A General Decomposition Theorem that Extends the Baum-Welch and Expectation-Maximization Paradigm to Rational Forms

Peter N. Yianilos
July 22, 2001
Netrics, Inc.
pny@netrics.com

Abstract

We consider the problem of maximizing certain positive rational functions of a form that includes statistical constructs such as conditional mixture densities and conditional hidden Markov models. The well-known Baum-Welch and expectation maximization (EM) algorithms do not apply to rational functions and are therefore limited to the simpler maximum-likelihood form of such models.

Our main result is a general decomposition theorem that like Baum-Welch/EM, breaks up each iteration of the maximization task into independent subproblems that are more easily solved – but applies to rational functions as well. It extends the central inequality of Baum-Welch/EM and associated high-level algorithms to the rational case, and reduces to the standard inequality and algorithms for simpler problems.

Keywords: Baum-Welch (forward backward algorithm), Expectation Maximization (EM), hidden Markov models (HMM), conditional mixture density estimation, discriminative training, Maximum Mutual Information (MMI) Criterion.

1 Introduction

Let (V, E) be a directed acyclic graph (DAG) with n vertices $v_1, \ldots, v_n$ and m edges $e_1, \ldots, e_m$, having a single source v_1 and sink v_n. A nonnegative parameterized weight function is associated with each edge. The *value* of a source-sink path is the product of the weights along it. The *value* of the graph is the sum over all source-sink paths, of each path value.

We refer to such a setting as a *simple product flow*, and the NP-complete problem of maximizing the graph's value over the weight function parameters [21] corresponds to maximum-likelihood parameter estimation for an HMM, mixture density, and other statistical models. We refer to the ratio of two simple product flows that share the same parameter space as a *rational product flow*, and the maximization problem corresponds to conditional (discriminative) parameter estimation. That is, a rational product flow consists of two distinct graphs. Its value is the quotient of their values regarding each as a simple product flow. Typically the numerator graph is a subgraph of the denominator induced by the labels provided in the supervised discriminative setting.

The edge weights are a function of a parameter space Ψ containing variable members as well as fixed ones. In statistical modeling the variables correspond to mixture element probabilities, symbol output probabilities, state transition probabilities, the parameters of a Gaussian density, or other such model components. The fixed parameters correspond to the data input to the estimation procedure. In these models an individual edge weight is determined by a single model component so that it depends on a subset of the total set of variable parameters – and distinct model components share no variables.

Our product flow setting is an abstraction of these models and an equivalence class on edges associates each edge e with a weight function denoted $w_{c(e)}$, where $c(e)$ gives the equivalence class index ranging from 1 to k for edge e. Each weight function w_i has corresponding parameters Ψ_i and in general will be associated with many edges. In the statistical models we're abstracting, the value of a model component also depends on the edge itself. So our weight function also accepts the index $n(e)$ of edge e as parameter to capture this dependency. The weight attached to edge e is then written $w_{c(e)}(\Psi_{c(e)}, n(e))$. For brevity we will write just $w_e(\Psi_e)$.

The value of a graph is computed in linear time by visiting vertices in topological order and accumulating partial results along the way. This corresponds to the well-known α computation of hidden Markov model evaluation and reestimation[19, 12]. The value α_i recorded at each intermediate vertex v_i corresponds to the sum of all path values from the source to that vertex. The sink v_n is visited last and its value α_n is that of the entire graph.

Considering the DAG with all edges reversed and performing the same algorithm corresponds to the β computation of HMM reestimation. Here each vertex records the value of all paths from it to the sink, and the source's value is that of the entire graph.

Let edge e connect vertex v_r with v_s and have weight w. Then γ_e is defined as $\alpha_r w \beta_s / \alpha_n$. This corresponds to the γ computation of HMM reestimation. If the edge weights are interpreted as probabilities of transiting an edge, then γ_e is the probability that a random source-sink path passes through edge e.

The Baum-Welch/EM optimization paradigm is iterative. The value of the graph is regarded as a function of parameter set Ψ', and the goal is to maximize it. The existing parameter set is denoted Ψ and is regarded as a fixed reference during optimization. Based on Ψ each iteration begins with the computation of γ values for each edge. Based on these γ values, k subproblems are then spawned; one for each weight function.

$$\left\{ \text{argmax}_{\Psi'_i} \sum_{e \in c^{-1}(i)} \gamma_e \log w_e(\Psi'_e) \right\}, i = 1, \ldots, k \tag{1}$$

The mathematics of Baum-Welch/EM tell us that any progress in one or more of these subproblems will strictly increase the value of the graph. The value of a graph is a complex interaction of edge values; each dependent on a weight function and on the edge itself. The fact that the task of optimizing a graph's value can be decomposed so neatly, is in our view the essence of the paradigm. All interactions are, in effect, sufficiently accounted for by the γ computation. Commonly used weight functions include discrete probability functions and Gaussian densities. In both cases simple and intuitive closed-form solutions exist for these subproblems – contributing to the popularity of the paradigm.

