IMPLICIT SOFTMAX TRANSFORMS FOR DIMENSIONALITY REDUCTION

Andreas Tuerk

Telecommunications Research Center Vienna (ftw.)
andreas.tuerk@ieee.org

ABSTRACT

This paper develops implicit softmax transforms (IST) which are dimensionality reducing transforms that are defined implicitly by minimisation of a weighted sum of Kullback-Leibler distances (WKL). The parameters of an IST are trained in combination with the parameters of the polynomial exponents of softmax functions. The resulting gradient of the WKL can be efficiently calculated and the computational effort scales well with the size of the training set. The paper compares IST's to PCA and LDA in classification experiments with two different types of classifiers on three different datasets, two of them from the UCI machine learning repository. It is shown that IST's outperform PCA and LDA in a majority of the cases. In one case reducing the dimension with an IST even gives an improvement over the high dimensional baseline system.

Index Terms— Pattern recognition, multidimensional signal processing, information theory

1. INTRODUCTION

The methods employed for dimensionality reduction fall into two major classes, namely feature selection [1] and the calculation of transformations into a low-dimensional space. Although the former represent a subset of the latter the approach to feature selection differs considerably from the calculation of dimension reducing transforms. In feature selection a combination of the most valuable features of a high dimensional data set is sought. On the contrary, dimensionality reducing transforms are not required to leave the original features unchanged and will, in general, produce transforms which combine the original features. Among the methods for calculating dimensionality reducing transforms the most well known are principal component analysis (PCA) [2], linear discriminant analysis (LDA) [2] and independent component analysis (ICA) [3]. Another method is the one developed in [4, 5, 6], where a transformation is trained by maximising the mutual information between the transformed data and the class labels.

The implicit softmax transform (IST) developed in this paper is similar to the latter, as it also optimises an information theoretic measure. Here, instead of maximising the mutual information, however, a sum of weighted Kullback-Leibler distances (WKL) between the true class probabilities and the estimated ones is minimised. In contrast to the method in [4, 5, 6], the computational effort for the calculation of the gradient of WKL scales well with the size of the training set and IST's can therefore be trained relatively easily on large datasets.

2. IMPLICIT SOFTMAX TRANSFORMS

An implicit softmax transform (IST) T maps a data point x in feature space X into a data point $T(x) = y$ in another feature space Y. In this paper T will be considered to be a linear transform and is therefore given by a matrix $T = (t_{ij})$. In order to optimise IST's they are concatenated with softmax functions and optimised together with the parameters of the softmax functions. For N classes and a data point x from feature space X, the concatenation of an IST T and softmax functions S_n is given by

$$S_n(T(x)) = \begin{cases} \frac{e^{q_n(T(x))}}{1+\sum_{i=1}^{N-1} e^{q_i(T(x))}} & : \quad 1 \leq n \leq N-1 \\ \frac{1}{1+\sum_{i=1}^{N-1} e^{q_i(T(x))}} & : \quad n = N \end{cases} \tag{1}$$

The $q_i(y)$ in (1) can, in principal, be any function of y, in this paper, however, for computational reasons, only polynomials in y will be considered. Therefore, the $q_i(y)$ can be written as follows.

$$q_i(y) = \sum_l c_l^i y^l \tag{2}$$

Here l can be a multi-index if the dimension of y is greater than one. The parameters to be trained are the coefficients of the polynomials in (2), i.e. the c_l^i, and the parameters of the IST, i.e. the t_{ij}. The cost function to be optimized is the sum over the weighted Kullback-Leibler distances (WKL) between the $S_n(T(x))$ and the true probabilities $p_n(x)$ for x to lie in class n, i.e.

$$\mathrm{WKL}(p, S \circ T) = \sum_x w_x \sum_{n=1}^{N} p_n(x) \log \frac{p_n(x)}{S_n(T(x))} \tag{3}$$

where $S \circ T$ denotes the totality of the concatenations of T and the $S_n(y)$ in (1). In (3) w_x is a positive weight that specifies

the importance of data point x. In this paper all points will be considered to be of equal importance and the w_x will therefore always be 1. In most classification tasks, a data point x will be uniquely associated to one of the classes and the $p_n(x)$ will therefore be 1 for the associated class and 0 otherwise.

The fact that T is implicitly defined together with the parameters of the softmax functions c_l^i through optimisation of WKL justifies the name implicit softmax transforms.

