Perceptron-based Class Verification

Michael Gerber, Tobias Kaufmann, and Beat Pfister

Speech Processing Group
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
gerber@tik.ee.ethz.ch

Abstract. We present a method to use multilayer perceptrons (MLPs) for a verification task, i.e. to verify whether two vectors are from the same class or not. In tests with synthetic data we could show that the verification MLPs are almost optimal from a Bayesian point of view. With speech data we have shown that verification MLPs generalize well such that they can be deployed as well for classes which were not seen during the training.

1 Introduction

Multilayer perceptrons (MLPs) are successfully used in speech processing. For example they are used to calculate the phoneme posterior probabilities in hybrid MLP/HMM speech recognizers (see for example [1]). In this case their task is to output for every phoneme the posterior probability that a given input feature vector is from this phoneme. They are thus used to *identify* a feature vector with a given phoneme. Expressed in more general terms the MLPs are used for the *identification* of input vectors with a class from within a closed set of classes.

There are applications however, where the identification of input vectors is not necessary but it has to be *verified* whether two given input vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are from the same class or not. In Section 2 we present two verification tasks in the domain of speech processing. In this work we show that MLPs have the capability to optimally solve verification problems. Furthermore we have observed in a task with real-world data that the verification MLPs can even be used to discriminate between classes which were not present in the training set. This is an especially useful property for two reasons:

- The verification MLP is usable for an open set of classes.
- Since we do not need training data from the classes present in the application but can collect training data from other classes which have the same classification objective (e.g. classifying speakers). Therefore we can build a training set of a virtually unlimited size.

In Section 2 we present the motivation for our approach to class verification and outline how MLPs can be used for that purpose. The structure and training of our verification MLPs is described in Section 3. Our evaluation methods are described in Section 4. In order to test whether verification MLPs are capable

of performing the verification task in an optimal way from a Bayesian point of view we made experiments with synthetic data. These experiments and their results are described in Section 5. The results of experiments with speech data are shown in Section 6. Finally, our conclusions are summarized in Section 7.

2 Motivation

Our method to decide whether two speech signals are spoken by the same speaker or not includes the following 3 steps: First equally worded segments are sought in the two speech signals. This results in a series of frame pairs where both frames of a pair are from the same phoneme. In a second step for each frame pair the probability that the two phonetically matching frames come from the same speaker is computed. Finally, the global indicator that the two speech signals were spoken by the same speaker can be calculated from these frame-level probabilities. See e.g. [2] for a more detailed description of the speaker-verification approach. We used the verification MLPs for the following two tasks:

- For seeking phonetically matching segments in two speech signals we use a phonetic probability matrix. This matrix is spanned by the two signals and every element $P_{ij}(x_{1i}, x_{2j})$ gives the probability that frame i of signal 1 given as feature vector x_{1i} and frame j of signal 2 given as feature vector x_{2j} are from the same phoneme. The probabilities $P_{ij}(x_{1i}, x_{2j})$ are calculated by an appropriately trained verification MLP.
- For every pair of phonetically matching frames we use a verification MLP to calculate a score which stands for the probability that the two frames are from the same speaker. In this case we use a MLP which was trained with data from speakers which are not present in the test. Therefore we make use of the generalization capability of the verification MLP.

3 Verification MLP

Since the MLP has to decide whether two given input vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are from the same class the MLP has to process vector pairs rather than single vectors. The target output of the MLP is o_s if the two vectors of the pair are from the same class and o_d if they are from different classes. The vectors are decided to belong to the same class if the output is closer to o_s and to different classes otherwise.

The sizes of the 3-layer perceptrons used for the experiments described in Sections 5 and 6 are as follows:

	input size	1^{st} hidden layer	2^{nd} hidden layer	output layer
synthetic data	$2 \cdot 2...5$	20 (tanh)	10 (tanh)	1 (tanh)
phoneme verification	$2 \cdot 26$	80 (tanh)	35 (tanh)	1 (tanh)
speaker verification	$2 \cdot 16$	70 (tanh)	18 (tanh)	1 (tanh)

The verification MLPs were trained by means of the backpropagation algorithm. The weights were randomly initialized and a momentum term was used during the training. For a hyperbolic tangent output neuron a good choice for the output targets is $o_s = 0.75$ and $o_d = -0.75$ such that the weights are not driven towards infinity (see for example [3]). With these settings we experienced that at the beginning of the training the difference between desired and effective output decreased quite slowly but that the training was never stuck in a local minimum.

