When Does Imbalanced Data Require more than Cost-Sensitive Learning?

Dragos D. Margineantu
Department of Computer Science
Oregon State University
303 Dearborn Hall
Corvallis, Oregon 97331-3202
margindr@cs.orst.edu

Abstract

Most classification algorithms expect the frequency of examples form each class to be roughly the same. However, this is rarely the case for real-world data where very often the class probability distribution is non-uniform (or, *imbalanced*). For these applications, the main problem is usually the fact that the costs of misclassifying examples belonging to rare classes differ significantly from the costs of misclasifying examples from classes represented in a higher proportion in the data. Cost-sensitive learning studies and provides methods for the design and evaluation of classification algorithms for arbitrary cost functions. This paper outlines an issue that can occur in the imbalanced data setting but has not been studied, according to our knowledge, in the cost-sensitive learning literature—the situation when the class probability distribution on the training data differs significantly from the class probability distribution test data. We will present a brief overview of cost-sensitive learning methods applied on imbalanced data and we will extend the existing theoretical results for the setting in which training and test class priors are different.

Introduction

An increasing variety of application problems have been approached lately using supervised learning techniques.

The model for supervised learning assumes that a set of labeled examples $\langle \mathbf{x}_i, y_i \rangle$ (called training data) is available, where $\mathbf{x}_i$ is a vector of continuous or discrete values called *attributes* and y_i is the *label* of $\mathbf{x}_i$. The model further assumes that there exists an underlying, unknown function, $f(x) = y$ that maps the attribute vectors to the set of possible labels. A learner outputs a hypothesis $h(x)$ which is an approximation of $f(x)$, with respect to some error function (a parametric function measuring the overall accuracy of the predictions).

The labels can be elements of a discrete set of classes $Y = \{y_1, y_2, \ldots, y_k\}$ in the case of *classification*, or elements drawn from a continuous subset of a continuous set (e.g. a continuous subset of the reals) in the case of *regression*.

For the rest of this paper we will concentrate on classification tasks and we will denote with m the number of attributes and with k the number of classes.

Most classification algorithms assume uniform class probability distribution, i.e. they assume that the proportions of examples from each class are roughly equal. On the other hand many real-world applications require classifiers that are trained and tested on non-uniformly distributed (or imbalanced) data. For example, in a life-threatening disease diagnosis task, the number of patients diagnosed as being ill is usually much smaller than the number of patients diagnosed as being healthy, and therefore the data used to train a classifier for automatic diagnosis would be highly imbalanced. Predicting important events in event sequences (Fawcett & Provost 1997; Tjoelker & Zhang 1998), pattern detection in remotely-sensed images (Kubat, Holte, & Matwin 1998) and document classification (Koller & Sahami 1997) are a few other examples of real-world classification tasks in which the data is also imbalanced.

This paper highlights the main reason why class imbalace matters in real-world applications: the fact that the real cost of misclassifying examples is not a uniform function over the set of examples—usually rare examples are very expensive to be misclassified. We will then bring to the attention of the reader a practical situation that has been given little or no attention in previous research: training classifiers on data with different class priors than the priors of the previously unseen (test) examples. We derive a decision-theoretic rule for the optimal prediction in such situations and we present some preliminary experimental results. The paper concludes with a discussion of the results and the future directions of research.

What error function needs to be minimized?

In order to derive a classification scheme for a given task, first of all one needs to know what is the error function that has to be minimized.

The main problem in the case of tasks with imbalanced data is that the error function is an asymmetric function (also called *loss function* or *cost function*) rather than the raw misclassification rate (also referred to as *0/1 loss*). It is usually the case that the misclassification of examples that belong to rare classes induces a high misclassification cost.

Cost-sensitive learning (Turney 1997) studies methods for building (Pazzani *et al.* 1994; Knoll, Nakhaeizadeh, & Tausend 1994; Bradford *et al.* 1998; Kukar & Kononenko 1998; Domingos 1999) and evaluating (Bradley 1997; Provost & Fawcett 1997; Margineantu & Dietterich 2000) classifiers when the error function is different from the *0/1 loss.*

In general, in cost-sensitive learning, the error function may be described either by a $k \times k$ cost matrix C, with $C(i,j)$ specifying the cost incurred when an example is predicted to be in class i when in fact it belongs to class j, or by a k-dimensional cost vector L, with $L(i)$ specifying the cost of misclassifying an example that belongs to class i. It is easy to observe that a cost vector L is equivalent with a cost matrix C in which the diagonal values are equal to 0 and $C(i,j) = L(j)$ for all extra-diagonal values.

The procedure that has been most oftenly applied in learning from imbalanced data is stratification, i.e. changing the frequency of classes in the training data in proportion to the costs specified in the cost vector. Stratification can be achieved either by oversampling or by undersampling the available data. The two main shortcomings of stratification is that it is not straightforward how it can be applied when the error function is represented as a cost matrix (see (Margineantu & Dietterich 1999) for possible solutions and a discussion on this issue) and, second, it distorts the original distribution of the data.

