Y}$. This finding also implies that we can rewrite the optimal decision rule in a condensed form as follows:

$$\begin{aligned}
f^*(x) &= 1_{\{\frac{P_{y|x}(y=1|x)}{1-P_{y|x}(y=1|x)}\geq\frac{l'(0,1)}{l'(1,0)}\}} && (9) \\
&= 1_{\{P_{y|x}(y=1|x)l'(1,0)-(1-P_{y|x}(y=1|x))l'(0,1)\geq 0\}} && (10)
\end{aligned}$$

Again, we can rewrite the optimal classifier in a different way by using Bayes' Rule and starting from the fourth line of the above derivation:

[5] As it turns out, the only thing that matters in the optimal classifier is the ratio of the differences $l(0,1) - l(0,0), l(1,0) - l(1,1)$. Thus, we can can focus on problems where one of the differences is set to 1 and the other difference is a free parameter. We don't make this restriction in this section because we think it's easier to interpret with each difference being free.

$$\begin{aligned} f^*(x) &= 1_{\{P_{y|x}(y=1|x)(l(1,1)-l(1,0))\geq P_{y|x}(y=0|x)(l(0,0)-l(0,1))\}} \\ &= 1_{\{P_{y|x}(y=1|x)(l'(1,0))\geq P_{y|x}(y=0|x)(l'(0,1))\}} \\ &= 1_{\{\frac{g_{x|y}(x|y=1)P(y=1)l'(1,0)}{g_{x|y=1}(x|y=1)P(y=1)+g_{x|y}(x|y=0)P(y=0)}\geq \frac{g_{x|y}(x|y=0)P(y=0)l'(0,1)}{g_{x|y=1}(x|y=1)P(y=1)+g_{x|y}(x|y=0)P(y=0)}\}} \\ &= 1_{\{g_{x|y}(x|y=1)P(y=1)l'(1,0)\geq g_{x|y}(x|y=0)P(y=0)l'(0,1)\}} \end{aligned} \tag{11}$$

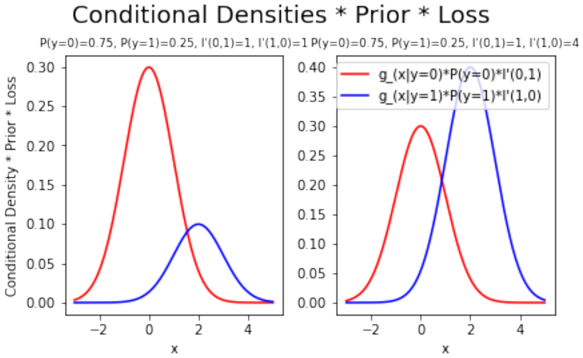

Figure 2. Interpreting Unbalanced Costs Decision Rule

To understand Figure 2, suppose again that $\mathcal{X} = \mathbb{R}$ so that our characteristic is a one dimensional value. In both plots, in red is the graph $g_{x|y=0} * P(y = 0) * l'(0, 1)$ and in blue is the graph $g_{x|y=1} * P(y = 1) * l'(1, 0)$. In both cases, we say that the priors $P(y = 1) = 0.25, P(y = 0) = 0.75$. However, on the left we say that the cost $l'(1, 0) = l'(0, 1) = 1$ while on the right $l'(1, 0) = 4, l'(0, 1) = 1$. For a given x, the optimal classifier picks $y = 0$ if the red graph is above the blue and else picks $y = 1$. In the right plot, we care much more about the error of predicting class $f(x) = 0$ when in fact we should've predicted the other class $y = 1$ than the opposite error. Thus, the blue graph on the right is magnified so that we have a lower threshold for picking $y = 1$ than on the left.

