THE GRADIENT PROJECTION METHOD FOR THE TRAINING OF HIDDEN MARKOV MODEL

Qiang Huo [†‡] & Chorkin Chan [†]

[†]Department of Computer Science
University of Hong Kong, Hong Kong

[‡]Department of Radio and Electronics
University of Science and Technology of China, P.R.C.

ABSTRACT: By looking at the training of HMM as a general constrained optimization problem with linear constraints, in this paper, a gradient projection method for nonlinear programming with linear constraints has been presented to solve for "optimal" values of the model parameters. The presented algorithm has been shown to be convergent and to have a linear convergence rate. When this method is applied to the training of HMMs with discrete or Gaussian mixture observation densities, a very simple formulation has been derived due to the special structure of the constraints of HMM parameters.

INTRODUCTION

Hidden Markov Models (HMMs) owe their current popularity to the existence of an efficient training procedure, the Baum-Welch algorithm. While this kind of reestimation formulas provides an elegant method for finding a local maximum of the objective function (likelihood function), their success depends critically on the particular form of the objective function and constraints imposed on the HMM (Baum and Egon, 1967; Baum et al., 1970; Liporace, 1982; Juang, 1985). There are many cases of HMM training for speech recognition where the conditions required by the Baum-Welch formulation are apparently not satisfied.

Generally speaking, in HMM-based speech recognition, the purpose of training is to find the HMM parameter set λ which will result in a decoder of the lowest possible recognition error rate. This is done by maximizing (or minimizing) some objective function $R(\lambda)$. There are thus two important and difficult problems to consider. The first is to determine a meaningful objective function which should be such that, whenever $R(\bar{\lambda})>R(\lambda)$, then $\bar{\lambda}$ produces a better decoder than that from λ. Once a function $R(\lambda)$ has been chosen, the second problem (the estimation problem) is to find the parameter set $\bar{\lambda}$ which maximizes it. We are not concerned about the first problem, but just focus our attention on the second one in this paper.

In this paper, by looking at the training problem of HMM as a general constrained optimization problem with linear constraints, a gradient projection method for nonlinear programming with linear constraints will be presented to solve for the "optimal" values of the model parameters. When this method is applied to the training of HMMs with discrete or Gaussian mixture observation densities, a very simple formulation will be derived due to the special structure of the constraints of HMM parameters, and this derivation requires no special form of the objective function.

THE BASIC ALGORITHM OF THE GRADIENT PROJECTION METHOD

Historically, the gradient projection method was proposed and extensively analyzed by Rosen (1960). Let x_i, $i=1,2,\cdots,m$, be the coordinates of a point $\mathbf{x}$ in the m-dimensional Euclidean space E^m. $\mathbf{x}$ can also be represented by a column vector whose transpose is $\mathbf{x}^T=(x_1, x_2, \cdots, x_m)$. Let

$f(\mathbf{x}) = f(x_1, x_2, \cdots, x_m)$ be the objective function with continuous and bounded first partial derivatives with respect to the x_i's. In this notation, the general nonlinear programming problem with linear constraints can be expressed as:

$$\textit{Maximize} \quad f(\mathbf{x}) \tag{1}$$

subject to linear equalities and inequalities of the form:

$$\mathbf{n}_i^T\mathbf{x} - c_i = 0 \quad i=1,2,\cdots,k \tag{2}$$

$$\mathbf{n}_i^T\mathbf{x} - c_i \geq 0 \quad i=k+1,\cdots,p \tag{3}$$

where the $\mathbf{n}_i$ are unit normals and the c_i are scalars. For simplicity, the unit normals $\{ \mathbf{n}_i \}$ are assumed linearly independent. The constraints restrict the solution to k hyperplanes and $p-k$ closed half-spaces. Their intersection R is, in general, a convex polyhedron and called the "feasible region".

Corresponding to each of the p constraints, an $(m-1)$-dimensional manifold can be defined which is a hyperplane denoted by H_i: $\mathbf{n}_i^T\mathbf{x} - c_i = 0$, $i=1,2,\cdots,p$. Considering any q linearly independent hyperplanes $H_1, H_2, \cdots, H_q$, the intersection of them is in general an affine subspace (or flat) of E^m and will be denoted by F_q. If all of these q hyperplanes contain the origin, this intersection is a linear manifold denoted by M_q. When movement is restricted to a particular flat F_q, the linear manifold M_q parallel to it is the "constraint manifold". With an $m \times q$ matrix $\mathbf{N}_q$ defined as: $\mathbf{N}_q = [\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_q]$, it can be shown (Rosen, 1960) that an $m \times m$ matrix defined by

