SEQUENCE ESTIMATION AND CHANNEL EQUALIZATION USING FORWARD DECODING KERNEL MACHINES

Shantanu Chakrabartty and Gert Cauwenberghs

Center for Language and Speech Processing,
Dept. of Electrical and Computer Engineering
Johns Hopkins University, Baltimore, MD 21218
{*shantanu,gert*}*@jhu.edu*

ABSTRACT

A forward decoding approach to kernel machine learning is presented. The method combines concepts from Markovian dynamics, large margin classifiers and reproducing kernels for robust sequence detection by learning inter-data dependencies. A MAP (maximum a posteriori) sequence estimator is obtained by regressing transition probabilities between symbols as a function of received data. The training procedure involves maximizing a lower bound of a regularized cross-entropy on the posterior probabilities, which simplifies into direct estimation of transition probabilities using kernel logistic regression. Applied to channel equalization, forward decoding kernel machines outperform support vector machines and other techniques by about 5dB in SNR for given BER, within 1dB of theoretical limits.

1. INTRODUCTION

Many digital communication receivers require equalizers to combat channel inter symbol interference (ISI) and co-channel interference to obtain reliable data transmission. Traditionally decision-feedback equalizers (DFE) implemented by FIR filters and maximum likelihood estimation (MLE) have been employed for this purpose [7, 5]. The inherent complexity of a true MLE decoding procedure renders it impractical in many implementations, and MLE performance is known to degrade under time-varying channel conditions. Symbol decision equalizers have a relatively simple architecture and training procedure but do not perform as well as nonlinear equalizers based on neural networks, such as multilayer perceptrons or radial basis functions [6].

Large margin classifiers, like support vector machines, have been the subject of intensive research in the neural network and artificial intelligence communities [2, 13]. They are attractive because they generalize well even with relatively few data points in the training set. Bounds on the generalization error can be directly estimated from the training data. Recently, support vector machines have been used for nonlinear equalization and have shown to provide very encouraging results compared to other nonlinear equalizers [11]. Figure 1 shows a system block diagram for channel equalization employing an SVM equalizer. The use of a standard SVM classifier inherently assumes that the data points are identically independently distributed. This is an unlikely scenario in ISI channels where there exists a sequential structure amongst the received symbols.

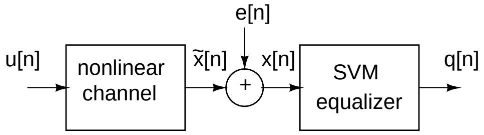

Fig. 1. *System architecture incorporating an SVM-based MAP equalizer to compensate for channel distortion.*

This paper describes a new architecture, which we term forward decoding kernel machine (FDKM), that augments the ability of large margin classifiers to perform sequence decoding and to infer the sequential properties of the data. FDKM performs a large margin discrimination based on the trajectory of the data rather than solely on individual data points and hence relaxes the constraint of *i.i.d.* data.

We incorporate Markovian dynamics into the framework of large margin classifiers using kernels, and provide a sequential algorithm to train the kernel machine. The time-complexity of the algorithm can be suitably improved by tuning the parameters of the learning model and also by pruning.

The paper is organized as follows. Section 2 introduces and formulates FDKM along with its training procedure. Section 3 applies FDKM to the problem to channel equalization. Section 4 presents experimental results. Finally Section 5 provides concluding remarks.

2. PROBLEM FORMULATION

The problem of sequence decoding and channel equalization is formulated in the framework of MAP (maximum a posteriori) estimation, combining Markovian dynamics with kernel machines. Consider a Markovian model with symbols belonging to S classes, as illustrated in Figure 2 for $S = 2$. Transitions between the classes are modulated in probability by observation (data) vectors $\mathbf{x}$ over time.