3 Defining cost matrices

All costs in a cost matrix must be strictly derived relative to a common baseline. The *baseline* is the state of the agent before it takes a decision regarding an example. After the agent has made the decision and the true outcome of the example is realized, if the agent is better off, there is a negative cost, if there is no change, there is no cost, and if the agent is worse off, there is positive cost.[6] All costs aim to quantify how much better or worse off the agent is after the classification result (ie., class predicted, action taken, and true class realized) relative to before classification.

There are some additional *reasonableness conditions* (as Elkan (2001) calls them) to think about when constructing cost matrices to avoid degenerate solutions. First, $l(i, i) < l(i, j) \quad i \neq j \in \mathcal{Y}$. In other words, it is always better to pick the true class for an example. In the binary case, this rule means $l(0, 1) > l(0, 0)$ and $l(1, 0) > l(1, 1)$ must be true. If the first inequality does not hold and the second does hold, the optimal decision rule will choose $f(x) = 1 \; \forall x \in \mathcal{X}$ Similarly, if the first inequality holds and the second does not, the optimal decision rule will choose $f(x) = 0 \; \forall x \; \in \mathcal{X}$. If $l(0, 1) < l(0, 0)$ and $l(1, 0) < l(1, 1)$, we either have the degenerate case of trying to purposefully mis-pick the class the example belongs to. If $l(0, 1) = l(0, 0)$ and $l(1, 0) = l(1, 1)$ the optimal decision rule can pick either class for any example.

The second reasonableness condition is that there do not exist j, k such that $l(i, j) \geq l(i, k) \;\; \forall i$ (and one i holds with strict inequality). In the binary example, it is not true that $(l(0, 0) \geq l(0, 1) \wedge l(1, 0) \geq l(1, 1)) \vee (l(0, 0) \leq l(0, 1) \wedge l(1, 0) \leq l(1, 1))$. Tabularly, there is no column m whose entries are greater than or equal to the entries (one row holds with strict inequality) in another column n for each respective row (we say m dominates n). If this is the case then the dominant column will never be selected.

[6] Elkan, 2001. To read more about the limitations of cost matrices and the specifics of defining a baseline, we recommend this paper.

3.1 Binary Classification Example

3.1.1 Some Binary Classification Definitions

In the binary classification setting, there are four possible outcomes of a prediction: true positive ($f(x) = 1, y = 1$; denoted by TP), true negative ($f(x) = 0, y = 0$; denoted by TN), false positive ($f(x) = 1, y = 0$; denoted by FP), and false negative ($f(x) = 0, y = 1$; denoted by FN). Depending on the application, the costs associated with TN, FP, FN, and TP may have different magnitudes.

3.1.2 Simplified Futures

To build intuition on constructing cost matrices with different baselines, let's start with a simplified example of futures trading with balanced costs. We aim to make a bet of \$100 such that if the price of the underlying asset moves up 10% we get a return of \$100 also denoted $+R$ where R stands for risk. If the underlying asset moves down 10% we lose \$100 or $-R$ and our position is liquidated. For simplicity our target lies at $+R$ such that we are out of the market if we double our principle. Whichever happens first, liquidation or we reach our take profit target, will signal the end of our trade. This assumes non-zero volatility, an infinite time-frame, and no slippage or fees.

Given characteristics $x \in \mathcal{X}$ of the asset and market at time t, we define a decision rule $f : \mathcal{X} \mapsto \{0, 1\}$ where $f(x) = 0$ means our system does not take a trade and $f(x) = 1$ means our system enters a trade. Each x is associated with a market outcome y where $y = 0$ means the position was liquidated and $y = 1$ means the asset hit the target successfully.

First, consider the baseline as our wealth prior to observing the example so that we're interested in our profit. If we choose to not engage in a trade (ie., $f(x) = 0$), then our wealth stays exactly the same regardless of the market outcome. If we choose to engage in a trade and our principle doubles, we gain $+R$ in wealth; if our position liquidates, we lose R in wealth. This baseline results in the cost matrix in Table 3.

