Classifying Skewed Data: Importance Weighting to Optimize Average Recall

Andrew Rosenberg[1]

[1]Department of Computer Science, Queens College (CUNY), Flushing, NY, USA
andrew@cs.qc.cuny.edu

Abstract

Promoted in part by its use in the Interspeech Challenges in 2009-2012, Average Recall has emerged as an attractive evaluation measure of classifier performance where the data has a skewed class distribution. In this paper, we show that importance weighting can be used to optimize Average Recall directly. We compare this approach to sampling techniques that have been previously used to classify skewed data. We demonstrate the use of this approach on the Interspeech 2009 Emotion Challenge tasks, and prosodic analysis tasks.

Index Terms: skewed class distributions, prosody, prosodic analysis, emotion classification.

1. Introduction

Many spoken language processing tasks require successful classification of classes that are not equally likely including topic classification, phone recognition, emotion classification and prosodic analysis. A skew in class distributions can pose difficulty for both classifier *training* and *evaluation*.

Density estimation classifiers have the problem that the estimation of the majority class is more reliable than the estimation of minority classes. A related problem can also be observed in loss-minimizing optimization. Since the likelihood of incurring loss is independent from the class value, incurring majority class loss is more likely than minority class loss. Thus loss-optimization approaches will favor reduction of majority class loss leading to decision boundaries that effectively classify majority class tokens potentially at the expense of the classification of minority class tokens.

The impact on evaluation is perhaps even more clearly observable than the impact on training. Detection tasks have the quality that there are many “uninteresting” instances and relatively few “interesting” instances. This is the case, for example, in information retrieval where there are many *irrelevant* documents, and few *relevant* documents. Under a skewed distribution of class values, accuracy (i.e., rate of correct predictions) can lead to unintuitive results. If 99% of all instances are members of the majority class, classification accuracy of 99% can be obtained by a degenerate majority-class classifier. For detection tasks, there are a number of available solutions for classifier evaluation including F-measure, ROC curves, AUC and others. These are based on the intuition that correct classification of majority class (i.e., “noise”) instances is less important than minority class (“signal”) instances. Thus overall performance evaluation can be better accomplished by measuring the false alarms, and misses of the minority class tokens.

There are many tasks that have a skewed class distribution, but do not fit into the detection task metaphor of a “relevant” or ”signal” class and an “irrelevant” or “noise”. For example, inventories of speech acts, emotional states, topics and prosodic categories all demonstrate uneven class distributions. There are fewer attractive options to evaluate classifier performance in multi-class classification tasks with skewed class distributions. Often experimenters will report the F-measure for each class, or the Mutual Information between hypothesized and target class labels. In part due to its use as the evaluation measure in the Interspeech 2009-2012 Challenges, Average Recall has emerged as an attractive measure to evaluate classification performance under skewed class distributions.

In this paper, we describe how importance weighting in loss-based classifier training is able to optimize Average Recall directly. We also demonstrate that the use of sampling techniques, commonly used to improve training with skewed class-distributions, serve to approximate this optimization. In Section 2, we define Average Recall and highlight some attractive qualities of this measure. We present a variety of sampling approaches that have been developed to improve skewed-class training in Section 3. In Section 4, we define importance weighting with particular attention to the impact on SVM training. In Section 5, we highlight some relevant related work. We describe results on experiments on the Interspeech 2009 Emotion Challenge and prosodic event detection and classification tasks in Sections 6.1 and 6.2 respectively. In Section 7, we conclude and provide directions for future work.

2. Average Recall

Average Recall (AR) is the unweighted average recall of each class. Let A be a contingency matrix, where A_{ij} is number of instances of class i that are classified as j and let K be the number of classes, then

$$AR = \frac{1}{K}\sum_{i=1}^{K}\frac{A_{ii}}{\sum_{j=1}^{K} A_{ij}}. \quad (1)$$

Since recall has a range between 0 and 1, Average Recall does as well. Another attractive quality is the baseline values of Average Recall. A majority-class baseline classifier yields an average recall value of $\frac{1}{K}$. (The recall of the majority class will be 1, while all other recalls will be 0.) A random baseline where the true class distribution is used as a prior over the hypotheses, also yields an Average Recall of $\frac{1}{K}$. The Recall of a class i is equal to $\frac{C_i}{N}$, where $C_i = \sum_{j=1}^{K} A_{ij}$ is the number of points in class i and N is the total number of data points in the contingency table. Since $\sum_{i=1}^{K} C_i = N$, $\frac{1}{K}\sum_{i=1}^{K}\frac{C_i}{N} = \frac{1}{K}$. Average Recall was called “Balanced Error Rate” by Read and Cox [1].