2.1. Decoding Formulation

The MAP forward decoder receives the sequence $\overline{\mathbf{X}}[n] = \{\mathbf{x}[n], \mathbf{x}[n-1], \ldots, \mathbf{x}[1]\}$ and produces an estimate of the probability of the state variable $q[n]$ over all classes i, $\alpha_i[n] = P(q[n] = i \mid \overline{\mathbf{X}}[n], \mathbf{w})$, where $\mathbf{w}$ denotes the set of parameters for the learning machine. Unlike *hidden* Markov models, the states directly

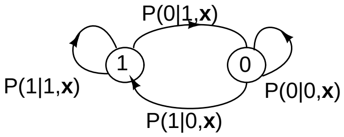

Fig. 2. *Two state markov model, where transition probabilities between states are modulated by the observation vector* **x***.*

encode the symbols, and the observations $\mathbf{x}$ modulate transition probabilities between states [3]. Estimates of the posterior probability $\alpha_i[n]$ for soft decoding are obtained from estimates of local transition probabilities using the *forward-decoding* procedure [1, 3]

$$\alpha_i[n] = \sum_{j=0}^{S-1} P_{ij}[n]\, \alpha_j[n-1] \tag{1}$$

where $P_{ij}[n] = P(q[n] = i \mid q[n-1] = j, \mathbf{x}[n], \mathbf{w})$ denotes the probability of making a transition from class j at time $n-1$ to class i at time n, given the current observation vector $\mathbf{x}[n]$. The forward decoding (1) embeds sequential dependence of the data wherein the probability estimate at time instant n depends on all the previous data. An on-line estimate of the symbol $q[n]$ is thus obtained:

$$q^{\text{est}}[n] = \arg\max_i \alpha_i[n] \tag{2}$$

The BCJR forward-backward algorithm [1] produces in principle a better estimate that accounts for future context, but requires a backward pass through the data, which is impractical in many applications.

Accurate estimation of transition probabilities $P_{ij}[n]$ in (1) is crucial in decoding (2) to provide good performance. We use kernel logistic regression [9], with regularized maximum cross-entropy, to model conditional probabilities.

2.2. Training Formulation

For training the MAP forward decoder, we assume access to a training sequence with labels (class memberships). Continuous (soft) labels can be assigned rather than binary indicator labels, to signify uncertainty in the training data over the classes. Like probabilities, label assignments are normalized: $\sum_{i=0}^{S-1} y_i[n] = 1, y_i[n] \geq 0$.

The objective of training is to maximize the cross-entropy of the estimated probabilities $\alpha_i[n]$ given by (1) with respect to the labels $y_i[n]$ over all classes i and training data n

$$H = \sum_{n=0}^{N-1}\sum_{i=0}^{S-1} y_i[n] \log \alpha_i[n] \tag{3}$$

To provide capacity control we introduce a regularizer $\Omega(\mathbf{w})$ in the objective function [8]. The parameter space $\mathbf{w}$ can be partitioned into disjoint parameter vectors $\mathbf{w}_{ij}$ for each pair of classes $i, j = 0, \ldots, S-1$ such that $P_{ij}[n]$ depends only on $\mathbf{w}_{ij}$. The regularizer can then be chosen as the L_2 norm of each disjoint parameter vector, and the objective function becomes

$$H = C\sum_{n=0}^{N-1}\sum_{i=0}^{S-1} y_i[n] \log \alpha_i[n] - \frac{1}{2}\sum_{j=0}^{S-1}\sum_{i=0}^{S-1} |\mathbf{w}_{ij}|^2 \tag{4}$$

where the regularization parameter C controls complexity versus generalization as a bias-variance trade-off [8]. The objective function (4) is similar to the primal formulation of a large margin classifier [13]. Unlike the convex (quadratic) cost function of SVMs, the formulation (4) does *not* have a unique solution and direct optimization could lead to poor local optima. However, a *lower bound* of the objective function can be formulated so that maximizing this lower bound reduces to a set of convex optimization sub-problems with an elegant dual formulation in terms of support vectors and kernels. Applying the convex property of the $-\log(.)$ function to the convex sum in the forward estimation (1), we obtain directly

$$H \geq \sum_{j=0}^{S-1} H_j \tag{5}$$

where

$$H_j = \sum_{n=0}^{N-1} C_j[n] \sum_{i=0}^{S-1} y_i[n] \log P_{ij}[n] - \frac{1}{2}\sum_{i=0}^{S-1} |\mathbf{w}_{ij}|^2 \tag{6}$$

with effective regularization sequence

$$C_j[n] = C\alpha_j[n-1]\,. \tag{7}$$

Disregarding the intricate dependence of (7) on the results of (6) which we defer to the following section, the formulation (6) is equivalent to regression of conditional probabilities $P_{ij}[n]$ from labeled data $\mathbf{x}[n]$ and $y_i[n]$, for a given outgoing state j. Estimation of conditional probabilities $\Pr(i|\mathbf{x})$ from training data $\mathbf{x}[n]$ and labels $y_i[n]$ can be obtained using a regularized form of kernel logistic regression [9]. For each outgoing state j, we construct one such probabilistic model for the incoming state i conditional on $\mathbf{x}[n]$:

$$P_{ij}[n] = \exp(-\mathbf{w}_{ij}.\mathbf{x}[n]) / \sum_{s=0}^{S-1} \exp(-\mathbf{w}_{sj}.\mathbf{x}[n])\,. \tag{8}$$