$l(y, f(x))$	$f(x) = 0$	$f(x) = 1$
$y = 0$	$l(0, 0) = 0$	$l(0, 1) = +R$
$y = 1$	$l(1, 0) = 0$	$l(1, 1) = -R$

Table 3. Trade Example Cost Matrix- Economic Baseline

Next, consider an invalid attempt at constructing an economically sensible cost matrix presented in Table 4.

$l(y, f(x))$	$f(x) = 0$	$f(x) = 1$
$y = 0$	$l(0, 0) = -R$	$l(0, 1) = +R$
$y = 1$	$l(1, 0) = +R$	$l(1, 1) = -R$

Table 4. Trade Example Cost Matrix- Invalid Economic Baseline

One possible rationalization of this cost-matrix is: "In the case ($f(x) = 0, y = 0$) we avoided a trade that would have lost $\$R$ and so we should reward ourselves $\implies l(0, 0) = -R$; in the case ($f(x) = 1, y = 0$) we entered a bad trade that lost us $\$R \implies l(1, 0) = +R$; in the case ($f(x) = 0, y = 1$) we missed a good trade that would have made us $\$R$ and so we have an opportunity cost $\implies l(0, 1) = +R$; in the case ($f(x) = 1, y = 1$) we entered a good trade that made us $\$R \implies l(1, 1) = -R$." This cost matrix doesn't make sense in a monetary sense. The costs $l(1, 1), l(0, 1)$ are defined relative to the agent's wealth had the agent not entered the trade, whereas the costs $l(1, 0), l(0, 0)$ are defined relative to the wealth had the agent entered the trade. Since the costs are defined relative to different baselines, we can't directly add costs from separate evaluations from the classifier to have a notion of profit. That's because if we observe an example $f(x) = 0, y = 1$ and an example $f(x) = 1, y = 1$, adding the two associated costs yields a total cost of 0 where as we actually just made $\$R$ across the two trade opportunities and should have negative total cost.

The rationalization above and wanting to include opportunity costs doesn't make sense in terms of profit, but it does makes sense in the space of "utility" or "happiness". Consider an alternative baseline, the happiness of the trader before he sees the trade opportunity. The cost matrix in Table 4 is rational if the costs are units of utility relative to this baseline. In a sensible rationalization: "In the case ($f(x) = 0, y = 0$) we avoided a trade that would have lost $\$R$ and so we should be happy $\implies l(0, 0) = -R$; in the case ($f(x) = 1, y = 0$) we entered a bad trade that lost us $\$R$ and so we should be upset

$\implies l(1,0) = +R$; in the case $(f(x) = 0, y = 1)$ we missed a good trade that would have made us $\$R$ and so we have an opportunity cost and should be upset $\implies l(0,1) = +R$; in the case $(f(x) = 1, y = 1)$ we entered a good trade that made us $\$R$ and so we should be happy $\implies l(1,1) = -R$." These costs do make sense but only if the user values each possible outcome equally (in absolute value). That said, in portfolio management an opportunity cost tends to be much smaller than a real cost and you are judged primarily on raw performance and risk aversion metrics.

3.1.3 Cost Matrix Using Utility of Wealth

A more practical cost-matrix to consider would be one founded on the utility of wealth. In this case, we assume the agent has utility function $u : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}$ on his wealth that's monotonically increasing and concave: these functional constraints imply that the agent is happier with more money but there are diminishing returns to each dollar accumulated. Before being given the opportunity to engage in this trade, we assume that user has $\$W$ of wealth. We define the baseline as the agent's utility before classification (ie., $u(W)$). That allows us to construct the following cost matrix in Table 5:

$l(y, f(x))$	$f(x) = 0$	$f(x) = 1$
$y = 0$	$l(0,0) = 0$	$l(0,1) = u(W) - u(W - R)$
$y = 1$	$l(1,0) = 0$	$l(1,1) = u(W) - u(W + R)$

Table 5. Trade Example Cost Matrix- Utility of Wealth

If the agent doesn't engage in a trade, his change in utility is zero and so the resulting loss is 0. If the agent engages in a trade, his loss is the negation of the difference in utility after and before. One observation is that given the functional constraints on u (ie., monotonically increasing and concave), $|u(W)-u(W-R)| > |u(W)-u(W+R)| \implies |l(0,1)| > |l(1,1)|$. In other words, the agent suffers more when he loses a given sum of money R than when he makes that sum of money, relative to his initial wealth. This observation agrees with *Prospect Theory* and *Loss-Aversion Theory* which state that "losses loom larger than gains" (Kahneman and Tversky 1992).