Average Recall can also be represented as a loss function. We define Average Recall Loss (ARL) as $ARL = 1 - AR$. Maximizing Average Recall is thus equivalent to minimizing ARL. We can construct an error matrix, E, describing the error contribution of every cell of the contingency table. The diagonal of this error matrix is 0, the off-diagonal cells E_{ij} are equal to $\frac{1}{C_i}$ corresponding to the impact to the recall of class i by a misclassification of a member of class i as class j. That is, the impact to the loss function of misclassifying a data point is equal to the impact to the recall of its class label – specifically $\frac{1}{C_i * K}$. We can compare this error matrix to that which is typically used in classification training which optimizes accuracy. This accuracy error matrix contains $\frac{1}{N}$ in each off-diagonal entry. This suggests that the impact to the overall accuracy of every misclassification is equivalent. While under Average Recall optimization the impact of a misclassification is inversely proportional to C_i, the number of data points in class i. Using this error matrix, we can define ARL as follows. This can be shown to be equal to $1 - AR$.

$$ARL = \sum_{i=1}^{K} \sum_{j=1}^{K} E_{ij} A_{ij}. \tag{2}$$

3. Sampling Techniques

Classifier training is, in general, more effective when training data class distributions are approximately equal. When class distributions are skewed, sampling approaches can be used to modify the training data to generate a more evenly balanced distribution in the training data.

Undersampling [2] selects a sample of majority class instances such that the number of majority class tokens is reduced to the number of minority class tokens. When there are multiple minority classes, undersampling will either generate a sample of majority class instances containing a number of instances equal to that of the *second most* frequent class or undersample members of each class to that of the *least* frequent class. We undersample majority tokens to the size of the second most frequent class in the experiments reported in Section 6. By reducing the number of majority class tokens, the available number of majority-derived "losses" are reduced by a rate proportional to the difference in the size of the classes.

Oversampling generates additional minority class instances to even the class distribution. This can be done by duplicating training instances. SMOTE (Synthetic Minority Oversampling Technique) [3] is an algorithm that imputes minority class instances by the average (times a random scaling factor) of k nearest neighbors for each data point that is duplicated. Under oversampling approaches, the number of times that a minority class instance is duplicated is equal to the ratio of the size of the majority class C_m to the size of the minority class C_i. Thus, the impact of misclassification of minority class instances on the training loss is increased by a factor of $\frac{C_m}{C_i}$. This is obviously seen when minority class points are exactly duplicated; SMOTE and other random oversampling methods, increase the impact of minority instances via *approximate* duplication.

Ensemble sampling [4] is an approach where a set of K undersampled data sets are constructed such that each majority class instance is represented in one data set, while minority class tokens can be repeated across data sets. Using these data sets, K classifiers are trained. At evaluation, the predictions of the K classifiers are fused by majority voting. This provides some of the advantages of both under- and oversampling. However, the impact of each training data point to the overall loss function is consistent with the undersampling approach. Each training instance impacts the loss function at a rate that is approximately inversely proportional to the frequency of its class label. The main advantage of ensemble sampling is that no majority class instances are omitted from training.

All of these approaches have the effect of increasing the importance of minority class instances on the optimized objective function. Thus, each serve to optimize Average Recall without being explicitly formulated to do so.

4. Importance Weighting

Importance weighting is a technique for applying an importance value for each data point in a training set. As discussed in Section 2, the impact that an instance has on the Average Recall is inversely proportional to C_i, the relative frequency of its class label. Thus, assigning a weight to each data point equal to this value equal to $\frac{1}{C_i}$ will serve to optimize AR directly.

Importance weighting impacts classifier objective functions differently. In this paper, we will restrict our experimentation to Support Vector Machines trained with hinge-loss. The SVM optimization is

$$\min \vec{w}^T \vec{w} + C \sum_j \xi_j$$

$$\text{subject to } (\vec{w}^T \vec{x_j} + b) y_j \geq 1 - \xi_j \text{ and } \xi_i \geq 0$$

where C is the slope of the hinge loss function, $\vec{x_j}, y_j$ is a feature vector and label, $\vec{w}$ is a vector normal to the decision boundary, b is the distance between the margin and the decision boundary and ξ_j is a slack variable. Using importance weighting, this function is modified as follows, where γ_j is the importance weight of a data point j

$$\min \vec{w}^T \vec{w} + C \sum_j \gamma_j \xi_j$$

$$\text{subject to } (\vec{w}^T \vec{x_j} + b) y_j \geq 1 - \xi_j \text{ and } \xi_i \geq 0$$

Here, $\gamma_j = \frac{1}{C_j}$ where C_j is the total number of instances with the class label of data point j.

