Training HMM/ANN Hybrid Speech Recognizers by Probabilistic Sampling

László Tóth[1], András Kocsor[1]

[1] Research Group on Artificial Intelligence
H-6720 Szeged, Aradi vértanúk tere 1., Hungary
{tothl, kocsor}@inf.u-szeged.hu

Abstract. Most machine learning algorithms are sensitive to class imbalances of the training data and tend to behave inaccurately on classes represented by only a few examples. The case of neural nets applied to speech recognition is no exception, but this situation is unusual in the sense that the neural nets here act as posterior probability estimators and not as classifiers. Most remedies designed to handle the class imbalance problem in classification invalidate the proof that justifies the use of neural nets as posterior probability models. In this paper we examine one of these, the training scheme called probabilistic sampling, and show that it is fortunately still applicable. First, we argue that theoretically it makes the net estimate scaled class-conditionals instead of class posteriors, but for the hidden Markov model speech recognition framework it causes no problems, and in fact fits it even better. Second, we will carry out experiments to show the feasibility of this training scheme. In the experiments we create and examine a transition between the conventional and the class-based sampling, knowing that in practice the conditions of the mathematical proofs are unrealistic. The results show that the optimal performance can indeed be attained somewhere in between, and is slightly better than the scores obtained in the traditional way.

1 Introduction

Most machine learning algorithms are prone to inferior performance when the training data is imbalanced, that is when the number of training examples accessible from the various classes is significantly different. In such cases it is frequently observed that the classifier is biased towards predicting the more common classes, performing worse on the rarer classes. Although the exact explanation of this behavior may differ from algorithm to algorithm (see [9] for general reasons), in the hope of an improvement it is always possible to alter the effective class frequencies by presenting more examples from the rarer classes to the learning algorithm. These methods come under the general name of "resampling techniques" [9]. (See the material of the workshops [4] and [5] for more details on techniques proposed to handle class imbalance.)

The class imbalance problem is also present in speech recognition because the natural distribution of speech sounds (phones) is not uniform. However, the solutions proposed by the machine learning community are not necessarily applicable here. This is because most machine learning papers dealing with the topic focus on classification performance, while in speech recognizers the sub-unit models are used as probability

estimators. In particular, the so-called "Hidden Markov Model/Artificial Neural Net (HMM/ANN) hybrid recognizers" [2] apply ANNs to estimate the posterior probabilities of the classes. This is made possible by a nice theoretical proof which shows that, under ideal conditions, ANNs estimate the class posteriors [1]. In practice, however, the class imbalance of the training set can lead to inaccurate estimates. A natural idea is to apply the resampling techniques, but these invalidate the proof, so their application is theoretically questionable. In this paper we examine one peculiar resampling method, the "probabilistic sampling" training technique recommended by Lawrence et al. [6], and argue that it is still usable in training ANNs for HMM/ANN hybrids. First, in Section 2 we point out that theoretically it forces the network to estimate scaled class-conditional probabilities instead of class posteriors and this poses no real problem as the recognizer can be easily modified to work with these. Then we show experimentally in Section 3 that when the recognizer is built on a net trained by probabilistic sampling, it yields the same good or slightly better performance than with the conventional training. The paper rounds off with some conclusions and remarks in Section 4.

2 HMM/ANN Hybrids

Several ways of applying ANNs to speech recognition have been proposed (see [7] or [3] for a review), but the most popular of these is the "hybrid HMM/ANN" paradigm of Bourlard et al. [2]. This approach exploits the fact that, under ideal conditions, ANN classifiers approximate the class posteriors. That is, denoting the space of the local feature vectors by X and the set of class labels by C, we can use them to estimate $P(C|X)$. In the hybrid framework the HMM states play the role of the classes of the ANN, and the states usually directly correspond to phone classes. The HMM framework requires the class-conditionals $P(X|C)$, which can be calculated from the posteriors by Bayes' rule as $P(X|C) = P(C|X) \cdot P(X)/P(C)$. From the HMM optimization point of view $P(X)$ is a constant scaling factor and can be ignored. So the HMM/ANN hybrids work with $P(C|X)/P(C)$, which thus gives an estimate of $P(X|C)$ to within a scaling factor. The $P(C|X)$ values are produced by an ANN, and the $P(C)$ values are obtained by a simple frequency counting of the class labels over the training corpus.