3.1.4 An Example

Using the cost matrix defined with the utility of wealth, let's define $u(W) = \ln(W)$, which means we have log utility. Let us also assume that $W = \$1000$ and $R = \$100$. As we only are subject to a cost when we take the trade, we want to know under what circumstances, relative to our cost matrix, we will enter a trade. Beginning with Equation 11 from above, we have:

$$
\begin{aligned}
P_{y|x}(y = 1|x) &\geq \frac{l(0,1) - l(0,0)}{l(0,1) - l(0,0) + l(1,0) - l(1,1)} \\
&\geq \frac{u(W) - u(W - R)}{(u(W) - u(W - R)) - (u(W) - u(W + R))} \\
&\geq \frac{u(W) - u(W - R)}{u(W + R) - u(W - R)} \\
&\geq \frac{\ln(1000) - \ln(900)}{\ln(1100) - \ln(900)} \\
\implies P_{y|x}(y = 1|x) &\geq 0.525
\end{aligned}
$$

The consequence of using the utility of wealth function means that to justify risking 10% of and agent's wealth he has to have at the very least an edge of 2.5% or a 52.5% chance of hitting our take profit before liquidation, gross transactions and slippage to justify entering a trade. It is also important to note that this analysis assumes that capital is infinitely divisible which it is not as most positions have significant minimum size requirements. By customizing the utility of wealth function, an agent can introduce different levels of risk aversion to his loss function.

3.1.5 Maximizing Utility with Risk

An interesting adjacent result comes from the maximization of the utility function defined as $u(W) = ln(W)$, with respect to our risk R, namely the derivation of the Kelly Criterion for optimal betting sizes. To find the optimal Kelly Bet, R^*, we assume that we know W, and we assume we know the relevant probability of success $p := P_{y|x}(y = 1|x)$. We also assume that we know the odds, or the ratio of wealth we stand to gain over wealth we stand to lose. In the case of Table 5 above, the odds are R/R or 1. It is important to note that the odds are calculated as a ratio of wealth at risk, as opposed to a ratio of utility or costs. To solve for the optimal Kelly bet R^*, we write an expression for the expected utility of entering the bet with amount R– we then take the derivative of the formula for expected utility with respect to R, set the derivative equal to 0 and solve for R^*.

$$\begin{aligned}
\mathbb{E}[u] &= p * u(W+R) + (1-p) * u(W-R) \\
0 &= \frac{d\mathbb{E}[u]}{dR} \\
&= \frac{p}{W+R} - \frac{1-p}{W-R} \\
&= \frac{p(W-R) - (1-p)(W+R)}{(W+R)(W-R)} \\
&= \frac{pW - W + pW - R}{(W+R)(W-R)} \\
&= \frac{(2p-1)W - R}{W^2 - R^2} \\
\implies R^* &= (2p-1)W
\end{aligned}$$

