Fixed-Size Kernel Logistic Regression for Phoneme Classification

Peter Karsmakers [1,2]*, Kristiaan Pelckmans* [2]*, Johan Suykens*[2]*, Hugo Van hamme*[3]

[1]IIBT, K.H. Kempen (Associatie KULeuven),B-2440 Geel, Belgium
[2]ESAT-SCD/SISTA, K.U.Leuven, B-3001 Heverlee, Belgium
[3]ESAT-PSI/SPEECH, K.U.Leuven, B-3001 Heverlee, Belgium

[peter.karsmakers,kristiaan.pelckmans,johan.suykens,hugo.vanhamme]@esat.kuleuven.be

Abstract

Kernel logistic regression (KLR) is a popular non-linear classification technique. Unlike an empirical risk minimization approach such as employed by Support Vector Machines (SVMs), KLR yields probabilistic outcomes based on a maximum likelihood argument which are particularly important in speech recognition. Different from other KLR implementations we use a Nyström approximation to solve large scale problems with estimation in the primal space such as done in fixed-size Least Squares Support Vector Machines (LS-SVMs). In the speech experiments it is investigated how a natural KLR extension to multi-class classification compares to binary KLR models coupled via a one-versus-one coding scheme. Moreover, a comparison to SVMs is made.

Index Terms: phoneme classification, kernel logistic regression, large-scale, multi-class

1. Introduction

To tackle the task of phoneme classification we choose a Logistic Regression (LR) and Kernel Logistic Regression (KLR) approach. Hidden Markov models (HMMs) [16] are the state-of-the-art technique for current automatic speech recognition (ASR) systems. It is widely recognized that estimating the HMM parameters via a maximum likelihood criterion does not directly optimize the classification performance of the models. It is therefore of interest to develop alternative methods which infer the parameters by discriminative measures of performance. Several techniques were presented for the task of phoneme recognition such as Linear Discriminant Analysis (LDA) (e.g., [1]), Multi-Layer Perceptrons (MLPs) (e.g., [17]), Hidden Conditional Random Fields (HCRFs) (e.g., [13]), Support Vector Machines (SVMs) (e.g., [18]), KLR (e.g., [15]). Although SVMs have shown promising results for phoneme recognition, the choice for a LR or KLR approach over an empirical risk minimization approach such as SVM, is that the former yields probabilistic outcomes based on a maximum likelihood argument instead of a binary decision. KLR has an additional advantage that the extension to the multi-class case is well described, which must be contrasted to the commonly used coding approach (see e.g., [6],[5]). Obtaining phoneme probabilities offers ample perspective for integration of this work in an ASR system.

Unlike SVMs, KLR by its nature is not sparse and needs all training samples in its final model. Different adaptations to the original algorithm were performed to obtain sparseness such as in [6]. In this paper we employ a different practical technique, suited for large data sets, based on fixed-size Least Squares Support Vector Machines (LS-SVMs) [5], which we can use because KLR is related to a weighted version of LS-SVMs [12].

Our experiments are performed on the TIMIT data set, where we compare two different multi-class KLR implementations against binary SVM classifiers combined via a one-versus-one coding scheme.

This paper is organized as follows. In Section 2 we give an introduction to logistic regression. Section 3 describes the extension to kernel logistic regression. A fixed-size implementation is given in Section 4. Section 5 describes extension to multi-class KLR and Section 6 reports numerical results on the TIMIT speech data set. Finally we conclude in Section 7.