4 Performance evaluation

In order to evaluate a verification MLP, we measure its verification error rate for a given dataset and compare it to a reference error rate which is optimal in a certain sense. By formulating our verification task as a classification problem, we can use the Bayes error as a reference. The Bayes error is known to be optimal for classification problems given the distribution of the data.

To reformulate a verification task as a classification problem, each pair of vectors is assigned one of the following two groups:

G_S	group of all vector pairs where the two vectors are from the same class
G_D	group of all vector pairs where both vectors are from different classes

Provided that the same classes which are present in the tests are used to estimate the distributions of G_S and G_D, the Bayes error is optimal, since the two distributions are modeled properly. Otherwise there is a mismatch which leads to ill-modeled G_S and G_D and thus the Bayes classifier is not necessarily optimal any more.

In the case of synthetic data it is possible to calculate the Bayes verification error since the data distributions are given in a parametric form. For real-world problems the data distributions are not given in a parametric form and hence the Bayes verification error can't be computed directly. In this case we can use a k nearest neighbor (KNN) classifier to asymptotically approach the Bayes error as described below.

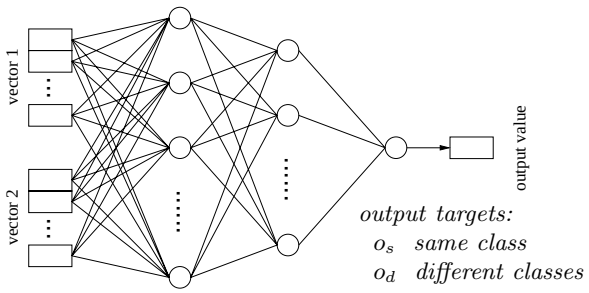

Fig. 1. Structure of the verification MLPs.

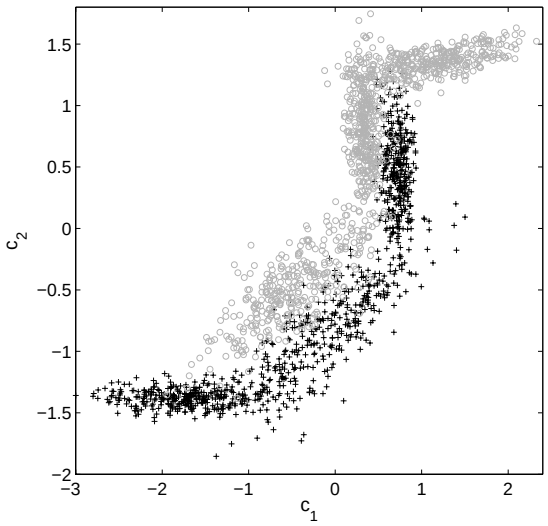

Fig. 2. Synthetic data: 2 classes with 2-dimensional non-Gaussian distributions.

The KNN approach is a straightforward means of classification. The training set for the KNN algorithm consists of training vectors with known classification $(a_{tr,i}, b_{tr,i})$ where $a_{tr,i}$ is the training vector and $b_{tr,i}$ is its associated class. A test vector $a_{tst,j}$ is classified by seeking the k nearest training vectors $a_{tr,i}$ and it is assigned to the class which is most often present among the k nearest neighbors. The KNN classifier is known to reach the Bayes error when an infinite number of training vectors is available (see e.g. [4]) and is therefore a means to approximate the Bayes error if the data distributions are not known in a parametric form.

5 Experiments with synthetic data

The aim of the experiments with synthetic datasets, i.e. datasets with known data distributions, was to test if the verification MLP achieves the lowest possible verification error from a Bayesian point of view. The data sets had 2 to 4 classes and were 2- to 5-dimensional. We illustrate these investigations by means of an experiment with a 2-dimensional dataset with 2 classes that are distributed as shown in Figure 2.

The number of training epochs which were necessary to train the verification MLP depended largely on the type of the dataset. We observed the following dependencies:

– If only a few features carried discriminating information and all other features were just random values the verification MLP learned quickly which features were useful and which ones could be neglected.

