Learning with Cost Intervals

Xu-Ying Liu and Zhi-Hua Zhou

National Key Laboratory of Novel Software Technology
Nanjing University, Nanjing 210093, China
{liuxy, zhouzh}@lamda.nju.edu.cn

Abstract. Existing cost-sensitive learning methods work with unequal misclassification cost that is given by domain knowledge and appears as precise values. In many real-world applications, however, it is difficult to have a precise cost information since the user maybe only knows that one type of mistake is much more severe than another type, yet not possible to give a precise description. We claim that, in such situations, it is more meaningful to work with *cost intervals* instead of a precise cost value. We propose the CISVM method, a support vector machine that can work with cost interval information. Experimental results show that when there is only cost interval information available, CISVM is superior to training a standard cost-sensitive SVM by using minimal cost, mean cost and maximal cost.

Key words: Cost-sensitive learning, cost interval, unequal cost

1 Introduction

In real-world tasks, different classification errors often lead to different losses. For example, in medical diagnosis, the loss of misdiagnosing a patient to be healthy is much more serious than misclassifying a healthy person as being sick, because the former may lead to the loss of a life. Unfortunately, traditional machine learning research assumes that all the classification errors will result in the same loss. Thus, standard classification methods try to minimize the number of errors rather than the total cost. To deal with unequal costs, cost-sensitive learning has attracted much attention [9, 14, 17, 18, 7, 12].

Existing cost-sensitive learning methods work with precise value of misclassification costs. The cost information is given by domain knowledge and appears as precise values. The classifiers will be well tuned to reduce the total cost associated with this particular cost. However, in many real situations, although the user knows that one type of mistake is more severe than another type, it may be difficult for the user to specify a precise cost value. One obvious case is that, cost modelling process is

required to determine the exact cost values in many cost-sensitive applications, such as intrusion detection [10] and risk management [8], but the situations are often too complex to model risk precisely. Another case is, the misclassification costs should be determined by end users, or, costs will change over time while it is not sure how it will change exactly.

In many situations, though precise information is not available, other useful information could be obtained. One of the most common and practical form to present imprecise information is to bound it with an interval. [8] provided an interval form of risk evaluation. An intuitive way to work with cost intervals is to apply existing cost-sensitive learning methods to reduce total cost associated with the median value of the cost interval (or, mean value of the cost interval, when the distribution of the cost is known). Such a solution, however, does not always work because the cost value used in the training process will affect the performance of the trained classifier, and thus will affect the distribution of its test results. The detailed analysis is given in Section 3 and evaluated in experiments.

In this paper, we propose to study the problem of learning with cost intervals and propose a simple method, CISVM, to handle cost intervals. Experimental results show it is better than directly applying standard cost-sensitive support vector machines with minimal cost, mean cost and maximal cost.

The rest of the paper is organized as follows. Section 2 briefly reviews some related work. Section 3 analyzes the problem of learning with cost intervals. Section 4 presents the CISVM method. Section 5 reports the empirical results and Section 6 concludes.

2 Related Work

Current cost-sensitive learning methods can only be applied when precise cost information is given. To be best of our knowledge, there is no methods learning with cost intervals. A related work is [15], which considers the situation of cost changing over time. But it assumes the cost is known at time of classification. In our assumption, true cost is always unknown.

ROC curve [2] has been proposed in order to compare classifiers' performance under imprecise class distributions and/or misclassification costs. AUC (area under ROC curve) based methods can used to produce robust classifier to imprecise misclassification cost. This essentially assumes that nothing whatsoever is known about the relative severity of misclassification cost, a situation which is very rare in real problems. In our problem settings, cost interval is known. Therefore, to use ROC as

performance measure and to use AUC as learning metric would not be a good choice.

3 Problem Analysis

Suppose there are n examples in the training set $S = \{(x_i, y_i)\}_{i=1,...,n}$, as well as a test set S' of size n'. Both S and S' are i.i.d sampled from the true class distribution $Pr(X, Y)$.