3 Probabilistic Sampling

Let us now examine why and when ANNs estimate the class posteriors, and what happens if training is performed by probabilistic sampling. Let us assume that the network has K outputs denoted by y_k $(k = 1, ..., K)$, and that it is trained by minimizing the sum-of-squares error [1]. We will also assume that the training data is sampled in such a way that its distribution follows the real distribution $P(X)$ of the data points over X. Under these conditions it can be shown that if the size of the training data is allowed to go to infinity, the error function can be written as

$$E = \frac{1}{2} \sum_k \int [y_k(x) - < t_k | x >]^2 \, P(x) dx + B, \tag{1}$$

[1] A similar proof exists for the minimum cross-entropy error criterion as well [1].

where B is a constant that is not important here, and $< t_k|x >$ is the conditional average of the target values t_k at x [1]. Obviously, Eq. (1) takes its minimum when $y_k =< t_k|x >$. Now, if the network structure and the labelling of the training data follow the 1-of-K coding scheme (that is t_k takes a value of 1 for the correct class output and 0 for the rest), it is easy to show that $< t_k|x >$ approximates $P(c_k|x)$ (again assuming a representative sampling and an infinite amount of sample data at point x).

Examining Eq. (1) more closely, we see that at any point x of the input space X it is $< t_k|x >$, the local ratio of positive and negative examples from class c_k, that determines the optimal value for y_k. The local errors of these estimates are in turn weighted by $P(x)$, which forces the network to give a closer approximation in those regions of the input space where the density of input data is high, and permits it to give a poorer approximation in regions where the data density is lower. If class labels correlate well with certain regions of the input space X (which we may assume, otherwise the learning task would be insoluble), then the data density will be lower in those regions where the sparsely represented classes lie. This provides the main reason why the network will perform worse on these classes.

This observation leads to the idea of altering the effective class frequencies by presenting more examples from the rarer classes to the learner. In practice, of course, we usually have no way of generating further samples from any class, so resampling is simulated by replicating some of the samples of the rarer classes. An extreme case of this is when the training data set is manipulated so that it contains the same amount of training examples from each class. When training an ANN with the backpropagation algorithm, there is of course no need to really replicate the samples: only the algorithm has to be modified slightly. Usually the training data items are presented to the algorithm in a random order, that is at each iteration a data item is randomly chosen from the full database. We will refer to this method as "full sampling". A possible alternative is to first choose a class at random, and then randomly pick a training sample from the samples belonging to this class. We will call this general, two-step sampling scheme "probabilistic sampling" [6], and the special case when each class is chosen with uniform probability "uniform class sampling". In general, however, the choice of the class can follow any distribution, not just a uniform one. For example, if class k is chosen with probability $P(c_k)$, that is its own prior probability, then the two-step sampling approach will be practically equal to the traditional one-step full sampling scheme. This will allow us to generate a continuum between full sampling and uniform class sampling by linearly interpolating the probability of class c_k between $P(c_k)$ and $\frac{1}{K}$.

Let us now discuss how the optimum of the error function of Eq (1) changes when using uniform class sampling instead of full sampling. We will see that manipulating the class frequencies influences both the global data distribution and the local conditional averages. First let us examine the data distribution, which was originally written as

$$P(X) = \sum_k P(X|c_k)P(c_k). \tag{2}$$

The manipulation of the class frequencies can be formalized by weighting the terms as

$$P'(X) = \sum_k P(X|c_k)P(c_k)W_k, \tag{3}$$

where W_k are class-dependent weights. From this we can see that modifying the class frequencies changes the focus of the error function, as it modifies $P(X)$. If class labels correlate well with certain regions of the input space, then giving more samples from the sparse classes indeed corresponds to giving more samples from the low data density regions, thus forcing the net to give a better approximation in these areas.