$$\mathbf{P}_q = \mathbf{I} - \mathbf{N}_q(\mathbf{N}_q^T\mathbf{N}_q)^{-1}\mathbf{N}_q^T \tag{4}$$

is a projection matrix which projects any vector in E^m onto the constraint manifold M_q parallel to F_q, where $\mathbf{I}$ is an $m \times m$ identity matrix. So, if $\mathbf{x}_0$ is in F_q, and $\mathbf{y} \in E^m$, then $\mathbf{x} = \mathbf{x}_0 + \mathbf{P}_q\mathbf{y}$ is still in F_q.

The main idea of the gradient projection method is to search along the projection of the gradient on the constraint space for a local maximum. Suppose the current "active constraints" correspond to the q hyperplanes as discussed above, then the search direction can be computed as $\mathbf{z} = \mathbf{P}_q\mathbf{g}(\mathbf{x})$, where $\mathbf{g}(\mathbf{x})$ denotes the gradient of $f(\mathbf{x})$. One can easily show that if $\mathbf{z} \neq 0$, then it is a direction of ascent for $f(\mathbf{x})$. Since $\mathbf{g}(\mathbf{x}) - \mathbf{z} = \mathbf{g}(\mathbf{x}) - \mathbf{P}_q \cdot \mathbf{g}(\mathbf{x}) = \mathbf{N}_q(\mathbf{N}_q^T\mathbf{N}_q)^{-1}\mathbf{N}_q^T\mathbf{g}(\mathbf{x})$ is orthogonal to $\mathbf{z}$, we have $\mathbf{g}^T(\mathbf{x}) \cdot \mathbf{z} = (\mathbf{g}^T(\mathbf{x}) - \mathbf{z}^T + \mathbf{z}^T) \cdot \mathbf{z} = \|\mathbf{z}\|^2$. Thus, if $\mathbf{z} \neq 0$, then $\mathbf{g}^T(\mathbf{x})\mathbf{z} > 0$ and it is a feasible direction of ascent on the working surface. So, the gradient projection method is essentially a steepest ascent method in the flat F_q defined by "the active constraints". Note that the "equality constraints" in (2) are always "active". The basic iterative algorithm of the gradient projection method can be stated as follows:

Let i denote the iteration number, at any iterate $\mathbf{x}^{(i)}$, q denotes the number of active constraints in the working set and I_i denote the set of indices of these constraints. Assume that a feasible starting point $\mathbf{x}^{(0)}$ is given. If $\mathbf{x}^{(0)}$ lies in the intersection (F_q) of q linearly independent hyperplanes, then these constraints are added to the working set. The k hyperplanes that correspond to the equality constraints (2), where $0 \leq k \leq q$, are added first followed by the other inequality constraints. Set $i \leftarrow 0$. Determine I_0 and q and execute the following steps :

Step 1: [Test for convergence]

Given $\mathbf{x}^{(i)}$, $\mathbf{g}^{(i)}$, compute $\mathbf{P}_q \cdot \mathbf{g}^{(i)}$ and the first-order estimation of the Lagrange multipliers $\mathbf{r} = (\mathbf{N}_q^T\mathbf{N}_q)^{-1}\mathbf{N}_q^T \cdot \mathbf{g}^{(i)}$. If $\mathbf{P}_q\mathbf{g}^{(i)} = 0$ and $r_j \leq 0$, $j = k+1, k+2, \cdots, q$, then $\mathbf{x}^{(i)}$ is a (constrained) stationary point, and the algorithm terminates with $\mathbf{x}^{(i)}$ as the solution. If this is not the case,

go on to step 2.

Step 2: [Choose which logic to perform]

Decide whether to continue maximizing in the current subspace or whether to delete a constraint from the working set:

(1) if $\|\mathbf{P}_q \cdot \mathbf{g}^{(l)}\| > \max\{0, \frac{1}{2} r_s b_s^{-1/2}\}$, then continue maximizing in the current subspace by going to step 3. Here $s = \arg \max_{j=k+1,\cdots,q} \{r_j\}$ and b_s is the sth diagonal element of $(\mathbf{N}_q^T \mathbf{N}_q)^{-1}$.

(2) if $\|\mathbf{P}_q \cdot \mathbf{g}^{(l)}\| \le \frac{1}{2} r_s b_s^{-1/2}$, then go to step 7.

Step 3: [Compute a feasible search direction]

Compute $\mathbf{z}^{(l)} = \mathbf{P}_q \cdot \mathbf{g}^{(l)}$, go on to Step 4.