The standard classification goal is to find a classifier $h \in H$ from the hypothesis space H to minimize the expected loss on test set S':

$$R^{\Delta}(h) = \int \Delta((h(x_1^{'}), ..., h(x_{n'}^{'})), (y_1^{'}, ..., y_{n'}^{'}))dPr(S') \tag{1}$$

where, Δ is the loss function of h over samples. When Δ can be decomposed linearly into a sum of loss function L over individual examples

$$\Delta((h(x_1^{'}), ..., h(x_{n'}^{'})), (y_1^{'}, ..., y_{n'}^{'})) = \sum_{i=1}^{n'} L(h(x_i^{'}), y_i^{'}) \tag{2}$$

the expression can be simplified to:

$$R^{L}(h) = \int L(h(x^{'}), y^{'})dPr(x^{'}, y^{'}) \tag{3}$$

The empirical risk on the training set S is

$$R_S^L(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x_i), y_i) \tag{4}$$

$R_S^L(h)$ is an estimate of the expected risk $R^{\Delta}(h)$. Discriminative learning methods will select an h to minimize this empirical risk.

In 2-class cost-sensitive classification problem, positive class has misclassification cost C_+, and negative class has cost C_-. We assume positive class has higher cost: $C_+ \geq C_-$. The loss function for a particular example x is in the form of

$$L(h(x), C_+, C_-) = C_+ \times I(h(x) \neq y | y = +) + C_- \times I(h(x) \neq y | y = -). \tag{5}$$

When a cost matrix is multiplied by a positive constant, the optimal decisions are unchanged [9]. Therefore, we can simplify the cost information by setting $C_- = 1$, and $C_+ = C$ ($C \geq 1$). There is only one free variable now. And the loss function becomes

$$L(h(x), y, C) = C \times I(h(x) \neq y|y = +) + I(h(x) \neq y|y = -). \quad (6)$$

The empirical risk over the training set S is:

$$R_S(h, C) = p_+ \times fn \times C + p_- \times fp \quad (7)$$

where, p_+ and p_- is the probability of the positive and negative class, respectively. fn is false negative rate, and fp is false positive rate:

$$\begin{array}{l} fn = p(h \neq y|y = +) \\ fp = p(h \neq y|y = -) \end{array} \quad (8)$$

When cost is imprecise, a cost interval is provided. The misclassification cost of the positive class is therefore a random number in $[C_{min}, C_{max}]$. Since we do not know the true cost value, we hope to achieve the ideal goal

$$h = \arg\min R_S(h, C), \forall C \in [C_{min}, C_{max}] \quad (9)$$

But generally, this ideal goal cannot be achieved. Suppose C is a random variable of distribution υ, with $p(C \in [C_{min}, C_{max}]) \approx 1$. Then, a less ambitious goal is to minimize the expected risk over distribution υ[1]:

$$\begin{array}{rl} ER_S(h) & = E_{C\sim\upsilon}[R_S(h, C)] \\ & = \int R_S(h, C)d\upsilon \\ & = \int (p_+ \times fn \times C + p_- \times fp)d\upsilon \end{array} \quad (10)$$

Note that, Eq. 10 cannot take the expectation of C, i.e.,

$$\begin{array}{rl} ER_S(h) & = \int (p_+ \times fn \times C + p_- \times fp)d\upsilon \\ & \neq p_+ \times fn \times \int Cd\upsilon + p_- \times fp \\ & = p_+ \times fn \times E[C] + p_- \times fp \\ & = R_S(h, E[C]) \end{array} \quad (11)$$

$ER_S(h) = R_S(h, E[C])$ only if fn and fp are independent of C. But this assumption should be checked very carefully.