As with SVMs, the dot products in (8) convert into kernel expansions over the training data $\mathbf{x}[m]$ by transforming the data to feature space [12]

$$\mathbf{w}_{ij}.\mathbf{x} = \sum_m \lambda_{ij}^m\ K(\mathbf{x}[m], \mathbf{x}) \tag{9}$$

where $K(\cdot,\cdot)$ denotes any symmetric positive-definite kernel that satisfies the Mercer condition, such as a Gaussian radial basis function or a polynomial spline [8, 14]. Parameters λ_{ij}^m are determined by maximizing a dual formulation of the objective function (6) through the Legendre transformation, which for logistic regression takes the form of an entropy-based potential function in the parameters [9]. We use a Newton-Ralphson iterative optimization scheme to arrive at dual parameter estimates λ_{ij}^m.

2.3. Recursive FDKM Training

The weights (7) in (6) are recursively estimated using an iterative procedure reminiscent of (but different from) expectation maximization. The procedure involves computing new estimates of the sequence $\alpha_j[n-1]$ to train (6) based on estimates of P_{ij} using previous values of the parameters λ_{ij}^m. The training proceeds in a series of epochs, each refining the estimate of the sequence $\alpha_j[n-1]$

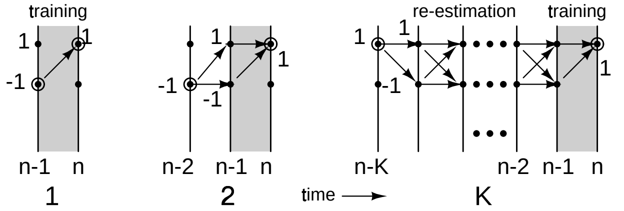

Fig. 3. *Iterations involved in training FDKM on a trellis based on the Markov model of Figure 2. During the initial epoch, parameters of the probabilistic model, conditioned on the observed label for the outgoing state at time $n-1$, of the state at time n are trained from observed labels at time n. During subsequent epochs, probability estimates of the outgoing state at time $n-1$ over increasing forward decoding depth $k = 1, \ldots K$ determine weights assigned to data n for training each of the probabilistic models conditioned on the outgoing state.*

by increasing the size of the time window (decoding depth, k) over which it is obtained by the forward algorithm (1).

The training steps are illustrated in Figure 3 and summarized as follows:

1. To bootstrap the iteration for the first training epoch ($k = 1$), obtain initial values for $\alpha_j[n-1]$ from the labels of the outgoing state, $\alpha_j[n-1] = y_j[n-1]$. This corresponds to taking the labels $y_i[n-1]$ as true state probabilities.
2. Train logistic kernel machines, one for each outgoing class j, to estimate the parameters in $P_{ij}[n]$ $i, j = 1, .., S$ from the training data $\mathbf{x}[n]$ and labels $y_i[n]$, weighted by the sequence $\alpha_j[n-1]$.
3. Re-estimate $\alpha_j[n-1]$ using the forward algorithm (1) over increasing decoding depth k, by initializing $\alpha_j[n-k]$ to $y[n-k]$.
4. Re-train, increment decoding depth k, and re-estimate $\alpha_j[n-1]$, until the final decoding depth is reached ($k = K$).

3. CHANNEL EQUALIZATION USING FDKM

FDKM can be directly applied to channel equalization, as depicted in Figure 1. Denote $u[n]$ a source of equiprobable binary symbols sent over a channel. The FDKM model can be made to fit any discrete time transmission channel by grouping L outputs of the channel into feature vectors

$$\mathbf{X}[n] = [x[n]x[n-1]...x[n-L+1]]^T \ . \qquad (10)$$

For training, the target label is taken as the input to the channel delayed by D samples, *i.e.*, $y[n] = u[n-D]$. The output of the channel $x[n] \in R$ is modeled as the sum of a deterministic function of $u[n]$ and additive white noise $e[n]$. The goal of the equalizer is to reproduce the desired output $u(n-D)$. The deterministic portion of the channel model could consist, for instance, of a linear finite impulse response (FIR) filter followed by a polynomial nonlinearity.