Step 4: [Compute a step length]

Compute $\bar{\tau}$, which represents the largest step that can be taken from $\mathbf{x}^{(l)}$ in the direction $\mathbf{z}^{(l)}$ without leaving the feasible region R :

$$\bar{\tau}_j = \left\{ \frac{c_j - \mathbf{n}_j^T \mathbf{x}^{(l)}}{\mathbf{n}_j^T \mathbf{z}^{(l)}} \;\middle|\; j \notin I_l \text{ and } \mathbf{n}_j^T \mathbf{z}^{(l)} < 0 \right\}$$

$$\bar{\tau} = \begin{cases} \min\limits_j \{\bar{\tau}_j\} & \text{if } \mathbf{n}_j^T \mathbf{z}^{(l)} < 0 \text{ for } \textit{some } j \notin I_l \\ +\infty & \text{if } \mathbf{n}_j^T \mathbf{z}^{(l)} \ge 0 \text{ for } \textit{all } j \notin I_l \end{cases}$$

Determine a positive step length $\tau^{(l)}$, $\tau^{(l)} \le \bar{\tau}$, which maximizes $f(\mathbf{x}^{(l)} + \tau^{(l)} \mathbf{z}^{(l)})$ (usually do a line search). If $\tau^{(l)} < \bar{\tau}$, go to step 6; otherwise, go on to step 5.

Step 5: [Add a constraint to the working set]

If $\tau^{(l)} = \bar{\tau}$ imposed by the constraint with index t, add t to I_l, set $q \leftarrow q+1$, and update $\mathbf{N}_q$, $\mathbf{P}_q$ accordingly. Go to step 6.

Step 6: [Update the estimate of the solution]

Set $\mathbf{x}^{(l+1)} \leftarrow \mathbf{x}^{(l)} + \tau^{(l)} \cdot \mathbf{z}^{(l)}$, $l \leftarrow l+1$, and go back to step 1.

Step 7: [Delete a constraint from the working set]

Delete the sth constraint from the working set, and set $q \leftarrow q-1$, update $\mathbf{N}_q$, $\mathbf{P}_q$, and go back to step 1.

That completes the basic algorithm. It has been shown that this algorithm is convergent, and generally speaking, the rate of convergence is linear and is determined by the eigenvalues of the Hessian of the Lagrangian restricted to the subspace tangent to the active constraints (Rosen, 1960; Luenberger, 1984).

How to do the univariate maximization efficiently in step 4 is still an open question. The gradient projection method doesn't need an accurate line search. The optimal accuracy of line search that allows the problem to be solved in the shortest time possible is problem dependent and can only be determined through extensive numerical experiments on the actual problem. It is emphasized here that it is not appropriate to compute a step length τ without restriction such as using a step-length algorithm for unconstrained optimization, and then setting τ to $\bar{\tau}$ if the unconstrained step length exceeds $\bar{\tau}$. The popular safeguarded line search procedure can be adapted to include an upper bound on the step length. For more detailed discussion of the stopping criteria for a line search, see Goldstein (1965) and Wolfe (1969).

TRAINING HMM WITH THE GRADIENT PROJECTION METHOD

Let an N-state HMM be denoted by $\lambda=(\pi, \mathbf{A}, \mathbf{B})$, where $\pi=\{\pi_i\}, i=1,2,\cdots,N$, is the initial state distribution; $\mathbf{A}=\{a_{ij}\}, i,j=1,2,\cdots,N$, the state transition matrix; and $\mathbf{B}$, the observation probability function. For the discrete HMM (DHMM), $\mathbf{B}=\{b_{jk}\}, j=1,2,\cdots,N, k=1,2,\cdots,M$, is the probability of observing symbol k, in state j. For the continuous density HMM (CDHMM), the probability density function is $\mathbf{B}=\{b_j(\mathbf{x})\}, j=1,2,\cdots,N$, and in this paper, Gaussian mixture densities of the form $b_j(\mathbf{x}) = \sum_{k=1}^{M} c_{jk} N(\mathbf{x}, \mu_{jk}, \mathbf{W}_{jk})$ are assumed, where $N(\mathbf{x}, \mu, \mathbf{W})$ denotes a D–dimensional normal density function of mean vector μ and inverse covariance matrix $\mathbf{W}$.