Our paper extends the paradigm to rational settings consisting of a numerator and denominator graph, where $C_N(\cdot)$ and $C_D(\cdot)$ denote the corresponding edge class functions. Theorem 2 is our main result. Its equation 3 extends equation 1 to the rational case by adding a second term. Our extension cleanly addresses the issue of decomposition, but we know of no simple closed-form solutions for the subproblems that compare with those of Baum-Welch/EM. General functional maximization techniques may be used and specialized techniques represent an interesting area for future work.

There is a long history of developments related to our work. The analysis of [3] marks the generally accepted beginning of the Baum-Welch/EM paradigm and hidden Markov modeling. Essentially the same mathematical idea is found in [9] which introduces EM in the specific context of maximum likelihood mixture density estimation. For completeness we remark that this idea can more generally be viewed as a projection operation with respect to Kullback-Leibler distance [16, 7] under Bregman's general framework for convex optimization [6]. This framework is closely related to the later work of I. Csiszár and G. Tusnády [8] and generalizes the much earlier work of [1, 18] and also [10, 5].

Interest in alternative estimation criteria has grown in recent years and [2] is a notable early milestone in this development. Viewing these more difficult problems in the context of polynomials [11] generalizes Baum-Welch/EM through an embedding that reduces the rational problem to a conventional one. Other work in the area includes the ECM algorithm [17], Hierarchical Mixtures of Experts [15], the CEM algorithm [13], and most recently [14]. Like [11] we consider the problem at an abstract level, but directly derive a different inequality and decomposition result for the rational case.

Jebara and Pentland in [13] (Equation 7) exploit the inequality $\log x \leq x - 1$ to obtain a decomposition result for simple mixture densities, where the rational form corresponds to the *a posteriori* likelihood of a labeled dataset. Note that throughout this paper natural logarithms are assumed. Our contribution is the application of this same inequality together with the inequality of lemma 3 to give an abstract and general decomposition result. Our result applies to all models we are aware of within the hidden Markov model class – to which mixture densities belong as a simple instance. For mixture densities, our decomposition reduces to theirs.

Our results are presented abstractly, rather than in the specific context of probability modeling. We suggest that the utility of this abstraction is i) to make clear that the result applies beyond conventional models to noncausal constructions or those with penalty functions, and ii) to provide a simple graph-oriented framework that makes it easier to reason about complex models. The origin of this outlook is [20, 21] where the development is limited to the simple (nonrational) case.

2 Rational Decomposition

Lemma 1 *Let $X = x_1, \ldots, x_r$ be positive values with sum S, and $X' = x'_1, \ldots, x'_r$ be another vector of positive values. Let $P_i = x_i/S$. Then:*

$$\log \frac{x'_1 + \ldots + x'_r}{x_1 + \ldots + x_r} = \underbrace{P_1 \left(\frac{x'_1}{x_1} - 1 \right) + \ldots + P_r \left(\frac{x'_r}{x_r} - 1 \right)}_{\overline{Q}(X,X')} + N(X, X') \qquad (2)$$

where $N(X, X') \leq 0$ and is equal to zero when $X = X'$, and $\overline{Q}(X, X')$ is defined as the bracketed terms above.

proof: From the elementary inequality $\log y \leq y - 1$ with equality at $y = 1$ we have that:

$$\log \frac{x'_1 + \ldots + x'_r}{x_1 + \ldots + x_r} = \frac{x'_1 + \ldots + x'_r}{x_1 + \ldots + x_r} - 1 + N(X, X')$$

and

$$\begin{aligned} \frac{x'_1 + \ldots + x'_r}{x_1 + \ldots + x_r} - 1 &= \frac{x'_1}{x_1 + \ldots + x_r} + \ldots + \frac{x'_r}{x_1 + \ldots + x_r} - 1 \\ &= P_1 \left(\frac{x'_1}{x_1} - 1 \right) + \ldots + P_r \left(\frac{x'_r}{x_r} - 1 \right) \end{aligned}$$

