Estimating the Class Probability Threshold without Training Data

Ricardo Blanco-Vega **RBLANCO@DSIC.UPV.ES**
César Ferri-Ramírez **CFERRI@DSIC.UPV.ES**
José Hernández-Orallo **JORALLO@DSIC.UPV.ES**
María José Ramírez-Quintana **MRAMIREZ@DSIC.UPV.ES**
Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, C. de Vera s/n, 46022 Valencia, Spain

Abstract

In this paper we analyse three different techniques to establish an optimal-cost class threshold when training data is not available. One technique is directly derived from the definition of cost, a second one is derived from a ranking of estimated probabilities and the third one is based on ROC analysis. We analyse the approaches theoretically and experimentally, applied to the adaptation of existing models. The results show that the techniques we present are better for reducing the overall cost than the classical approaches (e.g. oversampling) and show that cost contextualisation can be performed with good results when no data is available.

1. Introduction

The traditional solution to the problem of contextualising a classifier to a new cost is ROC analysis. In order to perform ROC analysis (as well as other techniques), we need a training or validation dataset, from which we draw the ROC curve in the ROC space. In some situations, however, we don't have any training or validation data analysis available.

This situation is frequent when we have to adapt an existing method which was elaborated by a human expert, or the model is so old that we do not have the old training data used for constructing the initial model available. This is a typical situation in many areas such as engineering, diagnosis, manufacturing, medicine, business, etc.

Therefore, the techniques from machine learning or data mining, although they are more and more useful and frequent in knowledge acquisition, cannot be applied if we have models that we want to adapt or to transform, but we do not have the original data.

An old technique that can work without training data is the recently called "cost-sensitive learning by example weighting" (Abe et. al., 2004). The methods which follow this philosophy modify the data distribution in order to train a new model which becomes cost-sensitive. The typical approach in this line is stratification (Breiman et. al., 1984; Chan and Stolfo, 1998) by oversampling or undersampling.

An alternative approach is the use of a threshold. A technique that could be adapted when data is not available can be derived from the classical formulas of cost-sensitive learning. It is straightforward to see (see e.g. Elkan, 2001) that the optimal prediction for an example *x* in class *i* is the one that minimises

$$L(x,i) = \sum_{j} P(j \mid x) C(i,j) \qquad \textbf{(1)}$$

where *P*(*j*|*x*) is the estimated probability for each class *j* given the example *x*, and *C*(*i*,*j*) is the cell in the cost matrix *C* which defines the cost of predicting class *i* when the true class is *j*. From the previous formula, as we will see, we can establish a direct threshold without having any extra data at hand. In fact, some existing works (Domingos, 1999) have used the previous formula to establish a threshold which generates a model which is cost sensitive.

One of the most adequate ways to establish a class threshold is based on ROC analysis. (Lachiche & Flach, 2003) extend the general technique and show that it is also useful when the cost has not changed. However, in these cases we need additional validation data, in order to draw the curves.

In order to tackle the problem that we have described at the beginning (adapting an existing model without data), it would be interesting, then, to analyse some techniques

Appearing in *Proceedings of the third Workshop on ROC Analysis in Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

which combine the direct threshold estimation based on formula 1 (which ignores any estimated probabilities) and methods which take them into account (either their ranking or their absolute value) in a similar way ROC analysis works, but without data.

In order to adapt the existing models, we use the mimetic technique (Domingos, 1997, 1998; Estruch, Ferri, Hernández & Ramírez, 2003; Blanco, Hernández & Ramírez, 2004) to generate a model which is similar to the initial model (oracle) but contextualised to the new cost. In order to do this, we propose at least six different ways to diminish the global cost of the mimetic model. Three criteria for adapting the classification threshold, as we have mentioned, and several different schemas for the mimetic technique are set out (without counting on the original data). We have centered our study on binary classification problems.

The mimetic method is a technique for converting an incomprehensible model into one simple and comprehensible representation. Basically, it considers the incomprehensible model as an oracle, which is used for labelling an invented dataset. Then, a comprehensible model (for instance, a decision tree) is trained with the invented dataset. The mimetic technique has usually been used for obtaining comprehensible models. However, there is no reason for ignoring it as a cost-sensitive adaptation technique since it is in fact a model transformation technique.

Note that the mimetic technique is a transformation technique which can use any learning technique, since the mimetic model is induced from (invented) data.