In the experiments in section 3 it was found to be of great importance that the rows of the IST are orthogonal. For this reason, a Gram Schmidt orthogonalisation was applied to the rows of an IST after training. This can be done, if the exponents $q_i(y)$ are polynomials, because for an arbitrary invertible transform W on Y, $q_i(W^{-1}(y))$ is again a polynomial on Y and $q_i(W^{-1}(W(T(x)))) = q_i(T(x))$. For this reason $W \circ T$ is associated to the same local minimum of WKL as T.

In a gradient based optimisation method, such as quasi-Newton, it is necessary to calculate the derivatives of WKL with respect to the polynomial parameters c_l^i and the transform parameters t_{ij}. After some calculation, these can be shown to be

$$\frac{\partial}{\partial c_l^i}\mathrm{WKL}(p, S\circ T) = -\sum_x w_x \left(p_i(x) - S_i(T(x))\right)(T(x))^l \tag{4}$$

and

$$\frac{\partial}{\partial t_{ij}}\mathrm{WKL}(p, S\circ T) \quad = \\ -\sum_x w_x \sum_{n=1}^{N-1} \left(p_n(x) - S_n(T(x))\right)\frac{\partial}{\partial y_i} q_n(T(x)) x_j \tag{5}$$

Equations (4) and (5) show that the computational cost of calculating the gradient of WKL scales linearly with the size of the training set and the number of classes. It also scales linearly with the dimension of X and Y and the degree of the polynomials $q_i(y)$, since the powers of the individual dimensions can be reused to calculate the multidimensional powers in (4) and (5). In comparison to the calculation of the gradient of the WKL cost function the method in [4, 5, 6] scales quadratically with the size of the training set. For this reason the experiments in [4] had to be restricted to subsets of the full training set of some of the databases of the UCI machine learning repository [7]. This was not necessary for the experiments in section 3 which made use of the full training sets.

Since the IST parameters t_{ij} are trained simultaneously with the parameters of the polynomials $q_i(y)$, the degree of these polynomials influences the resulting IST. This fact is illustrated in Figure 1 which also highlights some differences between LDA, PCA and IST. Figure 1 shows a simple two-dimensional classification task. The 1000 randomly generated data points in the two classes are represented by circles and stars, respectively. The solid line in Figure 1 which almost coincides with the x-axis is the line on which the data are projected by an LDA transform and an IST where the polynomial $q_1(y)$ has degree 3 or less. The dashed line which subtends and angle of roughly 45 degrees with the x- and the y-axis, and the dash-dotted line which is almost identical to the y-axis are the lines on which the data are projected by a PCA transform and an IST, respectively, where the IST was trained with a polynomial $q_1(y)$ of degree 4 or more. This is the minimal degree to obtain a transform onto the y-axis since the resulting decision boundary consists of 4 points on which the $q_1(y)$ has to be zero. In this example the transform onto the y-axis is clearly the best as it perfectly separates the two classes. This example therefore shows the importance of choosing a high enough degree for the polynomials $q_i(y)$ in order to obtain an IST with optimal separation between classes. It should be clear on the other hand that an unnecessarily high degree of the polynomials $q_i(y)$ will lead to over-fitting and a suboptimal IST. The optimal degree of the polynomials $q_i(y)$ has, so far, to be determined experimentally.

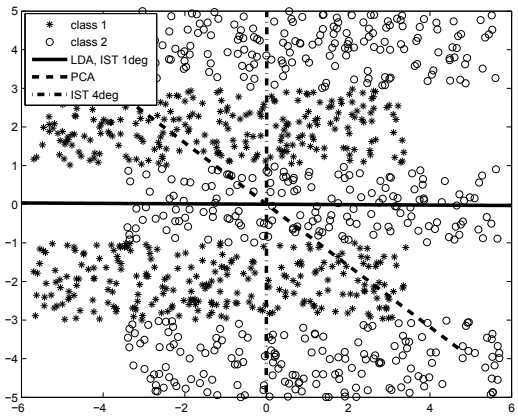

Fig. 1. Lines of projection of PCA, LDA and IST's for a simple two-class problem.