These parameters must satisfy the following constraints:

$$\sum_{i=1}^{N} \pi_i = 1 \quad \text{and} \quad \pi_i \geq 0 , \quad i=1,2,\cdots,N \tag{5}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{and} \quad a_{ij} \geq 0 , \quad i,j = 1,2,\cdots,N \tag{6}$$

For the discrete HMM:

$$\sum_{k=1}^{M} b_{jk} = 1 \quad \text{and} \quad b_{jk} \geq \varepsilon_1 , \quad j =1,2,\cdots,N, \; k = 1,2,\cdots,M \tag{7}$$

For the continuous HMM:

$$\sum_{k=1}^{M} c_{jk} = 1 \quad \text{and} \quad c_{jk} \geq 0 , \quad j = 1,2,\cdots,N, \; k = 1,2,\cdots,M \tag{8}$$

$$w_{jkdd} \geq \varepsilon_2 , \quad j = 1,2,\cdots,N, \; k = 1,2,\cdots,M, \; d = 1,2,\cdots,D \tag{9}$$

where $\varepsilon_1, \varepsilon_2$ are two small positive numbers, and w_{jkdd} is the d–th diagonal component of the inverse covariance matrix $\mathbf{W}_{jk}$.

For an HMM-based speech recognizer, the purpose of training is to determine the HMM parameter set λ which will result in a decoder with the lowest possible recognition error rate. This can be done by maximizing some objective function $R(\lambda)$. So the training of HMM can be viewed as a classical optimization problem with linear constraints (5) - (9). The gradient projection method presented in Section 2 can be used to solve this problem. Furthermore, one can see that those constraints in (5) - (9) can be divided into disjoint groups, i.e., no pair of constraint groups has variables in common. Each such constraint group has either the form *Type* 1: $\sum_{i=1}^{N} x_i = 1$ and $x_i \geq \varepsilon$, $i=1,2,\cdots,N$ or the form *Type* 2: $x_i \geq \varepsilon$, $i=1,2,\cdots,M$.

So, there are three kinds of variables, viz., variables with constraints of type 1, type 2 and variables with no constraint at all. Thus, all HMM parameters and their associated constraints can be divided into disjoint subsets, with the corresponding search directions computed and the working set for each subset determined independently. The overall search direction is just the concatenation of search directions of the disjoint subsets of HMM parameters.

(1) Computing the search direction for variables with type 1 constraints

For type 1 constraints above, it is assumed that there are q "active constraints". Note that the equality constraint is always active, the remaining $q-1$ active constraints corresponding to the $q-1$

variables, denoted as $x_{n_1}, x_{n_2}, \cdots, x_{n_{q-1}}$. Then one can define the "active constraint matrix" as : $\mathbf{N}_q = (\gamma_{ij})_{N\times q}$, where

$$\gamma_{i1} = \frac{1}{\sqrt{N}}, \text{ for} 1 \le i \le N \quad \text{and} \quad \gamma_{ij} = \begin{cases} 1 & \text{if } i = n_{j-1} \\ 0 & \textit{otherwise} \end{cases} \quad \text{for } j = 2,3,\cdots,q \tag{10}$$

With $\mathbf{N}_q$ defined this way, one can derive:

$$(\mathbf{N}_q^T\mathbf{N}_q)^{-1} = \frac{1}{N-q+1}\begin{bmatrix} N & -\sqrt{N} & -\sqrt{N} & \cdot & \cdot & -\sqrt{N} \\ -\sqrt{N} & N-q+2 & 1 & \cdot & \cdot & 1 \\ -\sqrt{N} & 1 & N-q+2 & & & 1 \\ \cdot & \cdot & & \cdot & & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ -\sqrt{N} & 1 & 1 & \cdot & \cdot & N-q+2 \end{bmatrix}_{q\times q} \tag{11}$$

After considerable algebraic manipulation using this result the search direction corresponding to these variables is: $\mathbf{z} = [\mathbf{I}-\mathbf{N}_q(\mathbf{N}_q^T\mathbf{N}_q)^{-1}\mathbf{N}_q^T]\cdot\nabla f(\mathbf{x})$, where

$$z_i = \begin{cases} \dfrac{\partial f}{\partial x_i} - Q & \text{for } i \ne n_1, n_2, \cdots, n_{q-1} \\ 0 & \text{for } i = n_1, n_2, \cdots, n_{q-1} \end{cases} \tag{12}$$

$$Q = \frac{1}{N-q+1}\sum_{\substack{j=1,2,\cdots,N \\ j\ne n_1,n_2,\cdots,n_{q-1}}} \frac{\partial f}{\partial x_j}$$

the first-order estimate of the Lagrange multipliers is:

$$\mathbf{r} = (\mathbf{N}_q^T\mathbf{N}_q)^{-1}\mathbf{N}_q^T\cdot\nabla f(\mathbf{x}) = [\sqrt{N}\cdot Q, \frac{\partial f}{\partial x_{n_1}} - Q, \cdots, \frac{\partial f}{\partial x_{n_{q-1}}} - Q]^T \tag{13}$$

and the sth ($s \ge 2$) diagonal element of $(\mathbf{N}_q^T\mathbf{N}_q)^{-1}$ is $\dfrac{N-q+2}{N-q+1}$. These terms will be used in step 2 of the basic algorithm discussed in Section 2 to decide if a constraint from the working set is to be deleted.

(2) Computing the search direction for variables with type 2 constraints

For type 2 constraints, one also assumes that there are q "active constraints", with the associated variables denoted by $x_{n_1}, x_{n_2}, \cdots x_{n_q}$. Then the "active constraint matrix" can be defined as: $\mathbf{N}_q = (\gamma_{ij})_{N\times q}$, for $j = 1,2,\cdots,q$ where

$$\gamma_{ij} = \begin{cases} 1 & \text{if } i = n_j \\ 0 & \textit{otherwise} \end{cases} \tag{14}$$

then one gets $\mathbf{N}_q^T\mathbf{N}_q = \mathbf{I}_{q\times q}$, The corresponding search direction $\mathbf{z}$ is such that

$$z_i = \begin{cases} \dfrac{\partial f}{\partial x_i} & \text{for } i \ne n_1, n_2, \cdots, n_{q-1} \\ 0 & \text{for } i = n_1, n_2, \cdots, n_{q-1} \end{cases} \tag{15}$$

The first-order estimate of the Lagrange multipliers is:

$$\mathbf{r} = (\mathbf{N}_q^T\mathbf{N}_q)^{-1}\mathbf{N}_q^T\cdot\nabla f(\mathbf{x}) = [\frac{\partial f}{\partial x_{n_1}}, \frac{\partial f}{\partial x_{n_2}}, \cdots, \frac{\partial f}{\partial x_{n_q}}]^T \tag{16}$$

The sth diagonal element of $(\mathbf{N}_q^T\mathbf{N}_q)^{-1}$ is simply 1.

(3) For variables with no constraint, the search direction is simply their gradient vector.

From the discussion above, one can see that when the gradient projection method is applied to HMM training, the computation can be greatly simplified due to the special structure of the constraints. This derivation does not require the objective function to be of any special form. This may prove to be an advantage since the Baum-like algorithm is not applicable to all objective function. Furthermore, the constraints may be changed. Although the Baum-Welch algorithm can be somewhat generalized to suit the new constraints, it cannot be generalized to work with arbitrary linear constraints, whereas the gradient projection method discussed in Section 2 will be applicable to such cases.

CONCLUSION

A gradient projection method for nonlinear programming with linear constraints has been presented and shown to be convergent with a linear convergence rate. When this method is applied to HMM training, a simple formulation has been derived. This general optimization technique is not only a viable alternative of the classical Baum-Welch algorithm, but it can also serve as a preferable method in general HMM training problems when the objective function and constraints fail to satisfy the conditions demanded by the Baum-Welch reestimation formulas. Due to the existence of this kind of classical optimization methods, more flexible modeling of speech signal and more sophisticated training of model parameters for speech recognition can become viable.

REFERENCES

Baum, L. E. and Egon, J. A. (1967) *An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology,* Bull. Amer. Math. Soc., 73, 360-363.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970) *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Function of Markov Chains,* The Annals of Mathematical Statistics, 41, 164-171.

Goldstein, A. A. (1965) *On Steepest Descent,* SIAM J. on Control, 3, 147-151.

Juang, B. H. (1985) *Maximum-Likelihood Estimation of Mixture Multivariate Stochastic Observations of Markov Chains,* AT&T Technical Journal, 64(6), 1235-1249.

Liporace, L. R. (1982) *Maximum likelihood estimation for multivariate observations of Markov sources,* IEEE Transactions on Information Theory, IT-28, 729-734.

Luenberger, David G. (1984) *Linear And Nonlinear Programming* (Addison-Wesley Publishing Company).

Rosen, J. B. (1960) *The Gradient Projection Method For Nonlinear Programming--Part I: Linear Constraints,* J. Soc. Indust. Appl. Math., 8(1), 181-217.

Wolfe, P. (1969) *Convergence Conditions for Ascent Methods,* SIAM Review, 11, 226-235.