To clarify this, we need to introduction two concepts – training cost and test cost. Test cost is used to evaluate the total loss of a classifier. It is given by domain knowledge and it is fixed. And training cost is a parameter value provided to the learning algorithm to control the cost-sensitivity of the resulting learner. As we know, cost-sensitive learning

[1] The distribution is the underground truth. It could be totally unknown or could be obtained by domain knowledge

methods gain cost sensitivity by introducing a parameter to bias toward expensive class. For example, sampling-base cost-sensitive methods [18, 17] sample data to make sure that the probability of the expensive class is t times of the less expensive class. Weighting-based cost-sensitive methods [14] give higher weights to the expensive class. Thresholding-based cost-sensitive methods [18, 9, 6] move decision threshold toward the inexpensive class. Cost-sensitive large margin methods design cost-sensitive versions of surrogate loss functions. The loss function of cost-sensitive SVM is

$$I(y = +)(C(1 - yf)_+) + I(y = -)(1 - yf)_+.$$

The loss function of cost-sensitive boosting [12] is

$$I(y = +)e^{-Cyf} + I(y = -)e^{-yf}.$$

All these methods use a parameter to control the bias toward the expensive class. Usually, this parameter is equal to test cost. But it is not a must. The amount of bias should be depend on not only the test cost value, but also the decision boundary. As Brieman et al. [4] stated, training set size, class prior, cost of errors in different classes, and placement of decision boundaries are all closely connected. In a special case where two classes can be perfectly separated, there should be no bias to introduce no matter how large the cost is. Generally speaking, on easy tasks where most examples can be correctly classified, using test cost as bias parameter in the training process will make the expensive class over-biased. Using a smaller value as bias parameter instead could be much better. In this sense, the cost-sensitivity controller, training cost, is a bias parameter. Ciraco et al. [5] and Sheng & Ling [13] observed the best training cost is usually not equal to the test cost.

And since the training cost is a parameter to control bias toward positive class, it will affect the resulting classifier. So, the h optimizing Eq. 7 is a function of training cost C_{train}:

$$\begin{aligned} h(C_{train}) &= \arg\min R_S(h(C_{train}), C_{test}) \\ &= \arg\min p_+ \times fn(h(C_{train})) \times C_{test} + p_- \times fp(h(C_{train})). \end{aligned} \tag{12}$$

And fp and fn are functions of $h(C_{train})$:

$$\begin{aligned} fn(h(C_{train})) &= p(h(C_{train}) \neq y | y = +) \\ fp(h(C_{train})) &= p(h(C_{train}) \neq y | y = -) \end{aligned} \tag{13}$$

Since training cost controls the amount of bias toward expensive class, it should reflect test cost. That is to say, training cost is essentially a

function of test cost: $C_{train} = g(C_{test})$. In fact, $C_{test} = C$[2]. Therefore, fp and fn are functions of C_{test}. And Eq. 10 becomes

$$\begin{aligned} ER_S(h(C_{train})) &= E_{C\sim\upsilon}[R_S(h(C_{train}), C_{test})] \\ &= \int R_S(h(C_{train}), C_{test})d\upsilon \\ &= \int (p_+ \times fn(h \circ g(C_{test})) \times C_{test} + p_- \times fp(h \circ g(C_{test})))d\upsilon \end{aligned} \tag{14}$$

We can see from Eq. 14, the integral should not take over C_{test}. Therefore, $ER_S(h(C_{train})) \neq R_S(h(C_{train}), E[C_{test}])$. Thus, applying standard cost-sensitive learning methods with expected cost is not the best way to handle cost intervals.

4 The CISVM Method

Since the bias towards the positive class will affect $fp(h)$ and $fn(h)$, minimizing the expected loss over distribution υ is not a practical method. Alternatively, it is possible to learn with cost intervals by minimizing the least upper bound loss function of the true empirical risk. This loss function should: (1) be upper bound of the the empirical risk for every possible test cost; (2) be close to the true cost-sensitive loss function as near as possible; (3) has cost-sensitivity, can introduce relatively large enough bias towards positive class.