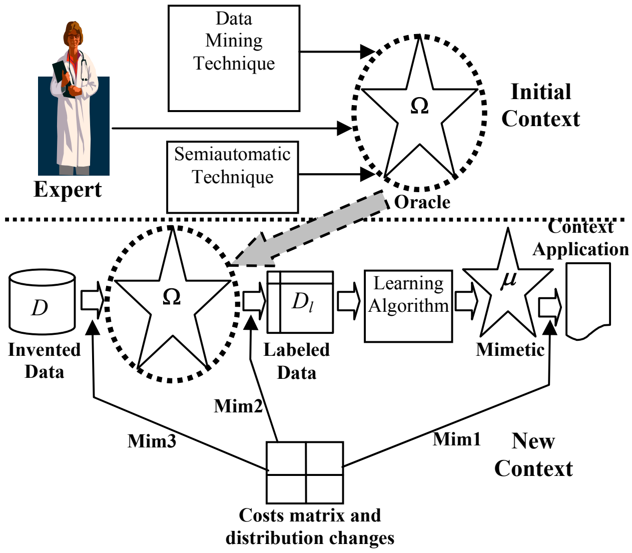

Figure 1. The mimetic context.

The mimetic context validation (see Figure 1) that we propose allows us to change the context of the initial model (oracle) so that it becomes sensitive to the new cost.

The main advantages of our proposal are that it does not require a retraining of the initial model with the old data and, hence, it is not necessary to know the original data. The only thing we need from the original data or for the formulation of the problem is to know the maximum and minimum values of its attributes.

From these maximum and minimum values and applying the uniform distribution we can obtain an invented dataset, which is labelled by using the oracle. We can use the cost information in different points: on the invented dataset, on the labelling of the data or on the thresholds. This settles three moments at which the cost information is used (see Figure 1, the points Mim1, Mim2 and Mim3). One of the points (Mim2) is especially interesting from the rest because it generates a "specific rule formulation" for the model, which might serve as any explanation of the adaptation to the new costs.

The paper is organised as follows. In section 2 we describe the three methods to determine the thresholds and analyse theoretically the relationships between them. In section 3 we describe the four styles for the generation of invented data and, the schemes used in this work for the learning of the mimetic models. In section 4 we describe the different configurations. We also include the experimental evaluation conducted and the general results, which demonstrate the appropriateness and benefits of our proposal to contextualise any model to a new cost context. Finally, section 5 presents the conclusions and future work.

2. Threshold Estimation

In this section, we present three different methods to estimate an optimal threshold following different philosophies. We also study some theoretical properties of the methods.

In contexts where there are different costs associated to the misclassification errors, or where the class distributions are not identical, a usual way of reducing costs (apart from oversampling) is to find an optimal decision threshold in order to classify new instances according to their associated cost. Traditionally, the way in which the threshold is determined is performed in a simple way (Elkan, 2001), only taking the context *skew* into account.

As we have said in the introduction, the methods based on ROC analysis (e.g. Lachiche & Flach, 2003) require a validation dataset, which is created at the expense of reducing data in the training dataset. Here, we are only interested in threshold estimation methods that don't require extra data, since we do not have any data available (either old or new training or test). Therefore, we will not study this method or others which are related which require a dataset. We will just present methods which can work without it.

In this section we consider two-class problems, with class names 0 and 1. Given a cost matrix C, we define the cost *skew* as:

$$skew = \frac{C(0,1) - C(1,1)}{C(1,0) - C(0,0)} \qquad \textbf{(2)}$$

2.1 Direct Threshold

The first method to obtain the threshold completely ignores the estimated probabilities of the models, i.e., to estimate the threshold it only considers the cost *skew*. According to (Elkan, 2001), the optimal prediction is class 1 if and only if the expected cost of this prediction is lower than or equal to the expected cost of predicting class 0:

$$P(0 \mid x) \cdot C(1,0) + P(1 \mid x) \cdot C(1,1) \leq P(0 \mid x) \cdot C(0,0) + P(1 \mid x) \cdot C(0,1)$$

If $p = P(1|x)$ we have:

$$(1 - p) \cdot C(1,0) + p \cdot C(1,1) \leq (1 - p) \cdot C(0,0) + p \cdot C(0,1)$$

Then, the threshold for making optimal decisions is a probability p^* such that:

$$(1 - p^*) \cdot C(1,0) + p^* \cdot C(1,1) = (1 - p^*) \cdot C(0,0) + p^* \cdot C(0,1)$$

Assuming that $C(1,0)>C(0,0)$ and $C(0,1)>C(1,1)$ (i.e. misclassifications are more expensive than right predictions), we have

$$p^* = \frac{C(1,0) - C(0,0)}{C(1,0) - C(0,0) + C(0,1) - C(1,1)}$$

$$p^* = \frac{1}{1 + skew}$$

Finally, we define the threshold as:

$$Threshold_{Dir} = 1 - p^* = \frac{skew}{1 + skew} \qquad \textbf{(3)}$$

2.2 Ranking or Sorting Threshold

The previous method for estimating the classification ignores the estimated probabilities in a proper way. This can be a problem for models that do not distribute the estimated probabilities. Imagine a model that only assigns probabilities within the range 0.6-0.7. In this situation, most of the *skews* will not vary the results of the model.