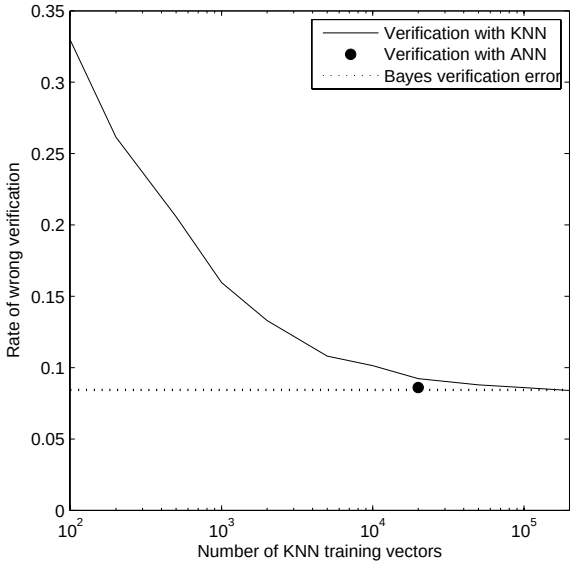

Fig. 3. Class verification error for the test set as shown in Figure 2: the KNN verification error is shown in function of the training set size. As expected, with increasing size it approximates the Bayes limit which is indicated by the dotted line. The error rate of the verification MLP is close to the Bayes error.

– The shape of the distributions strongly influenced the number of epochs that were necessary for the training. For example, two classes distributed in two parallel stripes or classes that had a non-linear Bayes decision boundary, such as those shown in Figure 2, required many epochs.

Figure 3 shows the error rates of different verification methods for data distributed as shown in Figure 2. It can be seen that the error of the verification MLP is almost as low as the Bayes error. Note that the MLP was trained with a fixed number of 20'000 vector pairs. We are only interested in the best possible verification error for a given task and not in the verification error in function of the number of training vectors (see Section 1). Therefore the MLP training set was chosen as large as necessary.

For all investigated datasets the verification error achieved with the verification MLP was not significantly higher than the Bayes verification error.

6 Experiments with speech data

6.1 Data description and feature extraction

For a speaker verification task with single speech frames we used speech signals from 48 male speakers recorded from different telephones. From short speech segments (32 ms frames) the 13 first Mel frequency cepstral coefficients (MFCCs)

were extracted and used as feature vectors for our experiments. For the phoneme verification task the first derivatives of the MFCCs were used in addition to the static MFCCs. The data from all speakers was divided into 3 disjoint sets (i.e. no speaker was present in more than one set). The MLP and KNN training vector pairs were extracted from the *training set* (26 speakers). The *validation set* (10 speakers) was used to stop the MLP training at the optimal point and to find the optimal value of k for the KNN classifier. The test vector pairs were taken from the *test set* (12 speakers). Note that all vector pairs were formed in a way that the two vectors of a pair were always from the same phoneme. In every set the number of pairs with vectors of the same speaker and the number of pairs with vectors from different speakers were equal.

6.2 Phoneme Verification

In this task the objective was to decide whether two speech feature vectors originate from the same phoneme. In this task the same classes (phonemes) are present in all 3 datasets since all signals have similar phonetic content. Yet all sets are extracted from different speakers as is described in Section 6.1.

Because of the large number of KNN training vectors which were necessary that the KNN algorithm converged, we used two types of input, namely pairs of concatenated vectors $\boldsymbol{p}_{in} = (\boldsymbol{x}, \boldsymbol{y})$ as mentioned above and coded vector pairs $\boldsymbol{p}_{in} = (|\boldsymbol{x} - \boldsymbol{y}|, \boldsymbol{x} + \boldsymbol{y})$ (see [5] for details about the input coding). This input coding sped up the training of the MLPs and led to a steeper descent of the KNN verification error in function of the number of training vectors.

The verification error of an MLP trained with 580000 vector pairs is shown in Figure 4. For comparison also the KNN error rate in function of the training set size is drawn. It can be seen that with this data the verification KNN converges only with a large number training vectors. This was expected because of the more complex nature of the problem. It can only be guessed where the asymptote and therefore the Bayes error will be. It seems that the verification error of the MLP is close to the Bayes verification error however. Since we did not have enough training data we could not prove this assumption. Furthermore it does not seem that the input coding had a big effect on the optimal verification result.

6.3 Speaker Verification

In this task the objective was to decide whether two speech frames are from speech signals of the same speaker or not. In this case all 3 sets of classes (speakers) were disjoint. Therefore a good generalization of the verification MLP is required.