3. CLASSIFICATION EXPERIMENTS

Classification experiments were carried out on two corpora from the UCI machine learning repository [7], namely the Landsat and Letter databases. In addition, the Phoneme classification task was used that is part of the LVQ_PAK toolkit [8]. These classification tasks have rather different characteristics as can be seen in Table 1. LDA, PCA and IST transforms were calculated on the training sets of these tasks. The IST's were derived by numerical minimisation of the WKL cost function and subsequent orthogonalisation of the rows by Gram-Schmidt. Numerical minimisation of WKL was performed with a quasi-Newton method and the help of the deri-

	classes	dimension	train. set	test set
Letter	26	16	16000	4000
Landsat	6	36	4435	2000
Phoneme	20	20	1961	1961

Table 1. Classification task characteristics

vatives in (4) and (5). Five random initialisations were chosen for each dimension of IST and each degree of the polynomials $q_i(y)$. These were then trained a maximum of 1500 iterations. Subsequently, the IST that gave the lowest value of the WKL cost function was used in the classification experiments. The PCA, LDA and IST were applied to both the training and test sets. Subsequently two different types of classifiers were trained on the transformed training sets and evaluated on the transformed test sets. The two types of classifiers used are support vector machines and learning vector quantisation. These were trained with the publicly available toolkits SVMTorch [9] and LVQ_PAK [8]. As the resulting LVQ depends on the order in which the training data are processed, LVQ's were trained for 10 different random orderings of the training data. The average error rate for the 10 different LVQ's is presented here. Table 2 gives the classification error rates in percent for the baseline systems where the LVQ and SVM classifiers were trained and tested on the untransformed original datasets. On the Letter, Landsat and Phoneme database

	LVQ	SVM
Letter	6.07	2.18
Landsat	9.86	8.05
Phoneme	9.61	8.97

Table 2. Classification error rates (%) for untransformed data

the LVQ classifier was trained with 4000, 1000 and 500 code books, respectively. The SVM classifier used Gaussian kernels throughout the experiments. In the case of the Letter dataset the standard deviation of the Gaussians was chosen to be 5. For the Landsat and Phoneme datasets, however, the standard deviation of the Gaussians was varied continuously between 5 and 100 and the classifier with the best performance was chosen. This was done since the optimal standard deviation varied considerably with the dimension and type of transforms. This is probably due to the small size of the training sets of the Landsat and Phoneme datasets as compared to the training set of the Letter database. The same training and testing setup as in the baseline experiments was also used in the experiments with PCA, LDA and IST transformed data.

IST's were calculated for various degrees of softmax polynomials and the transform with the lowest error rate is quoted in the tables below. The best transforms were the same for support vector machine and learning vector quantisation classifiers. This means that the quality of the IST's in these experiments is independent of the classifier.

The results for the LVQ experiments on the Letter database are collected in Table 3, where the columns stand for the different dimensions of the transforms. The IST's in these experiments were trained for softmax polynomials with third degree. It can be seen that the classification results for the

Dim.	1	2	3	4	6	8
LDA	85.49	60.62	47.40	32.27	19.34	13.04
PCA	96.28	86.32	64.26	43.28	19.85	12.60
IST	84.37	56.19	36.77	24.29	11.67	5.60

Table 3. Letter database. LVQ classification error rate (%)

IST consistently outperform that for LDA and PCA. The difference in error rate between LDA and IST is relatively small for one dimension but becomes more pronounced for higher dimensions. For dimensions 6 and 8, for instance, IST gives a relative reduction in error rate by 40.7% and 55.6%, respectively, over the minimal error rate of LDA and PCA. Comparing Tables 2 and 3, the IST error rate for dimension 8, i.e. 5.6%, is even smaller than that of the baseline system, i.e. 6.07%, which is trained on the full 16 dimensional feature vector. It should be noted that this was also observed for dimensions 10 and 12 and polynomial degree 2, where the error rate was even lower at 3.53% and 3.22%. Table 4 shows

Dim.	1	2	3	4	6	8
LDA	99.22	80.7	53.12	31.62	17.2	11.78
PCA	96.03	93.52	67.5	38.67	15.65	9.48
IST	92.97	67.08	41.38	25.25	10.30	4.70

Table 4. Letter database. SVM classification error rate (%)

the error rates for the same LDA, PCA and IST as in Table 3 this time for the SVM classifier. As previously, the difference between error rate for IST and PCA or LDA is relatively small for dimension one but becomes increasingly more pronounced for higher dimensions. For dimensions 6 and 8 IST gives a relative improvement of 34.2% and 50.4%, respectively, over the minimal error rate of LDA and PCA.

Tables 5 and 6 show the error rates of the LVQ and SVM classifiers, respectively, on the Landsat database. Since there are only 6 classes in the Landsat database, the maximal dimension of the LDA transform is 5. Consequently, there are no error rates available for the LDA transform in dimensions 9 and 15. The IST was trained with polynomials of degree 3 for dimension 1 and 2, degree 4 for dimension 3 and 4 and degree 1 for dimension 9 and 15.