In order to partially avoid this limitation, we propose a new method to estimate the threshold. The idea is to employ the estimated probabilities directly to compute the threshold. For this purpose, if we have n examples, we rank these examples according to their estimated probabilities of being class 0. We select a point (*Pos*) between two points (*a*,*b*) in this rank such that there are (approximately) $n/(skew{+}1)$ examples on the left side and $(n^*\ skew/(skew{+}1))$ examples on the right side. In this division point we can find the desired threshold. We can illustrate this situation with Figure 2:

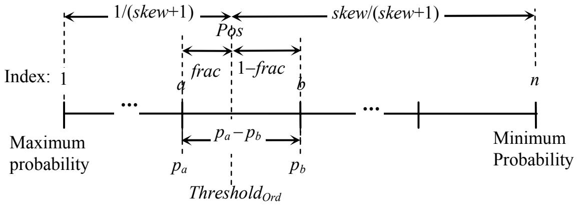

Figure 2. Position of the threshold in the sorting method

Following this figure, we have

$$Pos = \frac{n - 1}{skew + 1} + 1,\ a = Lower(Pos),\ \ b = a + 1$$

where *Lower* computes the integer part of a real number. Then we estimate the threshold as:

$$Threshold_{Ord} = p_a - (p_a - p_b) \cdot frac \qquad \textbf{(4)}$$

where

$$frac = Pos - a$$

In the case we find more than one example with the same estimated probability, we distribute these examples in a similar way. A complete explanation of the procedure can be found in (Blanco, 2006).

2.3 ROC Threshold

Although the previous method considers the *skew* and the estimated probabilities to compute the threshold, it has an important problem because the value of the threshold is restricted to the range of the probabilities computed by the model. I.e, if a model always computes probability estimates between 0.4 and 0.5, the threshold will be within this range for any *skew*.

Motivated by this limitation, we have studied a new method to compute the threshold based on ROC analysis. Suppose that a model is well calibrated, this fact means that if a model gives a probability 0.8 of being class 0 to 100 examples, 80 should be of class 0, and 20 should be of class 1. In the ROC space, this will be a segment going from point (0,0) to the point (20,80) with a slope of 4.

In order to compute this new threshold we define a version of the ROC curve named NROC. This new curve is based on the idea that a probability represents a percentage of correctly classified instances (calibrated classifier).

If we have a set of n examples ranked by the estimated probability of being class 0, we define Sum^0 as the sum of these probabilities. We consider normalised probabilities, then $Sum^0 + Sum^1 = n$. The space NROC is a 2 dimension square limited by (0,0) and (1,1). In order to draw a NROC curve, we only take the estimated probabilities into account, and we proceed as follows. If the first example has an estimated probability p_1 of being class 0, we draw a segment from the point (0,0) to the point $((1-p_1)/Sum^1,\ p_1/Sum^0)$. The next instance ($p_2$) will correspond to the second segment will be from $((1-p_1)/Sum^1, p_1/Sum^0)$ to $(((1-p_1)+(1-p_2))/Sum^1,\ ((p_1+p_2)/Sum^0)$. Following this procedure, the last segment will be between the points $(Sum^1-(1-p_n))/Sum^1,\ (Sum^0-p_n)/Sum^0)$ and (1,1).

Once we have defined the NROC space, let us explain how we use it to determine the threshold. First, since we work on a normalised ROC space (1×1) and Sum^0 is not always equal to Sum^1, we need to normalise the *skew*.

$$skew' = skew \cdot \frac{Sum^0}{Sum^1}$$

If *skew'* is exactly parallel to a segment, then the threshold must be exactly the probability that corresponds to that segment, i.e if $skew' = p_i/(1-p_i)$ the threshold must be p_i. This means:

$$Threshold_{ROC} = \frac{skew'}{1+skew'}$$

Using the relationship between *skew*' and *skew*:

$$Threshold_{ROC} = \frac{skew \cdot \frac{Sum^0}{Sum^1}}{1 + skew \cdot \frac{Sum^0}{Sum^1}}$$

$$Threshold_{ROC} = \frac{1}{1 + \frac{1}{skew} \cdot \frac{Sum^0}{Sum^1}} \qquad \textbf{(5)}$$

2.4 Theoretical analysis of the threshold methods

Now, we study some properties of the methods for obtaining the threshold which we have described in the previous subsections. First, we show that the threshold which is calculated by each of the three methods is well-defined, that is, it is a real value between 0 and 1, as expected. Secondly, we analyse which the relationship between the three thresholds is.