The experiment results are shown in Figure 5. It can be seen that the KNN verification error in function of the training set size decreases much less steeply than in the experiments done with synthetic data and does not even get as low as the verification error of the MLP. This is possible since the training and test set have some mismatch because the speaker sets are disjoint (see Section 4). Here it can be seen very well that the KNN which is based on coded vector pairs

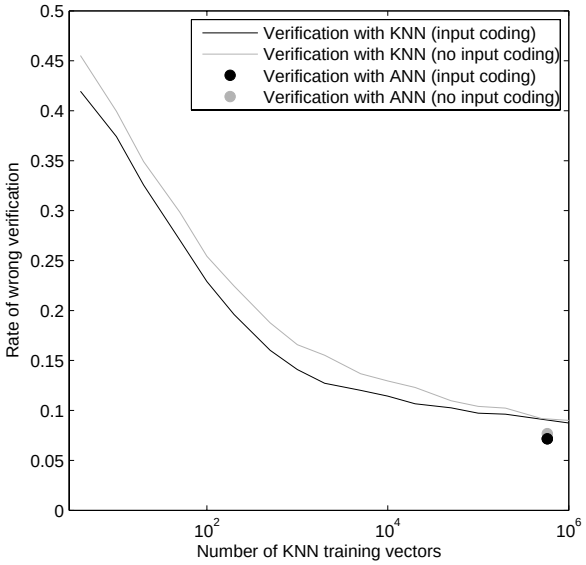

Fig. 4. Phoneme verification task: The KNN error rates decrease with increasing number of KNN training vectors. The error rates of the verification MLPs are shown as dots. The error rates for both, KNN and MLP are given for coded and uncoded input vectors.

converged with much less training vectors. In this case the verification MLP which used coded vector pairs was a bit worse however.

The verification error of the MLP is quite low if it is considered that the feature vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ were extracted from single speech frames only. If all phonetically matching frame pairs of two equally worded speech segments of about 1 s length are fed separately into the MLP and the output values of the MLP are averaged, the verification error rate is about 6 %. More detailed results can be found in [5], [6] and [2].

7 Conclusions

By means of experiments we have shown that the error rate of an appropriately configured and trained verification MLP is close to the Bayes error rate. Depending on the class distributions, the training can be fairly time-consuming, however. This is not critical in our application since the MLP is class independent and does not need to be retrained whenever new classes are added to the application.

For speech data with a virtually unlimited set of classes, as it is for example the case in speaker verification, MLP-based class verification has shown to be very efficient not only in terms of verification error but also with respect to computational complexity. For a speaker-verification task the good generalization property of the verification MLP could be shown. Thus the verification

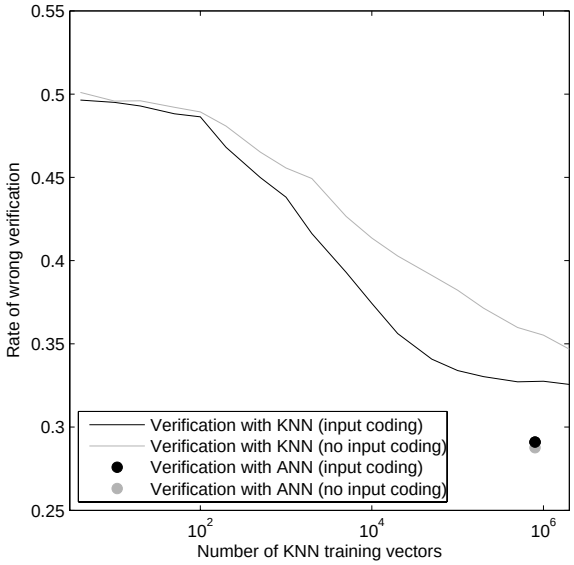

Fig. 5. Speaker verification: The KNN error rates decrease with increasing number of KNN training vectors. The error rates of the verification MLPs are shown as dots. The error rates for both, KNN and MLP are given for coded and uncoded input vectors.

MLPs are able to learn a general rule to distinguish between classes rather than class-specific features.

8 Acknowledgements

This work was partly funded by the Swiss National Center of Competence in Research IM2.

References

1. Hermansky, H., Ellis, D.P.W., Sharma, S.: Tandem connectionist feature extraction for conventional HMM systems. In: Proceedings of the ICASSP 2000. Volume 3. (2000) 1635–1638
2. Gerber, M., Beutler, R., Pfister, B.: Quasi text-independent speaker verification based on pattern matching. In: Proceedings of Interspeech, ISCA (2007) (to appear).
3. Haykin, S.: Neural Networks: A Comprehensive Foundation (2nd Edition). Prentice-Hall (1999)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley Interscience (2001)
5. Niesen, U., Pfister, B.: Speaker verification by means of ANNs. In: Proceedings of ESANN'04, Bruges (Belgium). (April 2004) 145–150
6. Gerber, M., Pfister, B.: Quasi text-independent speaker verification with neural networks. MLMI'05 Workshop, Edinburgh (United Kingdom) (July 2005)