If there is an uneven payout, such that the odds are not one to one, we can model this as well with a constant b which scales the reward relative to the wager. In the futures example that could mean that we have now moved our take profit order up, as to only close the trade are we to triple our position, instead of double it. However we still keep the assumption that we are as likely to hit our take profit order, as to be liquidated from our position. We would have 2 to 1 odds as our payout would be double our risk. [7]

$$\begin{aligned}
\mathbb{E}[u] &= p * u(W+Rb) + (1-p) * u(W-R) \\
0 &= \frac{d\mathbb{E}[u]}{dR} \\
&= \frac{pb}{W+Rb} - \frac{1-p}{W-R} \\
&= \frac{pb(W-R) - (1-p)(W+Rb)}{(W+Rb)(W-R)} \\
&= \frac{pWb - pRb - Rb - W - pRb + pW}{(W+Rb)(W-R)} \\
&= \frac{pWb - Rb - W - pW}{(W+Rb)(W-R)} \\
\implies R^* &= \frac{W(p(b+1)-1)}{b}
\end{aligned}$$

3.1.6 Example Continued

Along the lines of the example above from Section 3.1.4, we will continue with $W = \$1000$ and $R = \$100$. We will also start with equal odds, such the the pure wealth in the reward is equal to the pure wealth at risk. By plugging these values

[7] To reflect such odds in the cost matrix analysis from the example in Section 3.1.4, you would adjust $l(1, 1)$ to replace R with Rb.

into our Kelly equation we get $R^* = (2 * 0.525 - 1)1000 = 50$, we get an optimal betting size of $\$50$. This result is in line with the above cost matrix, in that $P_{y|x}(y = 1|x) \geq .525$ is a threshold for us to be able to justify risking $\$100$. Interpreting differently, it's a good idea to risk any amount less than $\$100$ at success probability of 0.525– it's optimal (according to Kelly) to risk $\$50$ with success probability of 0.525. In short the cost matrix threshold tells us the minimum probability to risk a percentage of our wealth, and the Kelly optimization tells us the optimal betting size given a certain probability of success.

Again as a sanity check, if we wanted to reflect odds of say 2 to 1 we would use the formula $R^* = \frac{W(pb+p-1)}{b} = \frac{1000(0.525*2+0.525-1)}{2} = 287.5$ with $b = 2$. If we wanted to reflect odds of 1 to 2 we use $b = 0.5$ and get $R^* = -425$. The negative optimal bet means that there is no risk size with a positive expected value given the assumptions taken. It makes sense that when we have better odds and the same wealth and outcome probabilities, then our optimal betting size should increase. The opposite holds when we have worse odds holding all else constant.

As a warning, these results depend on wealth being infinitely divisible, which in practice is not true. These examples are toys. Additionally, it is extremely rare that the estimated probabilistic outcomes of a bet are accurate in markets, and this analysis does not account for fees or slippage. We do not in any way recommend betting the full Kelly bet. For more, see (Thorp 2006).

3.1.7 Disease Detection

Another standard application of the cost matrix is that of disease diagnosis. As mentioned briefly in Section 2.3, the cost of diagnostic classification has varying costs that depend on whether the patient has the disease or not. Let us assume the disease is fatal, the value of a statistical life according to the United States FEMA is $\$7,500,000$ as of 2020. All patients will have gone to the doctors office for their initial appointment, which we will assume costs the same for all patients. Hence the cost of the initial visit is baked into the baseline. However, patients who are wrongfully diagnosed will undergo unnecessary subsequent treatment. We will assign a cost of $\$100,000$ to this treatment. Patients who have the disease, and are diagnosed correctly and treated, will undergo the same treatment and incur the same cost. Patients who have the disease and are incorrectly diagnosed, will not undergo treatment, and the cost incurred will be that of a human life.

$l(y, f(x))$	$f(x) = 0$	$f(x) = 1$
$y = 0$	$l(0, 0) = 0$	$l(0, 1) = +\$100k$
$y = 1$	$l(1, 0) = +\$7.5M$	$l(1, 1) = +\$100k$

Table 6. Disease Example Cost Matrix

As per Equation 11 above, we get the following classification threshold.

$$\begin{aligned}
P_{y|x}(y = 1|x) &\geq \frac{l(0,1) - l(0,0)}{l(0,1) - l(0,0) + l(1,0) - l(1,1)} \\
&\geq \frac{100,000}{100,000 + 7,500,000 - 100,000} \\
&\geq \frac{100,000}{7,500,000} \\
&\geq 0.013
\end{aligned}$$

In other words, given these costs, if we believe that the likelihood that an individual has the disease is greater than 0.013, it's better to say declare the individual has the disease and have them undergo the treatment.