The maximum and minimum values of the $Threshold_{Dir}$ and $Threshold_{ROC}$ depend on the *skew* by definition (formulae 3 and 5). Trivially, $Threshold_{Ord}$ belongs to the interval [0..1] since it is defined as a value between two example probabilities.

Maximum: For the direct and the ROC methods, the maximum is obtained when *skew*=∞:

$$\lim_{skew\to\infty} Threshold_{Dir} = \lim_{skew\to\infty} \frac{skew}{1+skew} = 1$$

$$\lim_{skew\to\infty} Threshold_{ROC} = \lim_{skew\to\infty} \frac{1}{1 + \frac{1}{skew} \cdot \frac{Sum^0}{Sum^1}} = 1$$

The upper limit of $Threshold_{Ord}$ is not necessarily 1, since it is given by the example with highest probability.

Minimum: For the direct and the ROC methods, the minimum is obtained when *skew*=0:

$$\lim_{skew\to 0} Threshold_{Dir} = \lim_{skew\to 0} \frac{skew}{1+skew} = 0$$

$$\lim_{skew\to 0} Threshold_{ROC} = \lim_{skew\to 0} \frac{1}{1 + \frac{1}{skew} \cdot \frac{Sum^0}{Sum^1}} = 0$$

As in the previous case, the lower limit of $Threshold_{Ord}$ is not necessarily 0, since it is given by the example with lowest probability.

Regarding the relationship among the three threshold methods, it is clear that we can found cases for which $Threshold_{Dir} > Threshold_{Ord}$, and viceversa, because, as we have just said, the $Threshold_{Ord}$ value depends on the example probability of being of class 0. A similar relationship holds between $Threshold_{ROC}$ and $Threshold_{Ord}$.

However, the relationship between $Threshold_{ROC}$ and $Threshold_{Dir}$ depends on the relationship between Sum_1 and Sum_0, as the following proposition shows:

Proposition 2. *Given n examples, let Sum^0 be the sum of the n (normalised) example probabilities of being in class 0, and let Sum^1 be $1-Sum^0$. If $Sum^0/Sum^1 > 1$ then $Threshold_{ROC} > Threshold_{Dir}$, if $Sum^0/Sum^1 < 1$ then $Threshold_{ROC} < Threshold_{Dir}$, and if $Sum^0/Sum^1 = 1$ then $Threshold_{ROC} = Threshold_{Dir}$.*

The following theorem shows that the three thresholds coincide when the probabilities are uniformly distributed.

.**Proposition 3.** *Given a set of n examples whose probabilities are uniformly distributed. Let P^0 be the sequence of these probabilities ranked downwardly:*

$$P^0 = \{1, \frac{m-1}{m}, \ldots, \frac{2}{m}, \frac{1}{m}, 0\}$$

such that the probability of example i being in class 0 *and class* 1 *are given respectively by*

$$p_i^0 = \frac{m-i+1}{m} \quad y \quad p_i^1 = \frac{i-1}{m}$$

where $m=n-1$.

Then, $Threshold_{ROC}=Threshold_{Dir}=Threshold_{Ord}$.

3. Mimetic Context

In this section we present the mimetic models we will study experimentally in the next section along with the threshold estimation seen in Section 2. For this purpose, we first introduce several ways to generate the invented dataset, as well as different learning schemes. Then, each configuration to be considered will be obtained by inventing its training dataset in a certain way, by applying one of the learning schemes and by using one of the thresholds defined in the previous section.

3.1 Generation of the training dataset for the mimetic technique

As we said in the introduction, we are assuming that the original dataset used for training the oracle is not available. Hence, the mimetic model is training by using only an invented dataset (labelled by the oracle) which is generated using the uniform distribution. This is a very simple approach, because in very few cases data follow this distribution. If we could know the a priori distribution of the data or we could have a sample where we could estimate this distribution, the results would be probably better. Note that, in this way we only need to make use of the range value of each attribute (that is, its maximum and minimum values).

In general, the invented dataset *D* can be generated by applying one of the following methods:

- **Type a: A priori method**. In this method, *D* preserves the class distribution of the original training dataset. To do this, the original class proportion has to be known at the time of the data generation.
- **Type b: Balanced method**. The same number of examples of each class is generated by this method. So *D* is composed by a 50% of examples of class 1 and a 50% of examples of class 0.
- **Type c: Random method**. The invented dataset *D* is obtained by only using the uniform distribution as it is (that is, no conditions about the class frequency in *D* are imposed).
- **Type d: Oversampling method**. This method makes that the class frequencies in the invented dataset are defined in terms of the *skew*, such that *D* contains a proportion of 1/(*skew*+1) of instances of class 0 and a proportion of *skew*/(*skew*+1) of instances of class 1.