2. Logistic regression

After introducing some notations, we recall the principles of logistic regression. Suppose we have a binary classification problem with a training set $\{(x_i, y_i)\}_{i=1}^N \subset \mathbb{R}^d \times \{-1, 1\}$ with N samples, where input samples x_i are i.i.d. from an unknown probability distribution over the random vectors $(\mathbf{X}, \mathbf{Y})$. We define the first element of x_i to be 1, so that we can incorporate the intercept term in the parameter vector w. The goal is to find a classification rule from the training data, such that when given a new input x_*, we can assign a class label to it. In logistic regression the conditional class probabilities are estimated via logit stochastic models

$$\begin{cases} Pr(\mathbf{Y} = -1 \mid \mathbf{X} = x; w) = \frac{\exp(w^T x)}{1+\exp(w^T x)} \\ Pr(\mathbf{Y} = 1 \mid \mathbf{X} = x; w) = \frac{1}{1+\exp(w^T x)}, \end{cases} \tag{1}$$

The class membership of a new point x_* can be given by the classification rule which is

$$\underset{c \in \{-1,1\}}{\arg\max}\ Pr(\mathbf{Y} = c | \mathbf{X} = x_*; w). \tag{2}$$

The common method to infer the parameters of the different models is via the use of penalized negative log likelihood (PNLL)

$$\begin{aligned} &\min_w \ell(w) = \\ &-\ln\left(\textstyle\prod_{i=1}^N Pr(\mathbf{Y} = y_i | \mathbf{X} = x_i; w)\right) + \frac{\nu}{2} w^T w, \end{aligned} \tag{3}$$

where the regularization parameter ν must be set such that the parameters in w stay small in order to obtain a good bias-variance trade-off and avoid overfitting.

We derive the objective function for LR by combining (1)

with (3) which gives

$$\ell_{LR}(w) = \sum_{i\in\mathcal{D}_1} \ln \frac{\exp(w^T x_i)}{1+\exp(w^T x_i)} + \sum_{i\in\mathcal{D}_2} \ln \frac{1}{1+\exp(w^T x_i)} + \frac{\nu}{2} w^T w, \tag{4}$$

where $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$, $\mathcal{D}_1 \cap \mathcal{D}_2 = \varnothing$ and $y_i = c, \forall x_i \in \mathcal{D}_c$. In the sequel we use the shorthand notation

$$p_{c,i} = Pr(\mathbf{Y} = c|\mathbf{X} = x_i; w). \tag{5}$$

This PNLL criterion for LR is known to possess a number of useful properties such as the fact that it is convex in the parameters w, smooth and has asymptotic optimality properties.

Until now we have defined a model and an objective function which has to be optimized to fit the parameters on the observed data. Most often this optimization is performed by a Newton based strategy where the solution can be found by iterating

$$w^{(k)} = w^{(k-1)} + s^{(k)}, \tag{6}$$

over k until convergence. The minimization in this case is equivalent to an iteratively regularized re-weighted least squares problem (IRRLS) (e.g. [6]) which can be written as

$$\min_{s^{(k)}} \quad \tfrac{1}{2}||Xs^{(k)} - z^{(k)}||^2_{W^{(k)}} + \tfrac{\nu}{2}(s^{(k)} + w^{(k-1)})^T(s^{(k)} + w^{(k-1)}), \tag{7}$$

where

$$z^{(k)} = (W^{(k)})^{-1} q. \tag{8}$$

where we define $X = [x_1; ...; x_N]$,$g_i = p_{1,i}(1 - p_{1,i})$, $W = \text{diag}([g_1; ...; g_N])$, $q_i = (p_{y_i,i} - 1)y_i$ and $q = [q_1; ...; q_N]$.

3. Kernel logistic regression

In this section we define the minimization problem for the kernel version of logistic regression. This result is based on an optimization argument as opposed to the use of an appropriate *Representer Theorem* [7]. The LR model as defined in (1) can be advanced with a nonlinear extension to kernel machines where the inputs x are mapped to a high dimensional space. Define $\Phi \in \mathbb{R}^{N\times d_\varphi}$ as X where x_i is replaced by $\varphi(x_i)$ and where $\varphi : \mathbb{R}^d \to \mathbb{R}^{d_\varphi}$ denotes the feature map induced by a positive definite kernel. With the application of the Mercer's theorem for the kernel matrix Ω as $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j), i, j = 1, \ldots, N$, it is not required to compute explicitly the nonlinear mapping $\varphi(\cdot)$ as this is done implicitly through the use of positive kernel functions K. For K there are usually the following choices: $K(x_i, x_j) = x_i^T x_j$ (linear kernel); $K(x_i, x_j) = (x_i^T x_j + h)^b$ (polynomial of degree b, with $h \geq 0$ a tuning parameter); $K(x_i, x_j) = \exp(-||x_i - x_j||_2^2/\sigma^2)$ (radial basis function, RBF), where σ is a tuning parameter. In KLR the models are defined as