Large margin methods use convex surrogate loss functions to approximate the true loss function L_0: $L_0 = I(f \neq y)$. SVM [16] has the loss function in the form of

$$L^{SVM} = (1 - yf)_+. \tag{15}$$

The goal of cost-sensitive SVM (CSSVM) [3] is to minimize the following cost-sensitive loss function:

$$L^{CSSVM} = I(y = +)C(1 - yf)_+ + I(y = -)(1 - yf)_+. \tag{16}$$

It is a convex upper bound of the true cost-sensitive loss of Eq. 6.

Cost-sensitive large margin methods have the loss function of the the following general form

$$L = I(y = +)L_+(C) + I(y = -)L-. \tag{17}$$

[2] For the sake of being clear, we use C_{train} as training cost, C_{test} as test cost, though C_{test} is C.

The difference between $L_+(C)$ and L_- controls the bias towards the positive class. So again, we can rewrite Eq. 17 as

$$L(C_{train}) = I(y = +)L_+(C_{train}) + I(y = -)L-. \tag{18}$$

When $C_{train} = C_{max}$, there will be the biggest bias towards positive class. In general, the total loss will increase when a classifier is over biased.

Assumption. Let $h_L(x)$ is the resulting classifier of minimizing cost-sensitive loss function L with training cost x, C^*_{train} be the training cost with the best bias. Then, if $C_{train} > C_{test}$, the following holds with high probability:

$$\begin{aligned} R_S(h_L(C_{train}), C_{test}) &= p_+ \times fn(h_L(C_{train})) \times C_{test} + p_- \times fp(h_L(C_{train})) \\ &> p_+ \times fn(h_L(C^*_{train})) \times C_{test} + p_- \times fp(h_L(C^*_{train})) \\ &= R_S(h_L(C^*_{train}), C_{test}) = R_S(h^*_L, C_{test}) \end{aligned} \tag{19}$$

When this assumption holds, $L(C_{max})$ results in the upper bound of the empirical risk for every C_{test}.

CSSVM's loss function with C_{max} as training cost can't provide a good candidate for least upper bound. Firstly, it is too far from the true loss function. Secondly, when $C_{test} = C_{min}$, the amount of overestimated loss for a false negative will be $(C_{max} - C_{min})(1 - yf)_+$. It will be as much as $2(C_{max} - C_{min})$ when $yf = -1$, which is twice the true overestimated loss.

Here, we consider a new loss function to satisfy the terms in the beginning of this section:

$$L^{CISVM} = I(y = +)(C_{max} - yf)_+ + I(y = -)(1 - yf)_+. \tag{20}$$

The illustration of true loss function, L^{CSSVM} and L^{CISVM} is shown in Fig. 1. Compared with L^{CSSVM}, L^{CISVM} is a better candidate for least upper bound of the true empirical risk because the following reasons: (1) It uses C_{max} as training cost. So, when the assumption holds, it is an upper bound of the the empirical risk for every possible test cost. (2) It is closer to the true loss function than L^{CSSVM}. And when $C_{test} = C_{min}$, the amount of overestimated loss for a false negative will be $C_{max} - C_{min}$ for whatever yf, which is the same as the true overestimated loss. (3) It has cost-sensitivity. The bias towards positive class is $\frac{(C_{max} - yf)_+}{(1-yf)_+}$. When $yf \leq 0$, the bias varies in $[\frac{1}{2}(C_{max} + 1), C_{max}]$.