In order to obtain the four types, we generate random examples and then we label them using the oracle. This process is finished when we obtain the correct percentage according to the selected type.

3.2 Mimetic Learning Schemes

In order to use the mimetic approach for a context sensitive learning, different mimetic learning schemes can be defined depending on the step of the mimetic process the context information is used: at the time of generating the invented dataset (scheme 3), at the time of labelling the invented dataset (scheme 1) or at the time of application of the mimetic model (scheme 2). We also consider another scheme (scheme 0) which corresponds to the situation where the context information is not used (as a reference). More specifically, we define the following mimetic learning schemes:

- **Scheme 0 (Mim0 model)**: This is the basic mimetic scheme. The mimetic model is obtained by applying a decision tree learner to the labelled data, namely the J48 classifier with pruning (Figure 3). Then, Mim0 is applied as a non sensitive context model that classifies a new example of class 0 if the probability for this class is greater or equal to 0.5 (threshold=0.5).

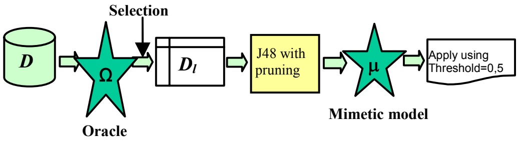

Figure 3. Scheme 0: The simple mimetic learning method.

- **Scheme 1 (Mim1 model)**: This is a posteriori scheme in that the context information is used when the mimetic model is applied. First, the mimetic model is obtained as usually (by using the J48 classifier without pruning). Then, the threshold is calculated from the mimetic model and the invented dataset. Finally, the Mim1 model uses these parameters to classify new examples. Figure 4 shows this learning scheme.

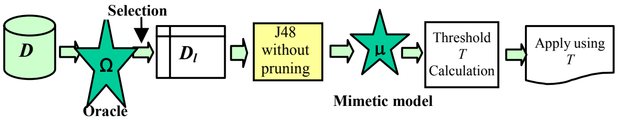

Figure 4. Scheme 1: The context information is used at the time of the mimetic model application.

- **Scheme 2 (Mim2 model):** This is a priori scheme in which the context information is used before the mimetic model is learned. Once the invented dataset has been labelled by the oracle, the threshold and the Ro index (if it is needed) are calculated from them. Then, the invented dataset is re-labelled using these parameters. The new dataset is used for training the mimetic model which is applied as in scheme 0. This learning scheme is very similar to the proposal of

(Domingos, 1999). Figure 5 illustrates this learning scheme.

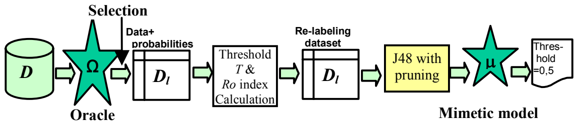

Figure 5. Scheme 2: The context information is used to re-label the invented dataset before the mimetic model is trained.

- **Scheme 3 (Mim3 model)**: This is a scheme in which the context information is used for generating the invented dataset using oversampling. Then, the mimetic model is generated and applied as in scheme 0 (Figure 6).

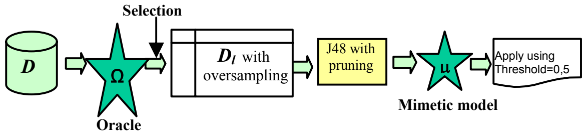

Figure 6. The context information is used at the time to generate the invented dataset by oversampling.

Note that the J48 learning method has been used with pruning in all the schemes except to scheme 1. This is due to the fact that, in this case, we need that the mimetic model provides good estimations of probabilities in order to calculate the threshold from them.

4. Experiments

In this section, we present the global results of the experimental evaluation of the mimetic technique as a model contextualization approach. A more exhaustive experimental evaluation can be found in (Blanco, 2006). The combinations we will analyse are obtained as follows. First, we combine the Mim1 and Mim2 models with the three thresholds defined in Section 2. This gives 6 different configurations. We also consider the Mim0 and Mim3 models. Finally, we combine all these models (except from Mim3) with the different ways of inventing the training dataset defined in Section 3. Summing up, the experimental configuration is composed by 22 mimetic models to be studied. In that follows, a mimetic model is denoted as Mim*nConfigType*, where *n* denotes a learning scheme (0≤*n*≤3), *Config* denotes the threshold used (Ord, Dir, ROC), and *Type* denotes the different types of invented dataset generation (a,b,c,d) described in section 3.1.

4.1 Experimental Setting

For the experiments, we have employed 20 datasets from the UCI repository (Black & Merz, 1998) (see Table 1**¡Error! No se encuentra el origen de la referencia.**).