$$\begin{cases} Pr(\mathbf{Y} = -1 \mid \mathbf{X} = x; w) = \frac{\exp(w^T\varphi(x))}{1+\exp(w^T\varphi(x))} \\ Pr(\mathbf{Y} = 1 \mid \mathbf{X} = x; w) = \frac{1}{1+\exp(w^T\varphi(x))}, \end{cases} \tag{9}$$

Starting from (8) we include a feature map and introduce the error variable e, this results in

$$\min_{s^{(k)},e^{(k)}} \frac{1}{2}e^{(k)^T}W^{(k)}e^{(k)} + \frac{\nu}{2}(s^{(k)} + w^{(k-1)})^T(s^{(k)} + w^{(k-1)}) \quad \text{such that } z^{(k)} = \Phi s^{(k)} + e^{(k)}, \tag{10}$$

which in the context of LS-SVMs is called the primal problem. In its dual formulation the solution to this optimization problem can be found by iteratively solving a linear system.

$$\left(\frac{1}{\nu}\Omega + W^{(k)^{-1}}\right)\alpha^{(k)} = z^{(k)} + \Omega\alpha^{(k-1)}, \tag{11}$$

where $z^{(k)}$ is defined as in (8). The probabilities of a new point x_* can be predicted using (9) where $w^T\varphi(x_*) = \frac{1}{\nu}\sum_{i=1}^N \alpha_i K(x_i, x_*)$. The proof can be found in [12].

4. Kernel logistic regression: a fixed-size implementation

4.1. Nyström approximation

In the previous paragraph we stated a primal and a dual formulation of the optimization problem. Suppose one takes a finite dimensional feature map (e.g. a linear kernel), then one can equally well solve the primal as the dual problem. In fact, solving the primal problem is more advantageous for larger data sets because the dimension of the unknowns $w \in \mathbb{R}^d$ compared to $\alpha \in \mathbb{R}^N$. In order to work in the primal space using a kernel function other than the linear one, it is required to compute an explicit approximation of the nonlinear mapping φ. This leads to a sparse representation of the model when estimating in primal space.

Explicit expressions for φ can be obtained by means of an eigenvalue decomposition of the kernel matrix Ω with entries $K(x_i, x_j)$. Given the integral equation $\int K(x, x_j)\phi_i(x)p(x)dx = \lambda_i\phi_i(x_j)$, with solutions λ_i and ϕ_i for a variable x with probability density $p(x)$, we can write

$$\varphi = [\sqrt{\lambda}_1\phi_1, \sqrt{\lambda}_2\phi_2, \ldots, \sqrt{\lambda}_{d_\varphi}\phi_{d_\varphi}]. \tag{12}$$

Given the data set, it is possible to approximate the integral by a sample average. This will lead to the eigenvalue problem (Nyström approximation [8])

$$\frac{1}{N}\sum_{l=1}^N K(x_l, x_j)u_i(x_l) = \lambda_i^{(s)}u_i(x_j), \tag{13}$$

where the eigenvalues λ_i and eigenfunctions ϕ_i from the continuous problem can be approximated by the sample eigenvalues $\lambda_i^{(s)}$ and the eigenvectors $u_i \in \mathbb{R}^N$ as

$$\hat{\lambda}_i = \frac{1}{N}\lambda_i^{(s)}, \hat{\phi}_i = \sqrt{N}u_i. \tag{14}$$

Based on this approximation, it is possible to compute the eigendecomposition of the kernel matrix Ω and use its eigenvalues and eigenvectors to compute the i-th required component of $\hat{\varphi}(x)$ simply by applying (12) if x is a training point, or for any new point x_* by means of

$$\hat{\varphi}(x_*) = \frac{1}{\sqrt{\lambda}_i^{(s)}}\sum_{j=1}^N u_{ji}K(x_j, x_*). \tag{15}$$