Another possible approach to the data imbalance problem (and to cost-sensitive problems in general) is based on the class probability estimates of the examples, $P(y|x)$. Assuming that we have a procedure that computes good estimates for the class probabilities of the examples, the optimal output of the classification procedure is the class label for which the value of the conditional risk (Duda & Hart 1973) is minimized:

$$h(x) = \underset{y \in Y}{\text{argmin}} \sum_{j=1}^{k} P(y_j|x)C(y, y_j), \quad (1)$$

or, equivalently, for a loss vector L:

$$h(x) = \underset{y \in Y}{\text{argmin}} \sum_{j=1, y_j \neq y}^{k} P(y_j|x)L(y_j). \quad (2)$$

This also assumes that the class frequencies in the training data are the same as the class frequencies that will be encountered on the test data—a condition that sometimes doesn't hold either because of the nature of the process that generates the data or because of the preprocessing of the training data by the means of a stratification procedure.

Different class priors on the training and on the test data

We will further study what is the optimal output of a classifier when the class priors (i.e., the frequency of classes) on the training data are different from the class priors for the test examples. The class priors on the test data are assumed to be known.

Let the probability values that refer to the training data be denoted as P_t and the probability values that refer to the test data be denoted as P_s. The priors of y on the training and on the test data will be denoted as $P_t(y)$ and $P_s(y)$, respectively.

We will assume that within each class the underlying probability density is the same for both the training and the test data: $P_t(x|y) = P_s(x|y)$, and we will also assume that $P_t(x) = P_s(x)$.

From the definition of the conditional probability we get:

$$P_t(y|x) = \frac{P_t(x|y)P_t(y)}{P_t(x)} \quad (3)$$

and, similarly:

$$P_s(y|x) = \frac{P_s(x|y)P_s(y)}{P_s(x)} \quad (4)$$

From (3) and (4) and from the assumptions made we obtain:

$$P_s(y|x) = P_t(y|x)\frac{P_s(y)}{P_t(y)} \quad (5)$$

When we plug this result into (1) we get the expression for the optimal prediction on an unseen (test) example:

$$h(x) = \underset{y \in Y}{\text{argmin}} \sum_{j=1}^{k} \frac{P_s(y_j)}{P_t(y_j)} P_t(y_j|x)C(y, y_j) \quad (6)$$

which becomes:

$$h(x) = \underset{y \in \Theta}{\text{argmin}} \sum_{j=1, y_j \neq y}^{k} \frac{P_s(y_j)}{P_t(y_j)} P_t(y_j|x)L(y_j) \quad (7)$$

for an error function represented by a loss vector L.

Experiments

We have conducted some preliminary tests using three classification methods.

Our first method is *C4.5-avg*, a version of the C4.5 algorithm (Quinlan 1993) modified to accept weighted training examples (this is equivalent to stratification). Each example was weighted in proportion to the average value of the column of the cost matrix C corresponding to the label of the example. This is the average cost (over the training set) of misclassifying examples of this class. (Breiman *et al.* 1984) suggests a similar method for building cost-sensitive decision trees and

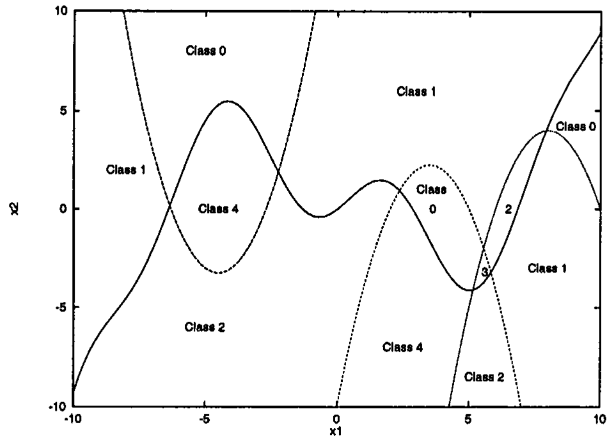

Figure 1: Decision boundaries for the Expf5 data set.

Table 1: Description of the two tasks on which we ran the tests.

Class Frequencies on Training Data	Class Frequencies on Test Data
[0.2,0.2,0.2,0.2,0.2]	[0.21,0.44,0.26,0.001,0.089]
[0.1,0.05,0.15,0.4,0.3]	[0.21,0.44,0.26,0.001,0.089]

(Margineantu & Dietterich 1999) compare this method against other methods for incorporating costs into the decision tree learning algorithm.

The second algorithm that we tested is *BagCost-C4*. *BagCost-C4* employs Bagging (Breiman 1997) over C4.5 decision trees to estimate the class probabilities of the unseen examples and then applies (6) to output a prediction.

Our third method is *Metacost-C4*, which is a version of Metacost (Domingos 1999).*Metacost-C4* estimates the class probabilities of the training examples using Bagged C4.5 trees, relabels them (the training examples) according to (6) and then grows a C4.5 decision tree using the relabeled data.

To compare the results we employed the BDelta-Cost procedure, a cost-sensitive evaluation procedure described in (Margineantu & Dietterich 2000).

We have tested the three procedures on Expf5, an artificial domain with two features and five classes. The decision boundaries of Expf5 are shown in Figure 1.