Datasets from 1 to 10 have been used for the experiments in an (almost)-balanced data scenario, whereas the rest of them have been used for two unbalanced data situations: first, considering class 1 as the majority class and, secondly, as minority class. In all cases, we use cost matrices with *skew* values of 1, 2, 3, 5, and 10. The mimetic models have been built using the J48 algorithm implemented in Weka (Witten & Frank, 2005). Also, we have used two oracles: a Neural Network and a Naive Bayes algorithm (their implementations in Weka). This allows us to analyse our approach both when the oracle is calibrated (the case of the neural network which provides good calibration) and non-calibrated (the Naive Bayes classifier). The size of the invented dataset is 10,000 for all the experiments and we use Laplace correction for all the probabilities. For all the experiments, we use 10x10-fold cross-validation. Finally, when we show average results, we will use the arithmetic mean. We show the means because the number of variants is too large to include here the table with the paired t-tests. You can see these results in (Blanco, 2006).

Table 1. Information about the datasets used in the experiments.

No.	Dataset	Balanced	Attributes		Size	Size	
			Num.	Nom.		Class 0	Class 1
1	credit-a	Almost	6	9	690	307	383
2	heart-statlog	Almost	13	0	270	150	120
3	monks1	yes	0	6	556	278	278
4	monks3	Almost	0	6	554	266	288
5	monks2	Yes	0	6	412	206	206
6	tic-tac	Yes	0	8	664	332	332
7	breast-cancer	Yes	0	9	170	85	85
8	labor	Yes	8	8	40	20	20
9	vote	Yes	0	16	336	168	168
10	diabetes	Yes	8	0	536	268	268
11	haberman-breast	No	3	0	306	81	225
12	monks2	No	0	6	601	206	395
13	abalone-morethan	No	7	1	4177	1447	2730
14	tic-tac	No	0	8	958	332	626
15	breast-cancer	No	0	9	286	85	201
16	labor	No	8	8	57	20	37
17	vote	No	0	16	435	168	267
18	credit-g	No	7	13	1000	300	700
19	diabetes	No	8	0	768	268	500
20	liver	No	6	0	345	145	200

4.2 General Results

An overview of our approach is shown in Table 2, which presents the cost average of the mimetic models obtained in all experiments grouped by *skew*. As can be observed, for *skew*=1, the cost is quite similar in all mimetic models. However, as the *skew* value increases, the cost differences are more meaningful. Globally, and for *skew* values greater than 2, Mim2Ordb model presents the best behaviour, followed by Mim2Orda. For lower *skew* values, the best models are Mim2Dira and Mim2Dirb. Hence, from a cost point of view, it seems preferable to apply the cost information before the mimetic model is built (Mim2 configurations).

Table 2. Cost averages of all the mimetic models grouped by *skew*. In bold those with the lowest global cost.

Model	*skew*					Mean
	1	2	3	5	10	
Mim0a	**19.74**	29.58	39.68	59.53	109.81	51.67
Mim0b	20.24	30.30	40.27	60.91	114.45	53.24
Mim0c	20.56	31.35	41.95	63.39	116.87	54.82
Mim1Dira	20.00	29.41	38.03	51.34	75.88	42.93
Mim1Dirb	20.37	30.05	38.37	51.19	73.11	42.62
Mim1Dirc	20.72	30.82	39.29	51.74	73.16	43.15
Mim1Orda	25.20	32.00	34.32	36.86	39.07	33.49
Mim1Ordb	20.67	29.36	33.03	36.43	38.76	31.65
Mim1Ordc	26.17	34.85	39.49	43.84	47.22	38.32
Mim1ROCa	20.25	29.65	37.80	51.03	71.92	42.13
Mim1ROCb	20.37	29.84	37.66	50.60	72.49	42.19
Mim1ROCc	20.73	29.31	36.54	48.25	68.92	40.75
Mim2Dira	**19.74**	**27.00**	31.86	39.25	51.74	33.92
Mim2Dirb	20.24	27.58	32.24	39.67	51.91	34.33
Mim2Dirc	20.56	29.02	34.69	42.98	58.87	37.22
Mim2Orda	23.80	29.98	32.75	35.77	37.42	31.94
Mim2Ordb	20.61	28.51	**31.69**	**34.84**	**37.03**	**30.54**
Mim2Ordc	24.75	33.26	38.11	41.83	45.74	36.74
Mim2ROCa	20.43	27.77	32.65	39.10	50.22	34.03
Mim2ROCb	20.58	28.13	33.75	40.83	53.47	35.35
Mim2ROCc	20.83	29.30	34.92	44.46	60.12	37.93
Mim3	20.33	28.51	34.85	44.69	61.65	38.01

Table 3. Accuracies and cost averages of all the models according to the experiment type. Acc is Accuracy.