4.2. Sparseness and large scale problems

Until now the entire training sample of size N to compute the approximation of φ will yield at most N components, each one of which can be computed by (14) for all x, where x is a row of X. However, if we have a large scale problem, it has been motivated [5] to use a subsample of $M \ll N$ data points to compute the $\hat{\varphi}$. In this case, up to M components, which are called support vectors, will be computed. External criteria such as entropy maximization can be applied for an optimal selection of the subsample: given a fixed-size M, the aim is to select the support vectors that maximize the quadratic Renyi entropy [9]

$$H_R = -\ln \int p(x)^2 dx, \tag{16}$$

which can be approximated by using $\int \hat{p}(x)^2 dx = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \Omega_{ij}$. The use of this active selection procedure can be important for large scale problems, as it is related to the underlying density distribution of the sample. In this sense, the optimality of this selection is related to the final accuracy of the model. This finite dimensional approximation $\hat{\varphi}(x)$ can be used in the primal problem (10) to estimate w with a sparse representation [5].

5. Multi-class kernel logistic regression

Kernel logistic regression can be naturally extended to a multi-class version. Suppose we have a multi-class problem with C classes ($C \geq 2$) with a training set $\{(x_i, y_i)\}_{i=1}^{N} \subset \mathbb{R}^d \times \{1, 2, ..., C\}$ with N samples, where input samples x_i are i.i.d. from an unknown probability distribution over the random vectors $(\mathbf{X}, \mathbf{Y})$. In multi-class KLR the conditional class probabilities can be written by

$$\begin{cases} Pr(\mathbf{Y} = 1 \mid \mathbf{X} = x; w_1, ..., w_m) = \frac{\exp(w_1^T \varphi(x))}{1 + \sum_{c=1}^{m} \exp(w_c^T \varphi(x))} \\ Pr(\mathbf{Y} = 2 \mid \mathbf{X} = x; w_1, ..., w_m) = \frac{\exp(w_2^T \varphi(x))}{1 + \sum_{c=1}^{m} \exp(w_c^T \varphi(x))} \\ \vdots \\ Pr(\mathbf{Y} = C \mid \mathbf{X} = x; w_1, ..., w_m) = \frac{1}{1 + \sum_{c=1}^{m} \exp(w_c^T \varphi(x))} \end{cases}. \tag{17}$$

where $m = C - 1$. Deriving this multi-class implementation results in one large learning problem without the use of a coding scheme. Other possible multi-class implementations can be built by combining several independent binary classifiers via a common coding scheme approach, e.g. one-versus-one and one-versus-all. When using one-versus-all one has to optimize C different smaller learning problems compared to one-versus-one where $C(C-1)/2$ small models have to be optimized. In [18] one-versus-one coding schemes resulted in better classification accuracies than one-versus-all we therefore chose to compare the natural extension of multi-class KLR to one-versus-one KLR. To obtain probabilities when using a one-versus-one coding scheme we use method 3 described in [3]. The resulting pairwise probabilities $\mu_{ij} = \Pr(\mathbf{Y} = i | \mathbf{Y} = i \text{ or } \mathbf{Y} = j, \mathbf{X} = x)$ are transformed to the a posteriori probability by

$$\Pr(\mathbf{Y} = i | \mathbf{X} = x) = 1 / \left(\sum_{j=1, j \neq i}^{C} \frac{1}{\mu_{ij}} - (C - 2) \right). \tag{18}$$

The probability outcomes in (18) are normalized so that they sum to one for each evaluation.

6. Experiments

To test the performance of KLR we used the TIMIT database [14]. Training was performed on the 'sx' and 'si' training sentences. These create a training set with $3,696$ utterances from 168 different speakers. For testing we chose the full test set. It consists of $1,344$ utterances from 168 different speakers not included in the training set. All utterances contain labels indicating the phoneme identity and the starting and ending time of each phoneme. The standard Kai-Fu Lee clustering [16] was used, resulting in a set of 39 phonemes.