Note that FDKM does not actually make use of a channel model. Instead it adaptively parameterizes the state transition probabilities in the forward decoding from the training data.

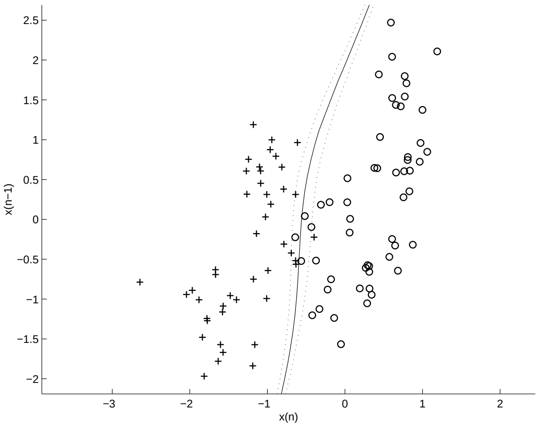

Fig. 4. *Trained probabilistic model $P(y|0, \mathbf{x})$ obtained by logistic regression before re-estimation ($k = 1$).*

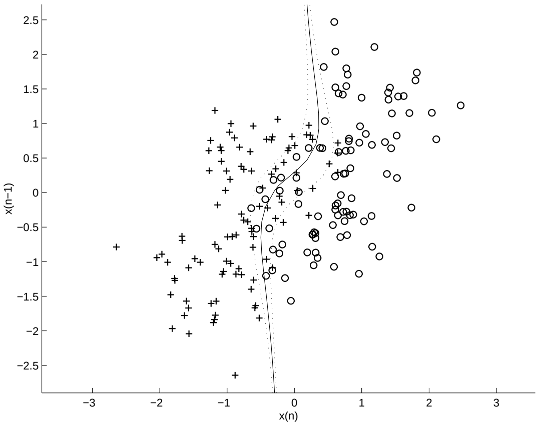

Fig. 5. *Trained probabilistic model $P(y|0, \mathbf{x})$ obtained by logistic regression after re-estimation ($k = K = 5$).*

Therefore one can extend the channel model to include, for instance, the effect of source coding.

4. EXPERIMENTS AND RESULTS

For a comparative figure of merit, we replicated linear channel models of [11, 5, 10]:

$$\begin{aligned} \tilde{x}[n] &= u[n] + 0.5u[n-1] && (11) \\ \tilde{x}[n] &= 0.407u[n] + 0.815u[n-1] + 0.407u[n-2] && (12) \end{aligned}$$

from which we generated datasets 1 and 2, respectively. 200 samples each were generated for training FDKM, and 10,000 samples each for testing. The buffer length L was chosen to be 2 and the equalizer delay D was taken to be 1. Therefore, the extent of ISI for the second data set exceeds the time horizon L of the feature vector $\mathbf{X}$. Nevertheless, the decoding depth for FDKM was chosen to be $K = 5$. For all the experiments reported here, a polynomial kernel of degree 3 was used, $K(\mathbf{x}[m], \mathbf{x}) = (1 + \mathbf{x}[m].\mathbf{x})^3$.

Figures 4 and 5 illustrate the improvements in margin distribution of the probabilistic model obtained after 5 epochs of the

Table 1. *Performance of equalization schemes at 6dB SNR*

Machine	Train1	Test1	Train2	Test2
SVM	7.3%	8%	14.2%	23%
Logistic regression	14.4%	16%	26%	34%
FDKM ($K = 1$)	0.1%	0.9%	0.4%	1.6%
FDKM ($K = 5$)	0.1%	0.1%	0.6%	0.8%