4 Classification in an Empirical Setting

In the real world, for classification problems, we're given a data set $D = \{z_i = (x_i, y_i)\}_{i=1}^{n}$ of n data points where $x_i \in \mathcal{X}$ is a set of characteristics and $y_i \in \mathcal{Y}$ is an associated class label for z_i. The main difference now, alluded to in a footnote earlier, is that we don't actually know the distribution $P_{x,y}$ nor any of the associated conditional and marginal distributions.

The objective of the real world classification problem is to find a decision rule $f : \mathcal{X} \mapsto \mathcal{Y}$ that does "a good job" at labeling the data points in D given the characteristics. Again, to quantify "good job", we define a loss function $l : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ so that $l(y, f(x))$ reports the penalty f receives for assigning class label $f(x)$ to some characteristics x instead of the correct label y. Since we don't know the distribution $P_{x,y}$ nor any of the associated distributions, we cannot proceed as in the previous sections to find the risk minimizing decision rule. Instead, we need to estimate (a subset of) the probability distributions first to then optimize over f. Specifically, we need to estimate either $P_{y|x}$ or $P_y, g_{x|y}$ to find the optimal classification rules f^*.

4.1 Assume We've Estimated $P_{y|x}$ (or $P_y, g_{x|y}$)

First, assume we've estimated $P_{y|x}$ given our data D. How do we come up with our optimal classifier f^* according to find the risk minimizing decision rule according to l? Recall equation 6: it says for any x, f^* assigns it to the class that results in the smallest expected loss given x. Thus, to find $f^*(x)$, we can use our estimated $P_{y|x}$ to estimate the expected losses for each prediction given x, and then $f^*(x)$ returns the predicted class with the smallest expected loss given x.

We can proceed very similarly if we've estimated $P_y, g_{x|y}$ by using Bayes' Rule.

Thus, finding our optimal classification rule f^* in an empirical setting reduces to estimating $P_{y|x}$ or $P_y, g_{x|y}$.

4.2 An Aside on Imbalanced Classes

In this article so far, we've theoretically addressed classification when costs of classification errors vary. A related but disjoint topic is that of imbalanced classes– our training dataset $D = (x_i, y_i)_{i=1}^N$ has relatively more examples of one class than another class. This observation may cause problems when we're training our model– without care, an imbalanced training dataset D may skew our estimates of the marginal distributions $g_{x|y}$, $P_{y|x}$, and prior distribution $P(y)$.

4.2.1 Unbiased Estimate of the Prior Probabilities of Classes

A first remark, is that when estimating the prior $P(y)$, from an *unbiasedness*[8] perspective, unbalanced classes in the training set are only problematic if they do not reflect the class distribution of the population and test sets. When estimating the prior distribution $P(y)$ by itself, an unbiased approach would be to estimate $P(y) := \sum_{i=1}^{|D|} \frac{1_{y_i==y}}{|D|}$; that is, for each class y, we estimate its prior as its sample frequency in D. If the sample frequencies of classes in the training set D do not reflect the frequencies of classes in the population and test sets, then our estimate of $P(y)$ will be biased. Notice that this problem can arise whether the classes are balanced or imbalanced– we require that the training set well represent the population and test sets in terms of class distribution. Thus, imbalanced classes in the training set are not alone a problem if the frequencies in the training set represent those in the population and test sets.

As a second remark, recall that $P_{y|x} \propto g_{x|y}P(y)$. That means, if our estimate of $P(y)$ is biased, our estimate of the marginal distribution $P_{y|x}$ will also be biased.