A key problem with conducting classification experiments with the TIMIT database is that the segments that we are seeking to classify are not of a uniform length. In order to use the machine learning techniques such as K-Nearest Neighbors (KNN)[4], Linear Discriminant Analysis (LDA) [1], SVM and KLR we must encode the waveform information in a fixed-length vector. We chose the same simple method of encoding the variable length segment information in a vector of fixed length as in [18]. We converted the utterances from their waveform representation into a sequence of 36 dimensional observation vectors. These observation vectors were obtained by means of mutual information based discriminant linear transformation [19] on 24 MEL spectra and their first and second order time derivatives. Each phoneme segment was broken into three regions in the ratio 3-4-3. The 36 dimensional vectors belonging to each of these regions were averaged resulting in three 36 dimensional vectors. In addition, the 36 dimensional vectors belonging to a window region centered at the start of the phonetic segments and with a 50 ms width were averaged, resulting in another 36 dimensional vector. The same was done for a window centered at the end of the segment. One additional feature indicating the log-duration of the phoneme segment was added. This resulted in a $142,910$ train vectors and $51,681$ test vectors with 181 dimensions.

We used a small part of the train set for tuning purposes. After tuning we used the full train set to train the algorithm. The phonetic classification accuracies on the full test set using different classifiers are shown in Table 1.

The SVM experiments are performed with the LIBSVM toolbox [10]. The binary SVM probability outputs are obtained after mapping the SVM distance outputs to a sigmoid function, described in [11]. Via a pairwise one-versus-one coding scheme the binary outcomes are combined using the second approach as described in [3]. Our result SVM results using an RBF-kernel is much higher than 76.3% as reported in [18]. In comparison to SVM classification with a voting strategy, where each binary classification is considered to be a voting and a data point is designated to be in a class with maximum votes, we noticed that because of the SVM probability estimate the accuracy increases from 82.2% to 82.9%. For the fixed-size multi-class KLR experiments without coding scheme we used an alternated descent version of Newton's method [12] to make it possible to set $M = 1,000$. In our theoretical explanation (17) each class model has the same feature map φ, this results in m models with $1,000$ support vectors. Although, certainly not optimal we have tried to incorporate more information in the model without increasing the computational complexity by using a different φ for each logit model by choosing half of the support vectors randomly from the model class and the rest we choose randomly out of all other classes. The one-versus-one multi-class

Table 1: Classification accuracies on the TIMIT full test set for different algorithms. The column **acc** gives the percentage of correctly classified phoneme segments. The percentages in the **10-best** column are equal to the proportion of estimated phonemes for which the correct class was one of the 10 most probable classes. The acronym oVso stands for one-versus-one coding scheme. RBF indicates that an RBF kernel was used and LIN stands for linear kernel.

alg	acc(%)	10-best(%)
SVM (RBF)	82.9	99.8
KLR oVso (RBF)	78.1	99.6
KLR (RBF)	76.3	99.6
SVM (LIN)	76.1	99.6
LR oVso	74.9	99.5
LR	72.8	99.2
KNN (k=1)	67.8	x
LDA	66.5	98.1

KLR approach is used as described in Section 5. Using this coding scheme results in 741 small models, where each model has $1,000$ support vectors. A possible explanation is that this approach gives better results than using the natural multi-class extension, were we have only 38 models with $1,000$ support vectors. As a consequence the model evaluation for new observations takes less long to compute than in the case of using a one-versus-one coding scheme.

Although the KLR results are not as good as those obtained with an accurately tuned SVM implementation, they are comparable with results we obtained when conducting an HMM experiment where we used context independent acoustical models with 2 to 4 states per phoneme with in total $5,550$ Gaussians (average 124/state). The same 36 dimensional observation vectors were used as in the other experiments. Using an unigram model we obtained 78.4% on the full test set.