This can be interpreted as adjusting the slope of the hinge loss depending on the class label of the instances, thereby providing increased optimization pressure to correctly classify minority class instances as compared to more frequent classes.

5. Related Work

In this work, the relationship between Average Recall and importance weighting based on the relative size of a class is made explicit. Importance weighting is proposed as a technique to optimize Average Recall directly.

The use of importance weighting for data points, and variable cost functions based on confusion matrix entries has been explored and developed in a number of settings. AdaBoost [13] operates by manipulating the importance of each data point at each iteration of its learning. MetaCost [14] is a generalized wrapper method that allows for arbitrary misclassification costs to be applied during training using any classifier. MetaCost works by training an ensemble of classifiers and manipulating the fusion method based on the user-specified costs. Cardie and Howe [15] addressed the problem of classifying skewed data by adjusting*feature* weights to promote correct detection of minority class instances. Ting [16] used instance-weighting to reduce the number of "high-cost errors". This approach allowed for an arbitrary cost of misclassifying an instance of class j as class i to be applied. Qiao and Liu [17] addressed this question by performing weighted learning, and applying weights based on the relative number of class instances. This approach goes on to automatically learn the optimal weights for each class.

6. Experiments

6.1. Interspeech 2009 Emotion Challenge

The Interspeech 2009 Emotion Challenge (IS09) [5] used the FAU-AIBO corpus as its data set. This is a corpus of German children aged 10-13 speaking to Sony Aibo robots. The data was elicited in a wizard-of-oz experiment, where children were led to believe that they were controlling the robots. Emotional reactions were elicited by causing the robot to behave disobediently. The FAU-AIBO corpus contains 18,216 tokens, where each token represents a "chunk" defined by prosodic/syntactic criteria. The data was labelled by five labelers, with gold standard labels being constructed from majority voting. The original annotation used ten emotional categories. For IS09, two clusterings of these categories were used, a clustering into two classes and another into five. The binary classes were *negative* (subsuming the labels 'angry', 'touchy', 'reprimanding' and 'emphatic') and *idle* (subsuming all other labels). The binary class distribution is modestly skewed in the training data, with 66.3% being *idle* (H=0.922). The 5-way distribution was more heavily skewed. The five classes are *Anger* (8.8%), *Emphatic* (20.9%), *Neutral* (55.9%) *Positive* (6.7%) and *Rest* (7.2%) (a miscellaneous category). The entropy of this class distribution is 1.78; an even distribution would have an entropy of 2.32.

IS09 included a baseline feature set of acoustic descriptors generated using OpenSMILE [6]. This comprised 384 features. The challenge baseline performance was computed using SMOTE. In these experiments, we evaluate each sampling technique, importance weighting and a no-modification baseline as a strategy to optimize Average Recall. These experiments are all based on training and testing using the IS09 partitions of the distributed binary and five-way feature sets. As in the IS09 baseline, each of these experiments uses support vector machines with linear kernels trained with sequential minimum optimization and pairwise coupling. All classifiers are trained with weka [7]. Results can be found in Table 1. We find that using importance weighting leads to performance on this task that is 0.47 worse than SMOTE in the binary classification task. Average Recall does not follow a binomial or normal distribution, so there are not good options for testing statistical significance. However, on this data set, a difference in Average Recall of 0.47 corresponds to correct classification of 56 majority or 28 minority class points in a test set of 8,257 points.

	Binary	5-way
Imp. weight	68.45 (56.3)	40.12
SMOTE (IS09)	68.92 (55.7)	38.20
Under	67.96 (56.2)	37.29
Ensemble	63.83 (47.6)	36.58
No mod. baseline	62.70 (47.6)	28.90
Majority Baseline	50.00 (0.0)	20.00
Dist. Entropy	0.92	1.78
Even Entropy	1.00	2.23

Table 1: *Average Recall and class distribution energy for the IS09 data sets. Minority class (NEG) F-measure in parentheses.*

On the five-way classification task, we find that importance weighting generates an average recall of 40.12, 1.92 *higher* than the SMOTE baseline. The larger skew of the five-way task causes importance weighting to perform better than when skew is more modest, yielding relatively small differences between SMOTE and importance weighting. As an aside, Lee et al. [8], the winning entrant in the Classifier Sub-challenge, obtained 41.57 Average Recall on this task.