Whereas the difference in error rates became more pronounced for higher dimensions on the letter database, on the Landsat database this trend appears to be reversed. Here the

difference in error rate is highest for dimension one and becomes smaller for higher dimensions. For dimensions 9 and 15, in fact, the error rate on the IST transformed data is slightly higher than on the PCA transformed data. This might be a consequence of the size of the training set which is not large enough to train the polynomial parameters c_l^i reliably in such high dimensions.

Dim.	1	2	3	4	9	15
LDA	49.03	24.4	14.91	13.65	$*$	$*$
PCA	53.64	17.54	14.64	13.24	10.45	10.11
IST	28.16	16.90	14.43	12.91	11.63	10.62

Table 5. Landsat database. LVQ classification error rate (%)

Dim.	1	2	3	4	9	15
LDA	48.25	20.45	13.4	11.95	$*$	$*$
PCA	47.55	15.95	13.1	10.9	9.25	8.35
IST	26.1	15.55	12.45	11.3	10.83	10.02

Table 6. Landsat database. SVM classification error rate (%)

Tables 7 and 8 show the error rates for the LVQ and SVM classifiers on the Phoneme database that comes as part of the LVQ_PAK distribution. The IST's were trained with polynomials of degree 2 for dimensions 1 to 3 and degree 1 for dimensions 4 to 9. In Table 7 the trend is similar to that in the experiments on the Landsat database. IST clearly outperforms both LDA and PCA in dimension one but the difference becomes less pronounced for higher dimensions. For dimension 9 LVQ classification on the IST transformed data gives slightly worse performance than on LDA and PCA transformed data whereas classification with the SVM classifier yields identical performance for PCA and IST transformed data with a slight degradation for LDA transformed data.

Dim.	1	2	3	4	6	9
LDA	59.44	32.09	25.67	18.6	16.13	12.82
PCA	57.15	29.34	21.87	17.81	15.51	12.02
IST	47.95	30.97	21.22	17.32	14.96	13.34

Table 7. Phoneme database. LVQ classification error rate (%)

4. CONCLUSIONS

This paper introduced implicit softmax transforms which are trained by minimisation of a weighted sum of Kullback-Leibler distances. The calculation of the gradient of this cost function scales well with the size of the training set and can therefore be effectively employed in a quasi-Newton minimisation method. The classification experiments showed that IST consistently outperforms PCA and LDA on a large training set. In the case of the LVQ classifier, reduction from 16 to 8, 10 and 12 dimensions even gave an improvement over the baseline system. For databases with little training data IST outperforms PCA and LDA on small dimensions while for larger dimensions there is insufficient data to train IST robustly.

Dim.	1	2	3	4	6	9
LDA	56.63	31.14	21.56	16.82	14.73	11.88
PCA	53.72	25.89	19.32	15.6	13	11.16
IST	47.4	25.13	19.11	14.49	12.75	11.66

Table 8. Phoneme database. SVM classification error rate(%)

5. ACKNOWLEDGEMENTS

I would like to thank K. Torkkola for valuable information about his testing environment. Thanks are also due to my colleagues T. Zemen, T. Nordström and M. Pucher for valuable discussions. This work has been partly funded by an FWF grant on speech recognition.

6. REFERENCES

[1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research 3*, pp. 1157–1182, 2003.

[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, John Wiley & Sons, 2nd edition, 2001.

[3] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, Wiley Interscience, 2001.

[4] Kari Torkkola and William M. Campbell, "Mutual information in learning feature transformations," in *Proc. 17th International Conf. on Machine Learning*. 2000, pp. 1015–1022, Morgan Kaufmann, San Francisco, CA.

[5] K. Torkkola, "Learning discriminative feature transforms to low dimensions in low dimentions," in *Advances in Neural Information Processing Systems*. 2001, vol. 14, MIT Press.

[6] K. Torkkola, "Feature extraction by non parametric mutual information maximization," *The Journal of Machine Learning Research*, vol. 3, pp. 1415–1438, 2003.

[7] A. Asuncion and D. J. Newman, "UCI machine learning repository," 2007.

[8] T. Kohonen, J. Kangas, J. Laaksonen, and K. Torkkola, "LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms," in *Proceedings IJCNN*, 1992, vol. 1, pp. 725–730.

[9] R. Collobert and S. Bengio, "SVMTorch: Support vector machines for large-scale regression problems," *Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.