Table 1 presents the tasks on which we ran the experiments. The first column indicates the class frequencies on the training data and the second column indicates the class frequencies on the test data. For each task the size of the training and test data sets was set to be 1000.

Each experiment involves testing several different cost matrices, C. These were generated randomly based on three different cost models. Table 2 shows the underlying distributions for each of the cost models. Our preliminary experiments were conducted on one randomly selected cost matrix for each cost model.

On both tasks and for all cost models BDeltaCost

Table 2: The cost models used in our experiments. Unif$[a, b]$ indicates a uniform distribution over the $[a, b]$ interval. $P(i)$ represents the prior probability of class i (from the second column in Table 1.

Cost Model	$C(i,j)$ $i \neq j$	$C(i,i)$
M1	Unif$[0, 1000 \times P(i)/P(j)]$	0
M2	Unif$[0, 10000]$	Unif$[0, 1000]$
M3	Unif$[0, 2000 \times P(i)/P(j)]$	Unif$[0, 1000]$

could not reject the null hypothesis when the classifiers built by *MetaCost-C4* and *BagCost-C4* were compared. On the other hand, both *MetaCost-C4* and *BagCost-C4* outperformed the decision tree classifier on all tasks.

Conclusions and Discussion

This paper has emphasized the cost-sensitive nature of the data imbalance problem in classification tasks. We have briefly reviewed some cost-sensitive procedures that are frequently applied in the case of imbalanced data. We have derived a generalization of the rule for optimal class labels for the case in which we have available a procedure that is trained to output good class probability estimates of the example. Further, we experimented with three cost-sensitive procedures. Two of the procedures *MetaCost-C4* and *BagCost-C4* label the examples based on some class probability estimates while the third one is a stratification procedure. The preliminary results show that the two methods based on probability estimates outperform the stratification procedure. This proves again that stratification is not a good method for cost-sensitive learning on imbalanced data. We also believe that the probability estimates computed by Bagging are not very accurate and therefore better probability estimates will produce even better decisions.

References

Bradford, J. P.; Kunz, C.; Kohavi, R.; Brunk, C.; and Brodley, C. E. 1998. Pruning decision trees with misclassification costs. In Nedellec, C., and Rouveirol, C., eds., *Lecture Notes in Artificial Intelligence. Machine Learning: ECML-98, Tenth European Conference on Machine Learning, Proceedings*, volume 1398, 131–136. Berlin, New York: Springer Verlag.

Bradley, A. P. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30:1145–1159.

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

Breiman, L. 1997. Arcing classifiers. Technical report, Department of Statistics, University of California, Berkeley.

Domingos, P. 1999. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 155–164. New York: ACM Press.

Duda, R. O., and Hart, P. E. 1973. *Pattern Classification and Scene Analysis.* John Wiley and Sons, Inc.

Fawcett, T., and Provost, F. 1997. Adaptive fraud detection. *Data Mining and Knowledge Discovery* 1(3).

Knoll, U.; Nakhaeizadeh, G.; and Tausend, B. 1994. Cost-sensitive pruning of decision trees. In Bergadano, F., and DeRaedt, L., eds., *Lecture Notes in Artificial Intelligence. Machine Learning: ECML-94, European Conference on Machine Learning, Proceedings*, volume 784, 383–386. Berlin, New York: Springer Verlag.

Koller, D., and Sahami, M. 1997. Hierarchically classifying documents using very few words. In *Machine Learning: Proceedings of the Fourteenth International Conference*, 170–178. Morgan Kaufmann.

Kubat, M.; Holte, R. C.; and Matwin, S. 1998. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning* 30(2/3).

Kukar, M., and Kononenko, I. 1998. Cost-sensitive learning with neural networks. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence.* Chichester, NY: Wiley.

Margineantu, D. D., and Dietterich, T. G. 1999. Learning decision trees for loss minimization in multi-class problems. Technical report, Department of Computer Science, Oregon State University.

Margineantu, D. D., and Dietterich, T. G. 2000. Bootstrap methods for the cost-sensitive evaluation of classifiers. In *Machine Learning: Proceedings of the Seventeenth International Conference.* San Francisco, CA: Morgan Kaufmann.

Pazzani, M.; Merz, C.; Murphy, P.; Ali, K.; Hume, T.; and Brunk, C. 1994. Reducing misclassification costs. In *Proceedings of the Eleventh International Conference on Machine Learning*, 217–225. Morgan Kaufmann.

Provost, F., and Fawcett, T. 1997. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 43–48. AAAI Press.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning.* San Francisco: Morgan Kaufmann.

Tjoelker, R., and Zhang, W. 1998. A general paradigm for applying machine learning in automated manufacturing processes. In *Conference on Automated Learning and Discovery, CONALD'98. Workshop on Reinforcement Learning and Machine Learning for Manufacturing.*

Turney, P. 1997. Cost-sensitive learning bibliography. Online Bibliography. Institute for Information Technology of the National Research Council of Canada, Ottawa. [`http://ai.iit.nrc.ca/bibliographies/cost-sensitive.html`].