These two results indicate that importance weighting has some significant advantages over SMOTE for generating baseline performance. Importance weighting directly optimizes the target measure, Average Recall, rather than approximating this optimization by imputing unseen data. Moreover, training is much faster and has fewer parameters. When training a model with SMOTE, the size of the training data can grow by a significant amount leading to much slower training. Importance weighted SVM training is as fast as unmodified training. SMOTE has a number of parameters that can be tuned to generate optimal performance, while the weights used in importance weighting are dictated by the use of Average Recall as an objective function.

6.2. Prosodic Event Detection and Classification

The ToBI Standard [9] describes American English intonation in terms of **break indices** describing the degree of disjuncture between consecutive words, and high (H) and low (L) **tones** which are associated with phrase boundaries and pitch accents. Pitch accented words are prominent from the surrounding utterance. Nine types of pitch accents are defined H*, !H*, L*, L+H*, L+!H*, L*+H, L+!H*, H+!H* and X*?, when the tone is unclear. Two levels of prosodic phrasing are defined: the intermediate phrase and the intonational phrase. The presence of a prosodic phrase boundary is indicated by perceived disjuncture between two words. Intonational phrases boundaries are defined by the highest degree of disjuncture. Each intonational phrase is comprised of one or more intermediate phrases. Each intermediate phrase has an associated phrase accent, describing the pitch movement leading to the phrase boundary. Phrase accents can have High (H-), downstepped High (!H-) or low (L-) tones. Intonational phrase boundaries have an additional boundary tone, describing the tone at the phrase boundary. These can be high (H%) or low (L%). Intonational phrase boundaries have associated phrase accents *and* boundary tones. There are five valid combinations of phrase accents and boundary tones (PABT pairs): L-L%, L-H%, H-L%, !H-L%, H-H%.

These categories of prosodic events each demonstrate skewed class distributions. For example, across all corpora roughly 75% of pitch accents are H* or !H*. In this section, we break the automatic prediction of ToBI labels into six tasks, and report the performance of SVM classification using each sampling strategy and a no-modification (no-mod) baseline. As in the previous set of experiments, each of these experiments uses support vector machines with linear kernels trained with sequential minimum optimization and pairwise coupling. All classifiers are trained with weka [7]. The six tasks include three binary detection tasks: 1) pitch accent detection, 2) intermediate and 3) intonational phrase detection. The remaining tasks are pitch accent, phrase accent and PABT pair classification. Each of the six classification tasks use only acoustic features based on pitch, intensity, spectral tilt and duration which are extracted using AuToBI [10]. These features are normalized speaker identity using the speech material available in the current utterance file. Each of the six classifiers uses a distinct feature set. Details of the feature sets can be found in Rosenberg (2010) [10].

In each of these experiments, the Boston Directions Corpus (BDC) [11] and the OBJECTS portion of the Columbia Games Corpus (CGC) [12] are used for training data. The ToBI annotated portion of the Boston Directions Corpus is approximately 110 minutes of speech spoken by four speakers. The speakers were asked to complete direction giving tasks. This spontaneous speech was transcribed, and two weeks later the speakers returned to the lab and read their own speech. This material contains 22,458 words. The CGC is a collection of 12 spontaneous task-oriented dydactic conversations. In each session, two subjects played a set of computer games requiring verbal communication to goals of identifying or moving images on a screen. In the OBJECTS game, both players were presented with a screen containing a set of icons. On one player's screen, one object was blinking. The other player's task was to move the object on their screen to the location in which it appeared on the describing player's screen. The OBJECTS data includes 36,515 words, leading to a training set size of 58,973 words. Of this data, 28,999 are accented words, 5507 are intermediate phrase final and not intonational phrase final (thus used in the phrase accent classification task), and 12824 are intonational phrase final (and used in the PABT pair classification task). The results of the detection tasks are reported using Average Recall and F-Measure of the detected event and can be seen in Table 2. The results of the classification tasks (Table 3) are reported using Average Recall. In addition the entropy of the class distribution is reported; the entropy of an even distribution of an equivalent number of classes is included for reference. We find that for the detection tasks, that importance weighting performs approximately as well as any of the sampling approaches. However, we find that ensemble sampling provides the best results on Intermediate Phrase Detection. On these tasks SMOTE generates performance that is consistently lower than importance weighting. In pitch accent classification, where the data is most skewed, importance weighting demonstrates the most best predictions. However, on the other two classification tasks, undersampling and SMOTE generate improved performance. The performance of undersampling on the phrase accent classification task may be explained by a fortunate generation of an effecting sample of the training data. On other tasks, undersampling generates results that are lower than ensemble sampling and importance weighting. The phrase accent/boundary tone task is anomalous; on all other tasks, we find importance weighting to perform approximately as well, or better than SMOTE. Here importance weighting performs worse even than the unmodified training data. In general we find that the improvement due to ensemble weighting over unmodified training is related to how much skew is observed in the training class distribution. The more skewed the distribution the more valuable importance weighting is.

