

GENERALIZATION AND MAXIMUM LIKELIHOOD FROM SMALL DATA SETS

William Byrne
Institute for Systems Research and
Department of Electrical Engineering
University of Maryland
College Park, MD 20742
bbyrne@src.umd.edu

INTRODUCTION

An often encountered learning problem is maximum likelihood training of exponential models. When the state is only partially specified by the training data, iterative training algorithms are used to produce a sequence of models that assign increasing likelihood to the training data. Although the performance as measured on the training set continues to improve as the algorithms progress, performance on related data sets may eventually begin to deteriorate. The cause of this behavior can be seen when the training problem is stated in the Alternating Minimization framework [1]. A modified maximum likelihood training criterion is suggested to counter this behavior. It leads to a simple modification of the learning algorithms which relates generalization to learning speed. Training Boltzmann Machines [2] and Hidden Markov Models [3, 4, 5, 6] is discussed under this modified criterion.

PROBLEM STATEMENT

A detailed presentation of this material is available in [7]. The visible portion of the model state is denoted y and the hidden portion x and they are assumed to come from a set $\mathcal{Y} \times \mathcal{X}$. The models are parameterized by a finite dimensional vector w and further specified by complete-data sufficient statistics $g(y, x)$. The models have the form

$$Q(y, x) = \exp(w \cdot g(y, x) + w_0). \quad (1)$$

The model family \mathcal{Q} is a subset of \mathcal{P} , the distributions on $\mathcal{Y} \times \mathcal{X}$. The probability that a machine is found in a given visible state is taken as the marginal $Q(y) = \sum_x Q(y, x)$.

Training data, $T \subset \mathcal{Y}$, is provided and the training goal is to find a model Q under the likelihood criterion:

$$\max_{Q \in \mathcal{Q}} \prod_{y \in T} Q(y). \quad (2)$$

An equivalent statement of the problem is obtained through the information divergence and the empirical distribution \hat{P}

$$\hat{P}(y) = \frac{\#_T(y)}{|T|} \quad (3)$$

where $\#_T(y)$ is the number of times y appears in the training set T ; the support of \hat{P} is

$$T = \{y : y \in T\}. \quad (4)$$

The empirical distribution defines a set of desired distributions under the marginal likelihood criterion

$$\mathcal{D} = \{P \in \mathcal{P} : \sum_x P(y, x) = \hat{P}(y) \quad \forall y \in T\}. \quad (5)$$

The goodness of a model is measured by its distance to \mathcal{D} under the information divergence

$$D(\mathcal{D} \parallel Q) = \min_{P \in \mathcal{D}} D(P \parallel Q) \quad (6)$$

where the divergence is defined

$$D(P \parallel Q) = \sum_{x,y} P(x,y) \log \frac{P(x,y)}{Q(x,y)}. \quad (7)$$

The goal of training is to find $\min_{Q \in \mathcal{Q}} D(\mathcal{D} \parallel Q)$.

ALTERNATING MINIMIZATION PROCEDURE

Under the alternating minimization procedure [1], an initial model Q^1 is chosen from the model family \mathcal{Q} . First, the I-Projection of Q^1 on \mathcal{D} is computed

$$P^1 : D(P^1 \parallel Q^1) = \min_{P \in \mathcal{D}} D(P \parallel Q^1). \quad (8)$$

A new model Q^2 is then found by solving

$$Q^2 : D(P^1 \parallel Q^2) = \min_{Q \in \mathcal{Q}} D(P^1 \parallel Q). \quad (9)$$

Repeatedly applying this procedure [1] produces a sequence of models $\{Q^p\}$ which approaches the set of desired distributions

$$D(\mathcal{D} \parallel Q^{p+1}) \leq D(\mathcal{D} \parallel Q^p) \quad (10)$$

and yields improvement in likelihood in that

$$\sum_{y \in \mathcal{T}} \hat{P}(y) \log Q^{p+1}(y) \geq \sum_{y \in \mathcal{T}} \hat{P}(y) \log Q^p(y). \quad (11)$$

In [1] it is shown that these two steps comprise the EM algorithm [8].

GENERALIZATION FROM SMALL DATA SETS

In many applications the observations which form the training set are only a small subset of the possible values which the observed variable can assume. Although training algorithms can only exploit data available in \mathcal{T} , the resulting model is intended to describe related data not found in the training set. If a training procedure produces a model with this property, it is said to *generalize* well from the training data. One approach to generalization is to find the smallest model which can describe the training data, as in [9]. However in many modeling tasks the model size is fixed beforehand. A technique is suggested here which may be useful in encouraging generalization in models which are “too large” for the training data.