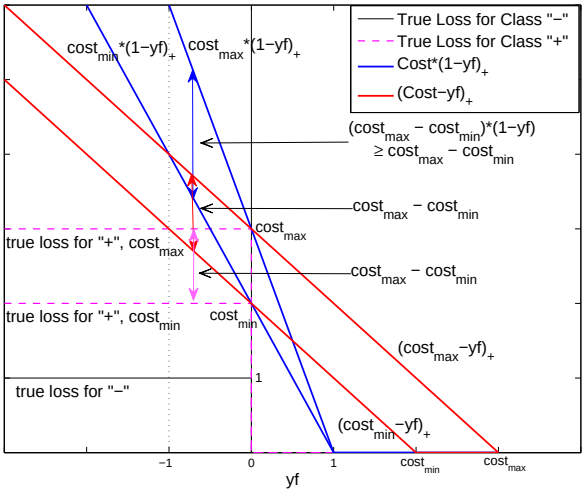

Fig. 1. Illustration of Loss Functions

CISVM Method CISVM method minimizes regularized loss function of L^{CISVM}:

$$\min \frac{1}{2}||w||^2 + \lambda \sum_{i=1}^{n} \xi_i$$

$$s.t. \quad y_i(w^T x_i + b) \geq C_{max} - \xi_i$$
$$\xi_i \geq 0$$

Dual Problem

$$\max \sum_{i=1}^{n} \alpha_i C_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K_{ij}$$

$$s.t. \quad 0 \leq \alpha_i \leq \lambda$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

where, $C_i = C_{max}$ for $y_i = +$, $C_i = 1$ for $y_i = -$. K is kernel matrix.

5 Experiment

5.1 Settings

In the empirical study, we compare three methods: standard SVM, CSSVM and CISVM on 5 UCI data set [1]. There are 3 versions of CSSVM: training with minimum cost (CSSVM$_{min}$), mean cost (CSSVM$_{med}$) and maximum cost (CSSVM$_{max}$).

Because class imbalance may affect cost-sensitive classifier's performance [11], the larger class is random sampled to the same size of another class. The information of these data sets are summarized in Table 1. "Distribution" includes distribution of current data set and the original one. "+" indicates which class is used as the positive class.

We choose 25 cost intervals in $[1, 30]$ with different size. The detailed information is shown in Table 2.

Table 1. Data Set Information

Dataset	Size	Attribute	#Class	Distribution	+
breast-w1	482	9	2	[241,241] ([458 241])	malignant
ionosphere1	351	34	2	[126,126] ([126 225])	g
heart-statlog1	240	13	2	[120,120] ([150 120])	present
sonar1	194	60	2	[97,97] ([97 111])	mine
spambase1	1000	57	2	[500,500] ([2788 1813])	1

Table 2. Cost Intervals

Step	Count	Intervals
3	10	$[1,3], [4,6], [7,9], [10,12], [13,15], [16,18], [19,21], [22,24], [25,27], [28,30]$
5	6	$[1,5], [6,10], [11,15], [16,20], [21,25], [26,30]$
11	5	$[1,11], [5,15], [10,20], [15,25], [20,30]$
15	4	$[1,15], [5,20], [10,25], [15,30]$

Thirty times stratified hold-out experiment are carried out, with 66% as training set and 33% as test set. The average values are recorded. All methods use RBF kernel. Parameters are choose in $\lambda = [0.01, 0.1, 1, 10, 100]$, and kernel parameter σ is $[1/10, 1/2, 1, 2, 10]$ times the mean squared distance of the training set. All parameters are chosen on the first hold-out training data by performing 5-fold cross validation. The parameter resulting in the smallest mean risk is the best.

5.2 Evaluation Criteria

Though in our problem settings, it is assumed the underlying distribution of C_{test} is unknown, we can still use some kind of distribution as true cost distribution to evaluate classifiers. For example, we can assume uniform or normal distribution as true underlying distributions. In this experiment, we assume cost is uniformly distributed in the given interval, and use expected loss as evaluation criteria.

Note that, ROC is not suitable to be used as evaluation criteria here, as we have discussed in Section 2.

5.3 Results

The results on each data set is shown in Fig. 2. X-axis shows on the bottom the odd cost interval values, and on the top the even ones. Y-axis is the ratio of the total loss of each method cost again that of SVM. So the performance of SVM is represented as the line of $y = 1$. The lower the value, the better the performance. A loss ratio above $y = 1$ means it is worse than cost-blind method SVM, which should not happen since the tasks are cost-sensitive. To make the figures clear, the cost intervals on X-axis are sorted to make CSSVM$_{med}$ has ascending loss ratio. Thus, the orders of cost intervals for each data set are different from each other.