□

Lemma 2 *(Baum et al.) Given the setting of Lemma 1*

$$\log \frac{x'_1 + \ldots + x'_r}{x_1 + \ldots + x_r} = \underbrace{P_1 \log \frac{x'_1}{x_1} + \ldots + P_r \log \frac{x'_r}{x_r}}_{Q(X,X')} + P(X, X')$$

where $P(X, X') \geq 0$ and is equal to zero when $X = X'$, and $Q(X, X')$ is defined as the bracketed terms above.

proof: For completeness we recast here the well-known elegant proof originating with [4] into our framework. Let $S' = \sum_{i=1}^{r} x'$, and $P'_i = x'_i/S'$. We must show that $\log S'/S = Q(X, X') + P(X, X')$, satisfying the conditions of the lemma. Now:

$$\begin{aligned}
\log S'/S &= \log S' - \log S \\
&= \sum_{i=1}^{r} P_i \log S' - \sum_{i=1}^{r} P_i \log S \\
&= \sum_{i=1}^{r} P_i \log (x'_i/P'_i) - \sum_{i=1}^{r} P_i \log (x_i/P_i) \\
&= \sum_{i=1}^{r} P_i \log (x'_i/x_i) + \underbrace{\sum_{i=1}^{r} P_i \log (P_i/P'_i)}_{D(P\|P')}
\end{aligned}$$

where $D(P\|P')$ is the Kullback-Leibler distance [16, 7] between stochastic vectors P and P', which is always nonnegative and has value zero given equal arguments. □

The previous two lemmas give opposing bounds and are combined in the following theorem, which is the first step toward decomposition of our rational optimization problem.

Theorem 1 *Let $x_1(\Psi'), \ldots, x_r(\Psi')$ denote a set of positive scalar functions with domain consisting of values Ψ' from some parameter space. Let $y_1(\Psi'), \ldots, y_s(\Psi')$ be similarly defined. Next let Ψ denote any fixed reference set of parameter values. Finally define $P_{x_i} = x_i(\Psi)/\sum_{i=1}^{r} x_i(\Psi)$, and define P_{y_i} similarly. Then:*

$$\begin{aligned}
\log \frac{x_1(\Psi') + \ldots + x_r(\Psi')}{y_1(\Psi') + \ldots + y_s(\Psi')} &= \sum_{i=1}^{r} P_{x_i} \log \frac{x_i(\Psi')}{x_i(\Psi)} - \sum_{i=1}^{s} P_{y_i} \left(\frac{y_i(\Psi')}{y_i(\Psi)} - 1 \right) + C_{XY}(\Psi) + B_{XY}(\Psi', \Psi) \\
&= Q_X(\Psi, \Psi') - \overline{Q}_Y(\Psi, \Psi') + C_{XY}(\Psi) + B_{XY}(\Psi, \Psi')
\end{aligned}$$

where $B_{XY}(\Psi, \Psi') \geq 0$, and is zero when $\Psi' = \Psi$, and $C_{XY}(\Psi)$ is constant with respect to Ψ'.

proof:

$$\log \frac{x_1(\Psi') + \ldots + x_r(\Psi')}{y_1(\Psi') + \ldots + y_s(\Psi')} = \log \frac{x_1(\Psi') + \ldots + x_r(\Psi')}{x_1(\Psi) + \ldots + x_r(\Psi)} - \log \frac{y_1(\Psi') + \ldots + y_s(\Psi')}{y_1(\Psi) + \ldots + y_s(\Psi)} + C_{XY}(\Psi)$$

where $C_{XY}(\Psi) = \log [(x_1(\Psi) + \ldots + x_r(\Psi))/(y_1(\Psi) + \ldots + y_s(\Psi))]$, and lemmas 1 and 2 apply to the first two terms giving:

$$\log \frac{x_1(\Psi') + \ldots + x_r(\Psi')}{y_1(\Psi') + \ldots + y_s(\Psi')} = Q_X(\Psi, \Psi') - \overline{Q}_Y(\Psi, \Psi') + C_{XY}(\Psi) + \underbrace{P_X(\Psi, \Psi') - N_Y(\Psi, \Psi')}_{B_{XY}(\Psi, \Psi')}$$

□

Each term x_i and y_i of theorem 1 represents the value of a source-sink path within the DAG. This value is the product of edge contributions along it. To decompose the optimization problem our objective is to rewrite each such product as a sum of edge contributions. This happens naturally and exactly for the Q function since it is expressed in terms of logarithms. But the $\overline{Q}$ function is more difficult and we will give a sum decomposition that upper bounds the actual product value and contacts it when all terms are equal.