One of the characteristics of the desired distributions is that their “visible support” is restricted to the training set. For $P \in \mathcal{D}$, it follows from Equation 3 that $P(\mathcal{T}) = 1$. This implies that

$$P \in \mathcal{D} : P(y, x) = 0 \quad \forall y \in \mathcal{T}^c, \forall x \in \mathcal{X} \quad (12)$$

because for $y \in \mathcal{T}^c$, $P(y, x) \leq P(y) \leq P(\mathcal{T}^c) = 0$.

If a training algorithm were to succeed in finding an exact, optimum solution, that is, to find a model Q^* such that $D(\mathcal{D} \parallel Q^*) = 0$, this model would necessarily belong to \mathcal{D} . Although the model would be optimum according to the maximum likelihood criterion, it would be unable to generalize about data not in the training set. In a sense, the ability of the training algorithm to produce models which generalize well requires that the algorithm not achieve its training goal.

In most applications these algorithms only find suboptimal solutions which do not belong to \mathcal{D} , but it is possible for the resulting models to be overtrained in the following sense. Suppose that an additional set of

data V , called the validation set, is also available during training. The validation data defines a family of distributions \mathcal{V} in the same way as the training data defines \mathcal{D}

$$\mathcal{V} = \{P \in \mathcal{P} : \sum_x P(y, x) = \frac{\#_V(y)}{|V|}\}. \quad (13)$$

Ideally, while the training algorithm yields improving performance on the training set according to Equation 10, the performance on the validation set also continues to improve

$$D(\mathcal{V} \parallel Q^{p+1}) \leq D(\mathcal{V} \parallel Q^p). \quad (14)$$

If at some iteration this relationship is violated, the model is said to be overtrained. This overtraining is evidence of poor generalization.

THE SMALL DATA SET ASSUMPTION

The discussion that follows makes use of the small data set assumption

$$Q(T) \approx 0 \quad \forall Q \in \mathcal{Q} \quad (15)$$

or, similarly, $Q(T) \ll Q(T^c)$. Whether or not this is true depends upon the power of the models and the complexity of the training task. For the Boltzmann Machine, it could be plausible for large networks. For Hidden Markov Models used in speech recognition, it is usually true and leads to the well known problem of numerical underflow in HMM training algorithms.

A MODIFIED SET OF DESIRABLE DISTRIBUTIONS

Under the small data assumption the model family and desirable family are poorly matched in that $P(T) = 1$ while $Q(T) \approx 0$, for $P \in \mathcal{D}$ and $Q \in \mathcal{Q}$. A possible way to avoid this mismatch is to modify the definition of the set of desired distributions (Equation 5) by introducing a confidence parameter c

$$\mathcal{D}_c = \{P \in \mathcal{P} : \sum_x P(y, x) = c \hat{P}(y) \quad \forall y \in T\} \quad 0 \leq c \leq 1. \quad (16)$$

In this set of distributions the training set is given probability $P(T) = c$, while all other possible observations are given probability $P(T^c) = 1 - c$.

In typical applications of the EM algorithm the likelihood criterion completely specifies the desired distribution over the observed variable and the E-Step is used only to estimate the desired behavior of the hidden variables. Under the likelihood criterion presented here however, the likelihood criterion is incompletely specified outside the training set. No assumptions are made about the correct likelihood of the individual elements of T^c , other than that the linear constraint $P(T^c) = 1 - c$ is satisfied. As a result, the E-Step also estimates at each iteration the likelihood criterion where it is unspecified. Performing the I-Projection under the incomplete linear constraint leads to an estimate of the unspecified likelihood criterion according to the minimum discrimination information principle [10].