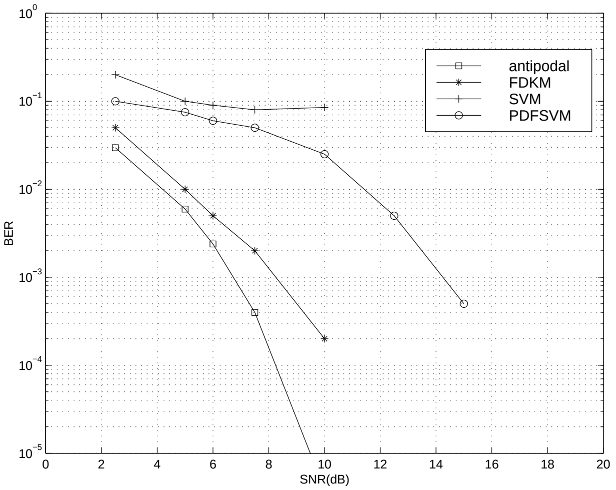

Fig. 6. *Performance evaluation for the channel model (12). The FDKM equalizer delivers a performance near that of the ISI-free theoretical limit.*

iterative FDKM procedure. Table 1 gives the comparative BER figures at 6dB SNR. The regularization parameter for SVM was $C = 5$, and for FDKM $C = 1$ and $K = 5$.

Interestingly, Table 1 indicates that a generative probability model obtained by logistic regression gives worse decoding performance than a discriminant model by SVM classification. However, the probabilistic model coupled with forward decoding gives a drastic improvement in decoding performance. Figure 6 compares the performance of FDKM with SVM and Perfect Decision Feedback SVM (PDFSVM) [11], and the theoretical optimum for non-ISI binary signaling. One can see that FDKM equalization delivers about a 4-5dB improvement in SNR for given BER, over PDFSVM and other techniques.

In another experiment we tested the performance of FDKM equalization with nonlinear channels in the presence of colored noise [11]. FDKM equalizer was found to perform nearly as well as the theoretical Bayes-optimal equalizer.

5. CONCLUSIONS

We presented a forward decoding architecture for large margin classifiers and evaluated its performance for combating ISI and nonlinearity due to the communication channel. Simulations have shown that the equalizer outperforms several other adaptive estimation techniques. The decoding architecture is feedforward in nature and hence very amenable to hardware implementation [4]. The FDKM approach is model-free, and extends directly to account for various forms of source coding.

6. REFERENCES

[1] Bahl, L.R., Cocke J., Jelinek F. and Raviv J. “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Inform. Theory*, vol. **IT-20**, pp. 284-287, 1974.

[2] Boser, B., Guyon, I. and Vapnik, V., “A training algorithm for optimal margin classifier,” in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp 144-52, 1992.

[3] Bourlard, H. and Morgan, N., *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic, 1994.

[4] Chakrabartty, S., Singh, G. and Cauwenberghs, G. “Hybrid Support vector Machine/Hidden Markov Model Approach for Continuous Speech recognition,” *Proc. IEEE Midwest Symp. Circuits and Systems (MWSCAS’2000)*, Lansing, MI, Aug. 2000.

[5] Chen, S., Mulgrew B. and McLaughlin S. “Adaptive Bayesian Equalizer with Decision Feedback,” *IEEE Transactions on Signal Processing*, vol. **41**, September 1993.

[6] Chen, S., Mulgrew B. and Grant P.M. “A clustering technique for digital communications channel equalization using radial basis function networks,” *IEEE Transactions on Neural Networks*, vol.4 (4), 1993.

[7] Forney, G. “Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference,” *IEEE Transactions on Inform. Theory*, vol. **IT-18**, pp. 363-378, 1972.

[8] Girosi, F., Jones, M. and Poggio, T. “Regularization Theory and Neural Networks Architectures,” *Neural Computation*, vol. **7**, pp 219-269, 1995.

[9] Jaakkola, T. and Haussler, D. “Probablistic kernel regression models,” *Proceedings of Seventh International Workshop on Artificial Intelligence and Statistics* , 1999.

[10] Proakis, G.J, *Digital Communications* 3rd ed. New York: McGraw-Hill, 1995.

[11] Sebald, D. and Bucklew A. James. “Support Vector Machine Techniques for Nonlinear Equalization,” *IEEE Transactions on Signal Processing*, vol. **48**, pp 3217-3226, November 2000.

[12] Scholkopf, B., Burges, C. and Smola, A., eds., *Advances in Kernel Methods-Support Vector Learning*, MIT Press, Cambridge, 1998.

[13] Vapnik, V. *The Nature of Statistical Learning Theory,* New York: Springer-Verlag, 1995.

[14] Wahba, G. *Spline Models for Observational Data*, CBMF-NSF Regional Conference Series in Applied Mathematics, vol. 59, Philadelphia PA: SIAM, 1990.