Lemma 3 *Given positive values* $v_1, \ldots, v_t$, *then* $\sum_{i=1}^{t} \frac{v_i^t}{t} \geq \prod_{i=1}^{t} v_i$

proof: Since the arithmetic mean is always greater than or equal to the geometric mean we have: $\sum_{i=1}^{t} \frac{v_i^t}{t} \geq \sqrt[t]{\prod_{i=1}^{t} v_i^t} = \prod_{i=1}^{t} v_i$. □

Theorem 2 (Main Result) *Let k denote the number of weight functions that occur in a given rational product-flow problem, and ℓ denote the length of the denominator graph's longest source-sink path. Then increasing the value of any of the k separate subproblems below, increases the value of main problem.*

$$\left\{ argmax_{\Psi_i'} \left[\sum_{e \in c_N^{-1}(i)} \gamma_e \log w_e(\Psi_e') - \sum_{e \in c_D^{-1}(i)} \frac{\gamma_e}{\ell} \left(\frac{w_e(\Psi_e')}{w_e(\Psi_e)} \right)^{\ell} \right] \right\}, i = 1, \ldots, k \quad (3)$$

proof: We assume without loss of generality that all source-sink paths in the denominator DAG are of exactly length ℓ. This is justified because an equivalent regularized DAG that has this property can easily be constructed by inserting new *dummy* vertices, and *dummy* edges with constant unity weight as needed to stretch any short paths.

Next consider a source-sink path i in the denominator graph and let E_i denote the set of edges along it. Let the value of this path correspond to a single term y_i of theorem 1. Then:

$$\begin{aligned} P_{y_i} \left(\frac{y_i(\Psi')}{y_i(\Psi)} - 1 \right) &= P_{y_i} \left(\prod_{e \in E_i} \frac{w_e(\Psi_e')}{w_e(\Psi_e)} - 1 \right) \\ &\leq \sum_{e \in E_i} \frac{P_{y_i}}{\ell} \left(\frac{w_e(\Psi_e')}{w_e(\Psi_e)} \right)^{\ell} - P_{y_i} \qquad \text{(by lemma 3)} \end{aligned}$$

Now each edge e in the denominator graph occurs in many source-sink paths. Summing the P_{y_i} over all paths passing through edge e yields γ_e. Then observing that the subtraction of P_{y_i} does not affect the optimization yields the additive term:

$$\frac{\gamma_e}{\ell} \left(\frac{w_e(\Psi_e')}{w_e(\Psi_e)} \right)^{\ell}$$

for each edge. Starting from lemma 2, which is applied to the numerator graph, and summing over all paths containing an edge e yields the additive term $\gamma_e \log w_e(\Psi_e')$. Subtracting the denominator term above from this, and grouping by weight function completes the proof. □

3 Discussion

Using Baum-Welch/EM, multiple observations are dealt with by adding their corresponding Q functions. The same is true in our more general rational setting by adding $\overline{Q}$ as well. The DAG depth ℓ is then related to the complexity of processing of a single observation. For simple mixture models $\ell = 2$ (see explanation below), and for HMMs $\ell = 2T$ where T is the number of time series elements comprising a single observation in the training set.

The DAG for a simple mixture density is easily described. Each of the edges outbound from the source select a mixture component, and each edge's weight is the probability of that component. Each edge then leads to an interior vertex with in-degree one. From that vertex a single outbound edge leads to the sink. Its weight is the observation's probability given the corresponding density function. Hence $\ell = 2$ for mixture densities since all source-sink paths are of this length.

So our result gives a new form of CEM [13] where both mixture coefficients and density models are optimized in a single step. If they are trained one-by-one, then $\ell = 1$ and our decomposition is equivalent to Equation 7 of [13].

Returning to more complex models such as HMMs, we remark that if a single observation is made up of a long series of elements, then ℓ can become prohibitively large and reduce the step size that is possible using our results. There are two general strategies that might be used to combat this effect: 1) break-up a single iteration into subiterations that optimize weight functions one-by-one ($\ell = 1$), or in groups; in both cases recomputing γ values between subiterations, 2) develop an adaptive algorithm that searches for a reduced ℓ value that nevertheless allows the optimization to make progress.

At our level of abstraction little can be said regarding convergence. We do know that strictly greater product-flows will result given strict progress in any of the subproblems, so that if the flow value is bounded (as it is for any probability model) then convergence of this value is ensured. But this does not imply convergence of model parameters. Saying more about this requires that something more be assumed about the weight functions themselves, and is beyond the scope of this paper.