IMPOSING THE CONFIDENCE CONSTRAINT

The projection of Q^p onto \mathcal{D}_c has the following form

$$P^p(y, x) = \begin{cases} c \hat{P}(y) Q^p(x|y) & y \in T \\ (1 - c) Q^p(y, x)/Q^p(T^c) & y \in T^c \end{cases} \quad (17)$$

which under the small data assumption becomes

$$P^p(y, x) = \begin{cases} c \hat{P}(y) Q^p(x|y) & y \in T \\ (1 - c) Q^p(y, x) & y \in T^c \end{cases} \quad (18)$$

The relationship $D(\mathcal{D}_c \parallel Q^{p+1}) \leq D(\mathcal{D}_c \parallel Q^p)$ holds under the confidence constraint, but the improvement in likelihood becomes

$$\begin{aligned} c \sum_{y \in \mathcal{T}} \hat{P}(y) \log Q^{p+1}(y) + (1-c) \log Q^{p+1}(\mathcal{T}^c) \geq \\ c \sum_{y \in \mathcal{T}} \hat{P}(y) \log Q^p(y) + (1-c) \log Q^p(\mathcal{T}^c). \end{aligned} \quad (19)$$

The modified algorithm attempts to improve the empirical likelihood as in Equation 11, but is penalized if the support of the models becomes concentrated on the training set.

MODEL REESTIMATES

For simple exponential families the Alternating Minimization algorithm can be summarized in terms of moment updates. The new model Q^{p+1} is computed from P^p via the expectation parameters, or moments $p^p = E_{P^p} g$

$$p^p = \sum_{y, x} P^p(y, x) g(y, x). \quad (20)$$

In the Boltzmann Machine these parameters are found by clamping [11] and in HMMs they are usually found through the Forward-Backward algorithm [4]; in general, this is the E-Step in the EM algorithm [8].

The parameters w^{p+1} are then found so that the moments of Q^{p+1} , $q^{p+1} = E_{Q^{p+1}} g$, agree with the moments of P^p

$$w^{p+1} : q^{p+1} = p^p. \quad (21)$$

In the usual $c = 1$ case, the moment reestimates produced by the EM algorithm, denoted \tilde{p}^p , are found as

$$\tilde{p}^p = \sum_{y, x} \hat{P}(y) Q^p(x|y) g(y, x). \quad (22)$$

When $c < 1$, the desired moments are

$$p^p = c \tilde{p}^p + (1-c) \frac{1-c}{Q^p(\mathcal{T}^c)} [q^p - \sum_{I \in \mathcal{T}, S} g(I, S) Q^p(I, S)] \quad (23)$$

which, under the small data set assumption becomes

$$p^p = c \tilde{p}^p + (1-c) q^p. \quad (24)$$

If Equation 21 is satisfied exactly at the previous iteration this becomes

$$p^p = c \tilde{p}^p + (1-c) p^{p-1}. \quad (25)$$

The modification to the set of desirable distributions effectively slows the Alternating Minimization procedure by low-pass filtering the moment reestimation process.

Equation 25 leads to the descriptive term *moment decay*. In exponential models the weights and moments are *dual* [12] parameter sets, so in a sense this technique is dual to the well known weight-decay technique. The parameter and moment spaces are related in a curved manner, however, so filtering the moments before parameter reestimation may yield a very different model than filtering the parameters after reestimation.

Sequential Algorithms

In the Alternating Minimization procedure the model parameters are reestimated so that Equation 21 holds exactly

$$w^{p+1} : q^{p+1} = p^p. \quad (26)$$

In Maximum Likelihood HMM training, the Baum-Welch algorithm yields in one step a model which satisfies this relationship so that both steps of the alternating minimization procedure are satisfied exactly. In training

other models, such as the Boltzmann Machine [2], no such one-step, exact algorithm is available. Typically sequential, gradient descent algorithms are used to find the model with the correct moments. In some cases it is possible to implement the gradient search so that Equation 21 holds, for example Boltzmann Machine training can be formulated using Iterative Proportional Fitting to meet this constraint [11]. In other cases, though, the model parameters are modified in the direction of decreasing $D(\mathcal{D}_c \parallel Q)$, and the M-step is not solved exactly. This leads to a stochastic approximation to the Alternating Minimization algorithm (e.g. [13], Eq. 4) which uses sequential rather than exact updating.

The model parameters are updated as

$$w^{p+1} = w^p - \alpha \nabla_w D(\mathcal{D}_c \parallel Q)|_{Q^p} \quad (27)$$

for a small step size α . It can be shown that for the marginal likelihood scoring

$$\nabla_w D(\mathcal{D}_c \parallel Q)|_{Q^p} = -c (\tilde{p}^p - q^p) + (1 - c) \frac{1}{Q^p(\mathcal{T}^c)} [Q^p(\mathcal{T}) q^p - \sum_{\mathcal{T}, \mathcal{S}} g(I, S) Q^p(I, S)] \quad (28)$$

which under the small data set assumption becomes

$$\nabla_w D(\mathcal{D}_c \parallel Q)|_{Q^p} = -c (\tilde{p}^p - q^p) \quad (29)$$

$$= c \nabla_w D(\mathcal{D}_{c=1} \parallel Q)|_{Q^p}. \quad (30)$$

The parameter update is then

$$w^{p+1} = w^p - c \alpha \nabla_w D(\mathcal{D}_{c=1} \parallel Q)|_{Q^p} \quad (31)$$

The confidence parameter leads directly to slower learning in these sequential algorithms by reducing the step size.