4.2.2 Estimating the Marginal Distribution $g_{x|y}$

As a third remark, let's consider the impact of estimating $g_{x|y}$ given an imbalanced training set $D = (x_i, y_i)$ that is representative of the population and test sets. First, assuming that each $(x_i, y_i) \sim g_{x_i|y_i}$, there is no problem of *unbiasedness*. In fact, we could separate all samples in D by class $y_i \in Y$, and then estimate $g_{x|y_i}$ independently for each y_i. We only require that we have many examples of each class y_i so we can "confidently" estimate $g_{x|y_i}$. When there are unbalanced classes-there are many more examples of class y_i than y_j- if there are few examples of class y_j, we will come up with a "bad" estimate of $g_{x|y_j}$ and we will come up with "better" estimate of $g_{x|y_i}$. In other words, to cleanly estimate $g_{x|y}$, we don't require balanced classes in D, because in fact $g_{x|y_i}$ can be estimated separately for each class y_i. We require that we have enough examples of each class y_i to cleanly estimate $g_{x|y_i}$.

[8] An estimator $\hat{x}$ is *unbiased* in its estimate of x_0 iff $\mathbb{E}[\hat{x}] = x_0$.

4.2.3 Relationship between Imbalanced Classes and Imbalanced Costs

As a fourth remark, classification problems with imbalanced classes often come hand-in-hand with imbalanced costs of classification. Take for example the two-class disease example where we are trying to decide whether an individual has a fatal disease or is healthy based on some characteristics x. Typically, we have many more examples of healthy individuals than sick individuals (ie., imbalanced classes) and that's representative of the population. The error of declaring an individual healthy when they are in fact sick is much more costly than declaring the individual sick when they are in fact healthy (ie., imbalanced costs).

To directly address the problem of imbalanced costs, we have above proposed the idea of *thresholding* to decide which class an individual belongs to. If $P(y = 1|x) \geq p^*$, where p^* is a threshold that's possibly different from 0.5 and defined according to the imbalanced costs (see Equation 11), we declare the individual belongs to class $y = 1$.

Elkan (2001) in Theorem 1 notes that there is another solution that addresses the problem of imbalanced costs indirectly by seemingly addressing the imbalanced classes. Elkan establishes the equivalence of a classifier C, with threshold p^* trained on training data D to a classifier C' with threshold p_0 trained on training data D'. Suppose that we have a given probability threshold of p_0 (typically 0.5) and a target threshold p^* for declaring a sample belongs to class $y = 1$. We can make the the given probability threshold p_0 address the target threshold p^* by multiplying the number of examples of the $y = 0$ class by

$$\frac{p^*}{1-p^*} * \frac{1-p_0}{p_0}$$

The tricky proof of this statement is given at the end of Section 3 of the Elkan (2001) paper. To give an example, suppose that we have a target probability $p_0 = 0.5$ and a target probability $p^* = 0.01$, decided by the imabalanced costs. We multiply the prevalence of all $y = 0$ samples of the training class D by $\frac{0.01}{0.99}$ to produce D'. The theorem states that a classifier C, with threshold 0.01 trained on training data D will make the same decisions as a classifier C' with threshold 0.5 trained on training data D'.

4.2.4 Addressing Learning with Imbalanced Classes in Logistic Regression

As a final remark, I'd like to propose a modification to logistic regression that would address remark 3 and the difficulty estimating $g_{x|y}$ with imbalanced classes (and no imbalanced costs). I don't know if this modification has been studied before but it sounds interesting and useful to me.