It must be said that by extending to the rational case we leave behind a splendid property of conventional Baum-Welch/EM. Namely, that for common weight functions such as normal densities or discrete probability functions, the subproblems resulting from decomposition have globally optimal solutions that are trivially computed in closed form. Still, efficient approaches do sometimes exist. In [13] the authors consider simple mixtures of unnormalized Gaussians, and do not attempt to train all parameters simultaneously. The result is a practical prescription for this setting. General techniques for subproblem optimization is an interesting area for future work, but is beyond the scope of this paper.

Still, our work cleanly settles the issue of decomposition at a level of generality that includes all discrete-state models that the author is aware of within the hidden Markov family. In addition we suggest that our graph-based product-flow outlook makes it easier to reason about complex models and see, for example, that the paradigm's decomposition mathematics apply directly to variations including noncausal models, and penalized constructions that impose soft parsimony constraints such as MDL and MAP.

Acknowledgments

The author thanks Dan Gindikin and Sumeet Sobti for their comments on this manuscript, and Eric Sven Ristad with whom the original DAG-based outlook for simple product flows was developed.

References

[1] AGMON, S. The relaxation method for linear inequalities. *Canadian Journal of Mathematics 6*, 3 (1954), 382–392.

[2] BAHL, L. R., BROWN, P. F., DE SOUZA, P. V., AND MERCER, R. L. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *Proc. ICASSP-86* (1986), pp. 49–52.

[3] BAUM, L. E., AND EAGON, J. E. An inequality with application to statistical estimation for probabalistic functions of a Markov process and to models for ecology. *Bull. AMS 73* (1967), 360–363.

[4] BAUM, L. E., PETRIE, T., SOULES, G., AND WEISS, N. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math Stat. 41* (1970), 164–171.

[5] BREGMAN, L. M. Finding the common point of convex sets by the method of successive projections. *Dokl. Akad. Nauk SSSR 162* (1965), 487–490.

[6] BREGMAN, L. M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *Zh. vychisl. Mat. mat. Fiz. 7*, 3 (1967).

[7] COVER, T. M., AND THOMAS, J. A. *Elements of Information Theory*. Wiley, 1991.

[8] CSISZÁR, I., AND TUSNÁDY, G. Information geometry and alternating minimization procedures. *Statistics & Decisions Supplement Issue No. 1* (1984), 205–237. R. Oldenbourg Verlag, München 1984.

[9] DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Ser. B (methodological) 39* (1977), 1–38.

[10] EREMIN, I. I. A generalization of the motzkin - agmon relaxational method. *Usp. Mat. Nauk 20* (1965), 183–187.

[11] GOPALAKRISHNAN, P. S., KANEVSKY, D., NÁDAS, A., AND NAHAMOO, D. An inequality for rational functions with applications to some statistical estimation problems. *IEEE Trans. Information Theory 37* (1991), 107–113.

[12] HUANG, X. D., ARIKI, Y., AND JACK, M. A. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[13] JEBARA, T., AND PENTLAND, A. Maximum conditional likelihood via bound maximization and the cem algorithm. In *Advances in Neural Information Processing Systems* (1998), vol. 11, The MIT Press.

[14] JEBARA, T., AND PENTLAND, A. On reversing Jensen's inequality. In *Advances in Neural Information Processing Systems* (2000), vol. 13, The MIT Press.

[15] JORDAN, M., AND JACOBS, R. Hierachical mixtures of experts and the em algorithm. *Neural Computaion* (1994).

[16] KULLBACK, S., AND LEIBLER, R. A. On information and sufficiency. *Annals of Mathematical Statistics 22* (1951), 79–86.

[17] MENG, X., AND RUBIN, D. Maximum liklihood estimation via the ecm algorithm: A general framework. *Biometrika 80*, 2 (1993).

[18] MOTZKIN, T. S., AND SCHOENBERG, I. I. The relaxation method for linear inequalities. *Canadian Journal of Mathematics 6*, 3 (1954), 393–404.

[19] PORITZ, A. B. Hidden Markov models: a guided tour. In *Proc. ICASSP-88* (1988), pp. 7–13.

[20] RISTAD, E. S., AND YIANILOS, P. N. Finite growth models. Tech. Rep. 533-96, Princeton University, Department of Computer Science, 1996.

[21] YIANILOS, P. N. *Topics in Computational Hidden State Modeling*. PhD thesis, Princeton University, Department of Computer Science, June 1997.