Model	Balanced		Majority		Minority	
	Acc.	Cost	Acc.	Cost	Acc.	Cost
Mim0a	77.40	23.32	73.62	52.76	**73.65**	78.93
Mim0b	**77.53**	23.46	72.09	64.49	72.32	71.75
Mim0c	76.24	29.53	72.92	63.73	73.01	71.22
Mim1Dira	76.22	20.30	73.36	41.73	70.84	66.76
Mim1Dirb	76.32	20.34	73.06	47.58	69.27	59.94
Mim1Dirc	75.62	22.98	72.98	47.77	70.34	58.69
Mim1Orda	65.38	17.39	69.53	33.69	50.70	49.39
Mim1Ordb	65.49	17.44	70.40	30.51	55.04	46.99
Mim1Ordc	63.88	20.87	68.03	41.35	54.28	52.72
Mim1Roca	76.31	20.28	73.42	46.78	68.27	59.33
Mim1Rocb	76.37	20.29	73.29	46.48	68.71	59.81
Mim1Rocc	75.81	19.52	73.36	45.30	67.74	57.43
Mim2Dira	75.05	**14.59**	74.11	35.63	67.92	51.53
Mim2Dirb	75.06	**14.59**	73.37	39.25	66.46	49.14
Mim2Dirc	73.75	20.96	73.33	40.68	66.77	50.03
Mim2Orda	67.58	16.53	71.32	32.59	54.77	46.71
Mim2Ordb	67.64	16.50	71.56	**30.18**	58.03	**44.94**
Mim2Ordc	65.98	20.39	69.82	39.31	57.17	50.51
Mim2ROCa	75.64	15.50	**73.86**	38.76	65.53	47.83
Mim2ROCb	75.57	15.65	73.21	40.51	66.42	49.90
Mim2ROCc	74.73	20.03	72.81	42.26	67.07	51.48
Mim3	76.08	19.44	73.38	39.98	68.02	54.60
Oracle	81.43	20.61	77.57	54.70	77.55	60.47

Let us see now the effect of working with balanced or non-balanced datasets on the accuracy and cost average(Table 3). Regarding the cost, we observe the same minima as in the overview. The greater increase w.r.t. the cost of the oracle is due to those datasets in which the *skew* acts positively over the majority class. The improvement of cost w.r.t. Mim3, which represents the approach by oversampling, is also meaningful. In the cases in which the *skew* acts positively over the minority class, the reduction of cost is also important for some methods (like Mim2Ordb) but not for all (for instance, Mim1Dira). Concerning accuracy, we do not observe a meaningful decrease. Note that the mimetic technique itself provides models whose accuracy is always lower than the accuracy of the oracle. Nevertheless, as expected, the success ratio in the case of minority class has been the most affected. Finally, the balanced situation shows an intermediate behaviour.

Table 4 shows the AUC of the models depending on the type of datasets. From these results, we can conclude that Mim1 obtains slightly better AUC than the rest of models. The differences are more important for the non-balanced datasets. Comparing and we can see that Mim2Roca is a good option if we look between a compromise between cost and AUC.

Table 4 AUC of all the models according to the experiment type.

Model	Balanced	Majority	Minority
Mim0a	0.811	0.722	0.722
Mim0b	0.812	0.727	0.727
Mim0c	0.804	0.726	0.726
Mim1Dira	0.813	0.731	0.732
Mim1Dirb	**0.814**	**0.733**	**0.733**
Mim1Dirc	0.811	0.731	0.731
Mim1Orda	0.813	0.731	0.732
Mim1Ordb	**0.814**	**0.733**	**0.733**
Mim1Ordc	0.811	0.731	0.731
Mim1Roca	0.813	0.731	0.732
Mim1Rocb	**0.814**	**0.733**	**0.733**
Mim1Rocc	0.811	0.731	0.731
Mim2Dira	0.796	0.707	0.721
Mim2Dirb	0.797	0.710	0.722
Mim2Dirc	0.786	0.706	0.717
Mim2Orda	0.758	0.688	0.693
Mim2Ordb	0.758	0.673	0.684
Mim2Ordc	0.738	0.668	0.684
Mim2ROCa	0.799	0.714	0.721
Mim2ROCb	0.799	0.712	0.722
Mim2ROCc	0.790	0.708	0.717
Mim3	0.807	0.716	0.723
Oracle	0.862	0.798	0.798

5. Conclusions

In this paper, we have presented several methods to derive a class threshold without training or validation data and we have analysed them theoretically and experimentally. As a result we can affirm that the introduced techniques are useful to reduce the costs of the model, being superior to the classical approach based on oversampling. So, not having data is not an obstacle if we want to adapt an existing model to a new cost context.