	Pitch Accent	Inter. Phrase	Inton. Phrase
Imp. weight	78.49 (83.08)	75.35 (35.97)	78.63 (69.16)
SMOTE	74.35 (81.60)	73.70 (33.40)	78.40 (68.90)
Under	78.38 (83.07)	64.73 (26.66)	78.36 (68.86)
Ensemble	78.50 (83.09)	75.58 (37.00)	78.36 (68.86)
No mod.	78.50 (83.09)	51.50 (10.64)	78.36 (68.86)
Baseline	50.00 (0.00)	50.00 (0.00)	50.00 (0.00)
Classes	2	2	2
Dist. Entropy	0.999	0.533	0.764
Even Entropy	1.00	1.00	1.00

Table 2: *Prosodic Event Detection results in Average Recall and F-Measure, in parentheses.*

	Pitch Accent	Phrase Accent	Phr.Acc Bound.Tone
Imp. weight	15.00	43.84	23.10
SMOTE	11.11	42.53	29.92
Under	12.84	44.33	23.90
Ensemble	13.80	42.61	30.42
No mod.	11.11	42.26	30.42
Baseline	11.11	33.33	20.00
Classes	9	3	5
Dist. Entropy	1.91	1.33	2.06
Even Entropy	3.17	1.58	2.23

Table 3: *Prosodic Event Classification results in Average Recall*

7. Conclusions and Future Work

In this paper, we show that importance weighting allows for direct optimization of Average Recall. Sampling techniques that are often used to improve classification accuracy under skewed class distributions can be interpreted as optimizing Average Recall by increasing the relative importance of minority class tokens. Sampling approaches have some needling problems. Undersampling discards potentially valuable data, while oversampling, including SMOTE, is training a classifier on unobserved, imputed data. We find that importance weighting is able to train classifiers that, in general, achieve higher Average Recall than sampled classifiers, and train faster than SMOTE by avoiding any increase to the size of the training data.

In future work, AuToBI will incorporate this approach to improve prosodic event detection and classification. In this work all experiments used SVMs as the underlying classifier. It will be valuable to confirm whether or not these results are consistent across a variety of classification algorithms, or if they are specific to some aspect of the SVM.

8. References

[1] I. Read and S. Cox, "Automatic pitch accent prediction for text-to-speech synthesis," in *Interspeech*, 2007.

[2] G. M. Weiss and F. Provost, "The effect of class distribution on classifier learning," Department of Computer Science, Rutgers University, Tech. Rep., 2001.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[4] R. Yan, Y. Liu, R. Jin, and A. Hauptmann, "On predicting rare cases with SVM ensembles in scene classification," in *ICASSP*, 2003.

[5] B. Schuller, S. Steidl, and A. Batliner, "The INTERSPEECH 2009 emotion challenge," in *Interspeech*, 2009.

[6] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE - the Munich versatile and fast open-source audio feature extractor," in *ACM Multimedia*, 2010.

[7] I. Witten, et al., "Weka: Practical machine learning tools and techniques with java implementation," in *ICONIP/ANZIIS/ANNES*, 1999, pp. 192–196.

[8] C.-C. Lee, E. Mower, C. Busso, S. Lee, and S. Narayana, "Emotion recognition using a hierarchical binary decision tree approach," in *Interspeech*, 2009.

[9] K. Silverman, et al., "Tobi: A standard for labeling english prosody," in *ICSLP*, vol. 2, 1992, pp. 12–16.

[10] A. Rosenberg, "AuToBI – a tool for automatic tobi annotation," in *Interspeech*, 2010.

[11] C. Nakatani, J. Hirschberg, and B. Grosz, "Discourse structure in spoken language: Studies on speech corpora," in *AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, 1995.

[12] A. Gravano, "Turn taking and affirmative cue words in task-oriented dialog," Ph.D. dissertation, Columbia University, 2009.

[13] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *ICML*, 1996.

[14] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *KDD*, 1999.

[15] C. Cardie and N. Howe, "Improving minority class prediction using case-specific feature weights," in *ICML*, 1997, pp. 57–65.

[16] K. M. Ting, "An instance-weighing method to induce cost-sensitive trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 3, 2002.

[17] X. Qiao and Y. Liu, "Adaptive weighted learning for unbalanced multicategory classification," *Biometrics*, vol. 65, pp. 159–168, 2009.