To set the stage, consider a two-class classification problem where datapoints $D = (x_i, y_i)_{i=1}^N$ have characteristics $x_i \in \mathbb{R}^M$ and belong to classes $y_i \in \{0, 1\}$. Suppose class $y = 1$ occurs with much less frequency than class $y = 0$ but that is representative of the population. Also suppose, we try to estimate $P_{y|x}$ using a unregularized logistic model. That means, we define $P_{y|x}(y = 1|x) := 1 - \frac{1}{1+e^{\beta x_i}}$ and pick $\beta \in \mathbb{R}^M$ so as to maximize:

$$\begin{aligned}
\beta^* &:= \text{argmax}_{\beta \in \mathbb{R}^M} \Pi_{i=1}^N (P_{y|x}(y = 1|x))^{y_i} + (P_{y|x}(y = 0|x))^{(1-y_i)} \\
&= \text{argmax}_{\beta \in \mathbb{R}^M} \Pi_{i=1}^N (\frac{1}{1 + e^{\beta x_i}})^{y_i} + (1 - \frac{1}{1 + e^{\beta x_i}})^{(1-y_i)} \\
&= \text{argmax}_{\beta \in \mathbb{R}^M} \sum_{i=1}^N (y_i) \log(\frac{1}{1 + e^{\beta x_i}}) + (1 - y_i) \log(1 - \frac{1}{1 + e^{\beta x_i}})
\end{aligned}$$

When picking β^*, note that we "weight" all datapoints evenly in the summation. Since there are more datapoints of class $y = 0$, our estimator will be able to learn relatively more about samples with class $y = 0$ than those with class $y = 1$, that's consistent with remark 3. What if bootstrap more samples of class $y = 1$ from those we have (there are many approaches to do this that would need to be studied) to produce training set D' that has an even number of samples of each class. Then, we estimate $P'_{y|x}$ based on D'. Recall that according to remarks 1 and 2, $P'_{y|x}$ will be biased since we have adjusted the prior probabilities. We produce

$$P_{y|x}(y = 1|x) := P'_{y|x}(y = 1|x) * \frac{P(y = 1)}{P'(y = 1)}$$

where $P(y = 1)$ is the sample frequency of class $y = 1$ in the training data D and $P'(y = 1) = 0.5$ is the frequency of class $y = 1$ in the modified training set D'. I find this approach intriguing because, we will have kept an unbiased estimate for $P_{y|x}$ and "learned" more about the distribution of characteristics for the minority class $y = 1$ in our estimate.

REFERENCES

Multinomial logistic regression. URL: https://en.wikipedia.org/wiki/Multinomial_logistic_regression.

Jason Brownlee. 10 Standard Datasets for Practicing Applied Machine Learning, October 20, 2021. URL: https://machinelearningmastery.com/standard-machine-learning-datasets/.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

Charles Elkan. The Foundations of Cost-Sensitive Learning. *Proceedings of the Seventeenth International Conference on Artificial Intelligence: 4-10 August 2001; Seattle*, 1, 05 2001.

Pedro Felzenszwalb. Classification and Decision Theory, Spring 2017. URL: https://cs.brown.edu/people/pfelzens/engn2520/CS1420_Lecture_5.pdf.

Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn TensorFlow*. O'Reilly, 2017.

Charles Ling and Victor Sheng. Cost-Sensitive Learning and the Class Imbalance Problem. *Encyclopedia of Machine Learning*, 01 2010. URL: https://www.csd.uwo.ca/~xling/papers/cost_sensitive.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Philippe Rigollet. 18.657: Mathematics of Machine Learning, September 9, 2015. URL: https://ocw.mit.edu/courses/18-657-mathematics-of-machine-learning-fall-2015/7d229ed907a6d1410a3736c98a9d78d8_MIT18_657F15_L1.pdf.

Edward O. Thorp. The Kelly Criterion in Blackjack, Sports Betting, and The Stock Market. *Handbook of Asset and Liability Management*, 1, 2006.

Amos Tversky and Daniel Kahneman. Advances in Prospect Theory: Cumulative Representation of Uncertainty. *Journal of Risk and Uncertainty*, 5:297–323, 1992.

Martin Wainwright. EECS 281B / STAT 241B: Advanced Topics in Statistical Learning- Lecture 2, January 26, 2009. URL: https://people.eecs.berkeley.edu/~wainwrig/stat241b/lec2.pdf.