Theoretically, we have seen that the three approaches are similar if the probabilities are uniform. This is rarely the case. The approach based on ROC analysis is optimal, if the probabilities are well calibrated. However, this is not the case in many situations either. Consequently, the approach based on sorting the probabilities only assumes that the probabilities are reasonably well ordered and works well in this case. In general, this method seems to be better if no assumption is made on the quality of the probabilities.

From all the proposed configurations, Mim2 (a priori) is preferable and the reason can be found in the fact that the oracle is almost always better than its imitation (the mimetic model). So, the search of the threshold can be performed on the oracle more reliably. Secondly, from the three main types for the generation of the invented dataset, the results show that a) (a priori) and b) (balanced) are clearly better than c) (random). Hence, it is important to tune the proportion of classes which are labelled by the oracle. Although the differences between a) and b) are not high, they depend on the configuration and whether the dataset is balanced or not. Thirdly, regarding the method for determining the threshold, we can say that the direct method would work very well if the probabilities would be calibrated. Since this is not generally the case, we have to take the order of the probabilities into account as a more reliable thing and obtain the threshold according to this order (the sort method). This option seems to give the best results w.r.t. costs. Nonetheless, given that the threshold method is affected by the range in which the estimated probabilities can vary, we devised a method based on ROC analysis, and we proposed a threshold derivation based on the newly introduced NROC curves. Although they are worse on costs, they present a good compromise between cost, accuracy and AUC. The recommendation from the general results is that when the goal is to minimise the global cost the preferable configuration is to use the a priori method (i.e. Mim2), with the sort threshold and with the invented data in a balanced way (Mim2Ordb).

As future work it would be interesting to analyse the threshold derivation methods after performing a calibration. In this situation, we think that the method based on ROC analysis can be better than the other two. We have not tried this calibration for this paper since here we have considered a situation with almost no assumptions, in particular we do not have training or validation sets and, hence, we cannot calibrate the probabilities. An additional future work could be to find hybrid techniques between the Ord and ROC methods.

Acknowledgments

This work has been partially supported by the EU (FEDER) and the Spanish MEC under grant TIN 2004-7943-C04-02, ICT for EU-India Cross-Cultural Dissemination Project under grant ALA/95/23/2003/077-054, Generalitat Valenciana under grant GV06/301 and the Vic. de Innovación y Desarrollo (UPV) under project TAMAT ref 5771, SEIT-ANUIES scholarship and license commission of the Dirección General de Institutos Tecnológicos of México.

References

Abe, N., Zadrozny; B., Langford, J. (2004). An Iterative Method for Multi-class Cost-sensitive Learning. KDD'04, August 22–25, Seattle, Washington, USA.

Black C. L. ; Merz C. J. (1998). UCI repository of machine learning databases.

Blanco Vega, R. (2006). Extraction and contextualisation of comprehensible rules from black-box models. Ph.D. Dissertation. Universidad Politécnica de Valencia.

Blanco-Vega, R.; Hernández-Orallo, J.; Ramírez-Quintana M. J. (2004). Analysing the Trade-off between Comprehensibility and Accuracy in Mimetic Models. 7th International Conference on Discovery Science.

Bouckaert, R.; Frank, E. (2004). Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms, LNCS, Volume 3056, Page 3.

Breiman, L.; Friedman, J. H.; Olsen, R. A.; Stone, C. J. (1984) Classification and Regression Trees. Wadsworth International Group.

Chan, P.; Stolfo, S. (1998) Toward scalable learning with non-uniform class and cost distributions. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 164–168.

Domingos, P. (1999). MetaCost: A general method for making classifiers cost sensitive. In Procc of the 5th Int. Conf. on KDD and Data Mining, 155–164. ACM .

Domingos, P. (1997). Knowledge Acquisition from Examples Via Multiple Models. Proc. of the 14th Int. Conf. on Machine Learning, pp: 98-106.

Domingos, P. (1998). Knowledge Discovery Via Multiple Models. IDA, 2(1-4): 187-202.

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. Proc. of the 17th Int. Joint Conf. on A.I.

Estruch, V.; Ferri, C.; Hernandez-Orallo, J.; Ramirez-Quintana. (2003). M.J. Simple Mimetic Classifiers, Proc. of the Third Int. Conf. on ML & DM, pp:156-171.

Lachiche, N.; Flach, P. A. (2003) Improving Accuracy and Cost of Two-class and Multi-class Probabilistic Classifiers Using ROC Curves. ICML 2003: 416-423.

Witten, I. H.; Frank, E. (2005). Data Mining: Practical ML tools with Java implementations. (Second Edition). Morgan Kaufmann.