7. Conclusions

In this paper we applied a fixed-size algorithm for multi-class KLR models on the TIMIT speech data set. We showed that the performance in terms of correct classifications on this data is comparable to that of HMMs in combination with Gaussian mixture models. In the future we will investigate whether or not corrections for class imbalance, another support vector selection technique instead of random selection and a different coupling of coding scheme probabilities will improve the multi-class KLR classification accuracies.

8. Acknowledgments

Research supported by GOA AMBioRICS, CoE EF/05/006; (Flemish Government): (FWO): PhD/postdoc grants, projects, G.0407.02, G.0197.02, G.0141.03, G.0491.03, G.0120.03, G.0452.04, G.0499.04, G.0211.05, G.0226.06, G.0321.06, G.0553.06, G.0302.07. (ICCoS, ANMMM, MLDM); (IWT): PhD Grants,GBOU (McKnow), Eureka-Flite2 - Belgian Federal Science Policy Office: IUAP P5/22,PODO-II,- EU: FP5-Quprodis; ERNSI; - Contract Research/agreements: ISMC/IPCOS, Data4s, TML, Elia, LMS, Mastercard. JS is a professor and BDM is a full professor at K.U.Leuven Belgium. This publication only reflects the authors' views.

9. References

[1] T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning*, Springer, 2001.

[2] S.S. Keerthi, K. Duan, S.K. Shevade and A.N. Poo "A Fast Dual Algorithm for Kernel Logisic Regression ", *Machine Learning*, vol. 61, p. 151-165, 2005.

[3] T.-F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling., *Journal of Machine Learning Research*, vol. 5., p. 975-1005, 2004.

[4] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[5] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002.

[6] J. Zhu, T. Hastie, "Kernel logistic regression and the import vector machine", *Advances in Neural Information Processing Systems*, vol. 14, 2001.

[7] G. Kimeldorf, G. Wahba, "Some results on Tchebycheffian spline functions", *Journal of Mathematics Analysis and Applications*,vol. 33, pp. 82-95, 1971.

[8] C.K.I. Williams, M. Seeger "Using the Nyström Method to Speed Up Kernel Machines", *Proceedings Neural Information Processing Systems*, vol 13., MIT press, 2000.

[9] M. Girolami "Orthogonal Series Density Estimation and the Kernel Eigenvalue Problem", *Neural Computation*, vol. 14(3), 669-688, 2003.

[10] C.-C. Chang, C.-J. Lin, "LIBSVM : a library for support vector machines", *Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2001.

[11] H.-T. Lin, C.-J. Lin, R.-C. Weng, A note on Platt's probabilistic outputs for support vector machines, *Technical report*, 2003.

[12] P. Karsmakers, K. Pelckmans, J. A. K. Suykens, "Multi-class kernel logistic regression: a fixed-size implementation", *Internal Report 07-39*, 2007. Accepted for publication in Proc. of IJCNN 2007.

[13] A. Gunawardana, M. Mahajan, A. Acero and J.C. Plat, "Hidden conditional random fields for phone classification", *Proceedings of Eurospeech 2005*, Lisbon, 2005.

[14] TIMIT Acoustic -Phonetic Continuous Speech Corpus, *National Institute of Standards and Technology Speech Disc 1 -1.1*, NTIS Order No. PB91 -5050651996, 1990.

[15] M. Katz, M. Schaffner, E. Andelic, S. Krüger, A. Wendemuth "Sparse Kernel Logistic Regression for Phoneme Classification", *Proceedings of 10th International Conference on Speech and Computer (SPECOM)*, vol. 2, pp. 523-526, 2005.

[16] K.F. Lee and H.W. Hon, "Speaker-independent phone recognition using hidden Markov models", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641-1648, 1988.

[17] Y. Bengio, "Neural networks for speech and sequence recognition", *London International Thomson Computer Press* , 1995.

[18] P. Clarkson, P.J. Moreno, "On the use of support vector machines for phonetic classification", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2 , pp. 585-588, 1999.

[19] K. Demuynck, *Extracting, Modelling and Combining Information in Speech Recognition*, Ph.D. thesis, 2001.