However, the local conditional probabilities are also influenced by this weighting. Clearly, the new $P'(c_k|X)$ values can be written as

$$P'(c_k|X) = \frac{P(X|c_k)P(c_k)W_k}{\sum_j P(X|c_j)P(c_j)W_j}. \tag{4}$$

We can think of the denominator as a normalizing factor required to make the local estimates add up to one. In the case of uniform class sampling W_k is inversely proportional to $P(c_k)$ and cancels it out, so overall the $P'(c_k|X)$ values will be proportional to $P(X|c_k)$. These will be the local targets of the network, so we can say that with uniform class sampling the neural network learns the class-conditionals $P(X|c_k)$ within a scaling factor. This causes no problem when integrating the network into the HMM framework, and in fact makes it even simpler: the division by the class priors $P(c_k)$ can be omitted, and the scaling factor will not affect the final maximization process.

4 Experimental Results

All the results presented in this paper were obtained using the MTBA Hungarian Telephone Speech Database [8]. This is the first Hungarian speech corpus that is publicly available and has a reasonably large size. The most important data block of the corpus contains the recordings of phonetically balanced sentences that were read out aloud by 500 speakers. Recordings were made via mobile and line phones with the speakers varying both in age and gender. All the sentences were manually segmented and labelled at the phone level, and these manually allocated phone labels served as target classes when training the neural net. Altogether 58 different phonetic symbols occur in the database, but after fusing certain rare allophones we worked with only 52 phone classes in the experiments.

For training purposes 1367 sentences were selected from the corpus. The word recognition tests reported here were performed on another block of the database that contains city names. All the 500 city names (each pronounced by a different caller) were different. From the 500 recordings only 431 were employed in the tests, as the rest contained significant non-stationary noise or were misread by the caller. All words were assumed to have equal priors in the word recognition tests.

For acoustic preprocessing we applied the Hvite module of the well-known Hidden Markov Model Toolkit (HTK) [10]. We used the most popular preprocessor configuration, that is we extracted 13 MFCC coefficients along with the corresponding delta and delta-delta values, thus obtaining the usual 39-element feature vector [10]. For recognition we used our own HMM/ANN decoder implementation, which was earlier found to have a performance similar to that of the standard HTK recognizer.

The neural net used in the system contained 150 sigmoidal hidden neurons and a softmax output layer. Training was performed by conventional backpropagation. Be-

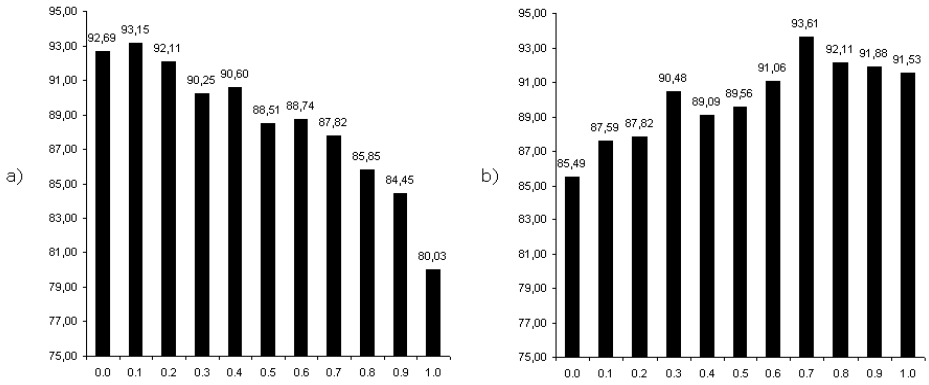

Fig. 1. Word recognition accuracies(%) as a function of λ, with and without division by the priors.

sides comparing the full sampling and uniform class sampling methods, we decided to create a transition between them by making the algorithm select class c_k with a probability $(1-\lambda)P(c_k)+\lambda\frac{1}{K}$, and tested it with various λ values between 0 and 1. We did so for purely empirical reasons. It should not be forgotten that the whole investigation here originated from the observation that the mathematical proof regarding the estimation of the posteriors assumes ideal conditions, and that in practice problems with imbalanced classes were reported. Our argument of Section 3 regarding the estimation of scaled class-conditionals also assumes ideal conditions that do not hold in reality. So while full sampling tends to behave poorly on rarer classes, uniform class sampling may do just the opposite due to over-compensation. This is why it seemed practically justified to create a transition between the two extremes.

As regards division by the class priors, we argued that theoretically it is required when using full sampling and not when using uniform class sampling. However, it is not obvious whether we should use it when the training scheme is somewhere in between. Furthermore, there is evidence that under certain conditions even the conventional model may not require this division [2]. Owing to these uncertainties, we decided to always run the recognizer with the division factor and without it.