The results show that, CSSVM$_{med}$ is the most competitive one to our proposed method CISVM. On *spambase*, CISVM is consistently better than CSSVM$_{med}$. On *heart* and *sonar*, CISVM is better in most cases, but not always. On *breast* and *ionosphere* CISVM is about half times better and half times worse than CSSVM$_{med}$. Note that, on *breast*, CSSVM$_{med}$ is worse than SVM on 9 cost intervals. But CSSVM$_{med}$ is only twice worse than SVM. CSSVM$_{min}$ is the worst among the compared cost-sensitive methods. It is seldom better than CSSVM$_{med}$, but could be often worse. CSSVM$_{max}$ is similar to, but still a slightly worse than CSSVM$_{med}$.

In general, CISVM is the best method to learn with cost intervals.

6 Conclusion

In many real-world applications, it is difficult to get precise cost information. In this paper, we study the problem of learning with cost intervals, and propose a method CISVM by minimizing an approximation of the least upper bound of empirical risk over all test costs in the interval. Experimental results show that, compared with standard cost-sensitive support vector machines, CISVM is able to achieve a better performance.

In future work, we will test the method in real-world applications, especially imbalanced data sets. Also, we will study the influence of the size of the cost intervals.

Acknowledgement

This work was supported by the National Science Foundation of China (60635030, 60721002), the Jiangsu Science Foundation (BK2008018) and the Jiangsu 333 High-Level Talent Cultivation Program.