Clamping in Boltzmann Machine Learning

Boltzmann Machines [2] are artificial neural networks of binary valued, stochastic units which can be made to learn in an approximation of the EM algorithm. Learning proceeds in two phases. The network visible units are *clamped* according to an environmental distribution \hat{P} and the hidden units operate freely. In the free-running phase, all units are allowed to operate freely. Statistics are accumulated while the network is operating in each mode, and then the network weights are modified so that the free-running behavior of the network better matches the clamped behavior.

The steady-state distribution of the clamped network corresponds to the I-Projection of the current model onto \mathcal{D} [11]

$$\bar{Q}(I, S) = Q(S|I) \hat{P}(I). \quad (32)$$

Under the discounted data criterion and the small data set assumption, the steady-state distribution according to the I-Projection onto \mathcal{D}_c is

$$\bar{Q}(I, S) = Q(S|I) [c \hat{P}(I) + (1 - c) Q(I)]. \quad (33)$$

The discounted data criterion effectively leads to a modified environmental distribution. While the statistics are accumulated, the network is clamped as usual to \hat{P} for a portion c of the duration of the accumulation. For the remaining portion of time, data is collected with the network free-running. Moment decay can be implemented in the Boltzmann Machine distributed computational architecture with only a minor change in the usual learning rule. Whether the weights are updated in an exact or sequential manner, it is not necessary to store the moments from the previous iteration because they are found again during the free-running portion of the moment accumulation.

TRAINING HIDDEN MARKOV MODELS

The architectural and algorithmic relationships between Neural Networks and Hidden Markov Models have been well studied [5, 6, 14, 15]. Maximum likelihood HMM training under the modified criterion presented

here is easily implemented and is described in detail in [7]. An example of applying the discounted likelihood criterion to modeling phonemes in context is presented in Figure 1.

A context-dependent model is trained using only instances of the phoneme found in that context. A shortcoming of this approach is that it is difficult to find enough instances of a phoneme in all possible contexts to train reliable context-dependent models. Here, the phoneme /k/ is modeled in the context /cl-k-ix/ as in “kettle” using data taken from Dialect Region 1 of the TIMIT database. The HMM trained is a three state, left-to-right model with a single mixture diagonal covariance Gaussian observation distributions. The observations are 12 Mel frequency cepstral coefficients, an energy term and their difference. The parameters and the model reestimates are computed using the HTK Hidden Markov Model Toolkit [16].