The stopping criterion is always a critical issue with every gradient-based algorithm. With our system we have the long-known observation that a certain fixed number of iterations (with a gradually decreased learning rate) produces a nearly optimal solution which cannot be significantly improved either by further iterations or subtle training criteria. However, because uniform class sampling changes the distribution of the data, we could not be sure that the usual amount of iterations were enough in this case. So in each case we allowed two further rounds of 10 iterations. The results reported are the averages of the three scores obtained after the three iteration cycles. We should mention here that these never differed significantly, their deviation always being around 1-1.5%, which can be attributed to the random factors present in the whole training process.

Figure 1 shows the recognition results for different λ values, both with and without division by the priors. Clearly, a λ around 0.1 seems optimal when dividing by the

priors, and a λ of 0.7 resulted in the best results when no division by the priors was applied. These are both better than the corresponding results at $\lambda = 0.0$ and $\lambda = 1.0$ which should have performed the best, according to the proofs discussed in Section 3. This justifies the point that in practice it is worth using the probabilistic sampling scheme for the training of ANNs of HMM/ANN hybrids as it can bring about a modest improvement over the conventional method ($\lambda = 0.0$, division by the priors).

5 Conclusions

This paper investigated the feasibility of the probabilistic sampling training scheme for the training of the ANN components of HMM/ANN hybrid speech recognizers. First we examined uniform class sampling, which is a special case of probabilistic sampling. We argued that although it invalidates the a posteriori probability proof of the conventional training scheme, it is still usable because it gives estimates of the class-conditional probabilities (within a scaling factor) and, in fact, the recognition system requires just these anyway. Second, we suspected that in practice it might be worth interpolating between the conventional full sampling and uniform class sampling, as the mathematical proofs made unrealistic assumptions. In the experiments we indeed found that the optima are somewhere in between – around $\lambda = 0.1$ and $\lambda = 0.7$ respectively, depending on whether we divide by the class priors or not. In both cases our results were slightly better that those obtained by the conventional approach ($\lambda = 0$, division by the priors). This justifies our use of the proposed training scheme in HMM/ANN hybrids.

References

1. Bishop C. M.: Neural Networks for Pattern Recognition. Clarendon Press (1995)
2. Bourlard, H. A., Morgan, N.: Connectionist Speech Recognition – A Hybrid Approach. Kluwer Academic (1994)
3. Bourlard, H. A., Morgan, N.: Hybrid HMM/ANN Systems for Speech Recognition: Overview and New Research Directions. In: Giles, C. L. and Gori, M. (eds.): Adaptive Processing, LNAI 1387, pp. 389-417, (1998)
4. Chawla, N. V., Japkowicz, N. and Kolcz, A. (eds.).: Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Data Sets. http://www.site.uottawa.ca/ nat/Workshop2003/workshop2003.html (2003)
5. Japkowicz, N. (ed.): Proceedings of the AAAI'2000 Workshop on Learning from Imbalanced Data Sets. AAAI Tech. Report WS-00-05 (2000)
6. Lawrence, S., Burns, I, Back, A, Tsoi, A. C., Giles, C. L.: Neural Network Classification and Prior Class Probabilities. In: Tricks of the Trade: Lecture Notes in Computer Science State-of-the-Art Surveys, ed. Orr, G., Müller, K. R. and Caruana, R., Springer, pp. 299-314. (1998)
7. Trentin, E., Bengio, Y., Furnlanello, C., De Mori, R.: Neural Networks for Speech Recognition. In: De Mori (ed.): Spoken Dialogues with Computers, Academic Pr., pp. 311-361. (1998)
8. Vicsi, K, Tóth, L., Kocsor, A., Csirik, J.: MTBA – A Hungarian Telephone Speech Database. Híradástechnika, Vol. LVII, No. 8 (2002) 35- 43 (in Hungarian)
9. Weiss, G. M., Provost, F.: The Effect of Class Distribution on Classifier Learning: An Empirical Study. Tech. Report ML-TR-44, Dep. Comp. Sci., Rutgers Univ. (2002)
10. Young, S. et al.: The HMM Toolkit (HTK) – software and manual. http://htk.eng.cam.ac.uk