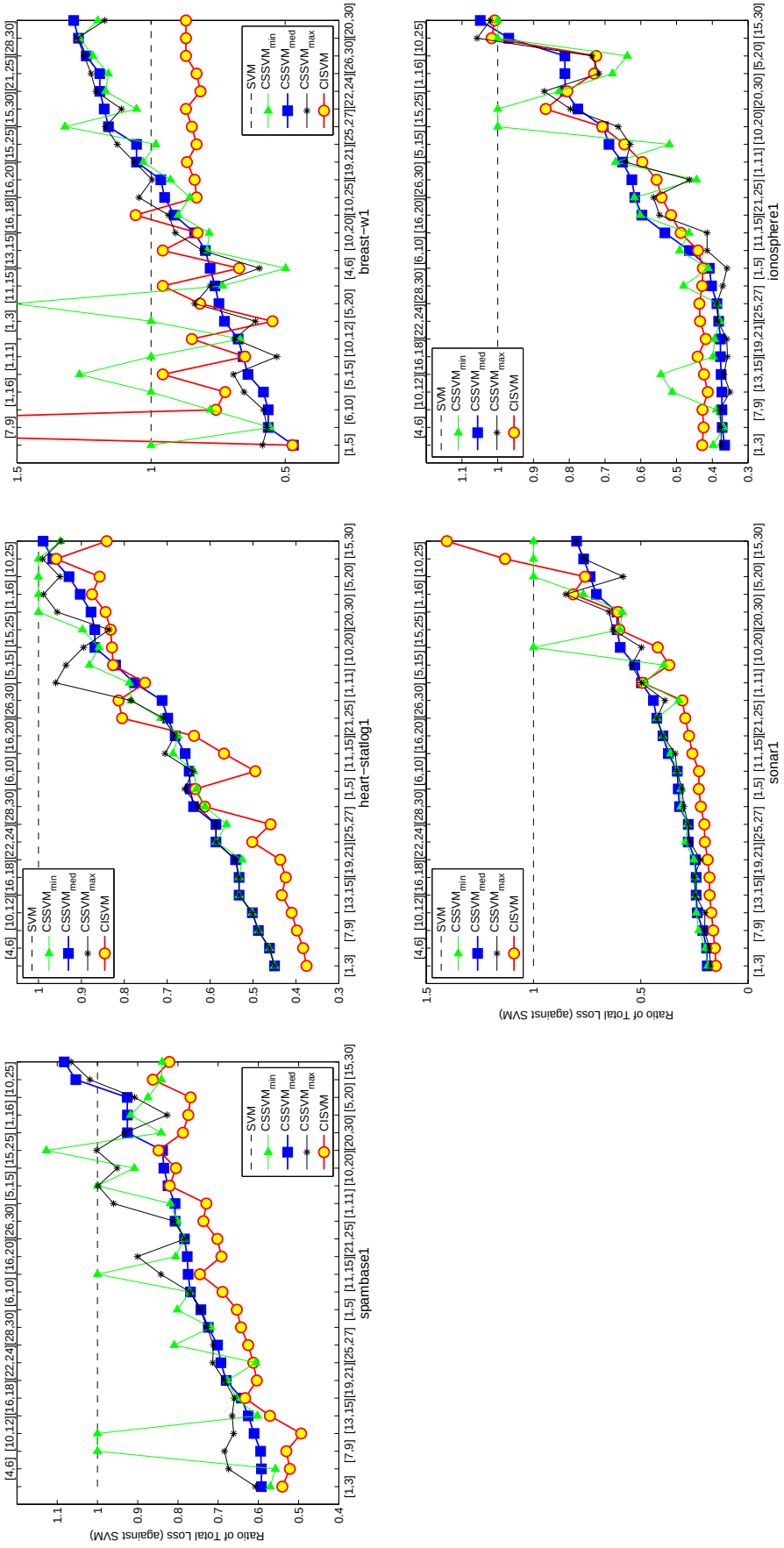

Fig. 2. Results

References

1. C. L. Blake and C. J. Merz. UCI repository of machine learning databases. `http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html`, 1998.
2. A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.
3. U. Brefeld, P. Geibel, and F. Wysotzki. Support vector machines with example dependent costs. In *Proceedings of the 14th European Conference on Machine Learning*, pages 23–34, Cavtat-Dubrovnik, Croatia, 2003.
4. L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.
5. M. Ciraco, M. Rogalewski, and G. Weiss. Improving classifier utility by altering the misclassification cost ratio. In *Proceedings of the 1st International Workshop on Utility-Based Data Mining*, pages 46–52, Chicago, IL, 2005.
6. P. Domingos. Metacost: A general method for kaking classifiers cost-sensitive. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, pages 155–164, San Diego, CA, 1999.
7. C. Drummond and R. C. Holte. Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, 65(1):95–130, 2006.
8. L. Ekenberg and M. Danielson. Handling imprecise information in risk management. Technical Report 95-004, Department of Computer and Systems Sciences, Royal Institute of Technology and Stockholm University, 1995.
9. C. Elkan. The foundations of cost-sensitive learning. In *Proceeding of the 17th International Joint Conference on Artificial Intelligence*, pages 973–978, Seattle, WA, 2001.
10. W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok. Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security*, 10(1-2):5–22, 2002.
11. X.-Y. Liu and Z.-H. Zhou. The influence of class imbalance on cost-sensitive learning: An empirical study. In *Proceedings of the 6th IEEE International Conference on Data Mining*, pages 970–974, 2006.
12. H. Masnadi-Shirazi and N. Vasconcelos. Asymmetric boosting. In *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.
13. V. S. Sheng and C. X. Ling. Thresholding for making classifiers cost-sensitive. In *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, MA, 2006.
14. K. M. Ting. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665, 2002.
15. K. Ming Ting and Z. Zheng. Boosting trees for cost-sensitive classifications. In *Proceedings of the 10th European Conference on Machine Learning*, pages 190–195, Chemnitz, Germany, 1998.
16. V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
17. B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 435–442, Melbourne, FL, 2003.
18. Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006.