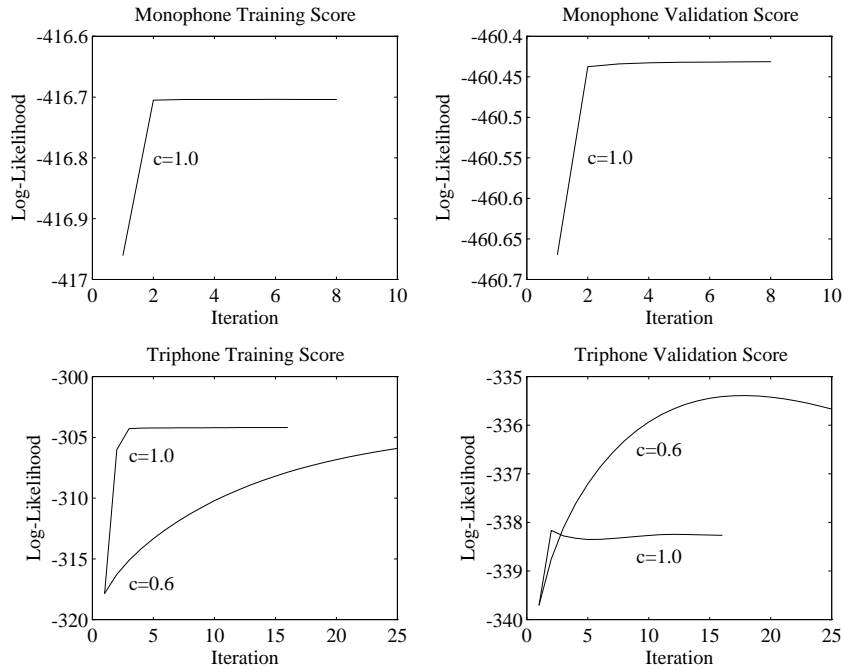


Figure 1: (Top) Training and validation scores in training a monophone model for /k/ using the $c = 1$ Baum Welch algorithm. $|\mathcal{T}| = 296$ and $|\mathcal{V}| = 78$. (Bottom) Training and validation scores in training a triphone model for /cl-k-ix/ using the Baum Welch algorithm for $c = 1.0$ and $c = 0.6$. $|\mathcal{T}| = 25$ and $|\mathcal{V}| = 7$.

Typically a reliable monophone (context-independent) model is first trained using the usually large amount of context-independent training data. This model is then used as an initial model in further training with context-dependent data. In the example shown here, no overtraining is evident in the initial monophone model. However, in further training using the much smaller amount of context dependent data, overtraining is observed in the usual $c = 1$, version of the Baum Welch algorithm after the first iteration. Under the modified likelihood criterion for for $c = 0.6$, however, the onset of overtraining is postponed and the overall score on the validation set is improved.

CONCLUSION

A technique has been described which can be used to prevent overtraining and encourage generalization in training under a maximum likelihood criterion. Applications to Boltzmann Machines and Hidden Markov Models are discussed. While the confidence constraint may slow the training algorithm, in general it should involve very little additional calculation.

The results presented here for HMMs are for training under a maximum likelihood criterion based on the marginal distribution. Similar modifications can be made to the Segmental K-Means and N-Best algorithms.

A manner for choosing the correct value of c is not known. While it appears that lower values of c encourage generalization, lower values of c also slow the training algorithms. In this formulation there is a clear trade-off between generalization and speed of learning.

References

- [1] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions, Supplementary Issue Number 1*, pages 205–237, 1984.
- [2] G. E. Hinton, T. J. Sejnowski, and D. H. Ackley. Boltzmann Machines: Constraint satisfaction networks that learn. Technical Report CMU-CS-84-119, Carnegie-Mellon University, Pittsburgh, PA 15213, May 1984.
- [3] L.E.Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, 41(1):164–171, 1970.
- [4] S.E.Levinson. Structural methods in automatic speech recognition. *Proceedings of the IEEE*, 73(11):1625–1650, November 1985.
- [5] J. S. Bridle. Alpha-nets: a recurrent Neural Network architecture with a Hidden Markov Model interpretation. *Speech Communication*, 9:83–92, 1990.
- [6] E. Levin. Hidden Control neural architecture modeling of nonlinear time varying systems and its applications. *IEEE Transactions on Neural Networks*, 4(1), 1 January 1993.
- [7] W. J. Byrne. An information geometric treatment of maximum likelihood criteria and generalization in Hidden Markov Models. Technical Report 93-50, Institute for Systems Research, University of Maryland, College Park, MD, 20742, 1993.
- [8] A.P.Dempster, N.M.Laird, and D.B.Rubin. Maximum likelihood from incomplete data. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [9] E. B. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, 1, 1989.
- [10] S. Kullback. *Information theory and statistics*. Wiley, New York, 1959.
- [11] W. J. Byrne. Alternating Minimization and Boltzmann Machine learning. *I.E.E.E. Transactions on Neural Networks*, 3(4):612–620, 1992.
- [12] S.-I. Amari. *Differential-Geometrical methods in statistics*. Springer-Verlag, New York, 1985.
- [13] E. Weinstein, M. Feder, and A. V. Oppenheim. Sequential algorithms for parameter estimation based on the Kullback-Leibler information measure. *I.E.E.E. Transactions on Acoustics, Speech, and Signal Processing*, 38(9):1652–1654, September 1990.
- [14] L. Niles and H. Silverman. Combining Hidden Markov Models and Neural Network classifiers. In *International Conference on Acoustics, Speech and Signal Processing*, pages 417–420, 1990.
- [15] A. Kehagias. Optimal Control for training: The missing link between Hidden Markov Models and Connectionist Nnetworks. Technical report, Division of Applied Mathematics, Brown University, Providence, RI 02912, June 1989.
- [16] P. Woodland and S.J.Young. Benchmark DARPA RM results with the HTK portable HMM toolkit. In *Proceedings of the Speech and Natural Language Workshop*, September 1992.