NOTES ON THE EM ALGORITHM

1. The Dempster-Laird-Rubin Presentation

The general situation of the EM algorithm is that we observe data y which is *incomplete*, and assocated with *complete* data x. That is, there is a projection

$$\pi : \mathcal{X} \longrightarrow \mathcal{Y}$$

which is many-to-one. We observe $y \in \mathcal{Y}$ and the actual x which generated y is *hidden*. The *fiber* over y is the set

$$\mathcal{X}(y) = \pi^{-1}(y) = \{x \mid \pi(x) = y\}.$$

We follow (roughly) the notation of [3], and assume that x has a probability density $f_\phi(x)$ which depends on some vector of *parameters* $\phi \in \Omega$, where Ω is an open subset of $\mathbb{R}^N$. Typically, Ω represents various constraints on the parameters, as will become clear the following sections. The density of y is obtained by integrating over fibers:

$$g_\phi(y) = \int_{\mathcal{X}(y)} f_\phi(x)\, dx\,.$$

The basic idea of the EM algorithm is the following. According to the statistical principle of maximum likelihood, one would like to choose the parameters so as to maximize the log-likelihood $\log f_\phi(x)$. But this doesn't quite make sense, since x is unobserved (hidden). Instead, one can try to maximize the conditional expectation of $\log f_\phi(x)$ given the observation y and an estimate of the parameters ϕ.

Here's how this idea is borne out. Define

$$k_\phi(x \mid y) = \frac{f_\phi(x)}{g_\phi(y)}\,.$$

That is, k is the conditional density of x given y, assuming the parameter vector ϕ. Then

$$L(\phi) \equiv \log\, g_\phi(y) = \log\, f_\phi(x) - \log\, k_\phi(x \mid y)\,.$$

Let E_ϕ denote expectation with respect to the parameters ϕ. Since $L(\phi')$ is a function of y,

$$L(\phi') = E_\phi[\log\, f_{\phi'}(x) \mid y] - E_\phi[k_{\phi'}(x \mid y) \mid y]\,.$$

To verify this directly, note that

$$\begin{aligned} E_\phi[\log\, f_{\phi'}(x) \,|\, y] &= \frac{\int_{\mathcal{X}(y)} \log\, f_{\phi'}(x)\, f_\phi(x)\; dx}{\int_{\mathcal{X}(y)} f_\phi(x)\; dx} \\ &= \frac{1}{g_\phi(y)} \int_{\mathcal{X}(y)} \log\, f_{\phi'}(x)\, f_\phi(x) \end{aligned}$$

and

$$E_\phi[\log\, k_{\phi'}(x \,|\, y) \,|\, y] = \frac{1}{g_\phi(y)} \int_{\mathcal{X}(y)} \log\, \frac{f_{\phi'}(x)}{g_{\phi'}(y)}\, f_\phi(x)\; dx\,.$$

It follows that

$$\begin{aligned} &E_\phi[\log\, f_{\phi'}(x) \,|\, y] - E_\phi[k_{\phi'}(x \,|\, y) \,|\, y] = \\ &\qquad = \frac{1}{g_\phi(y)} \int_{\mathcal{X}(y)} \log\, f_{\phi'}(x)\, f_\phi(x) - \frac{1}{g_\phi(y)} \int_{\mathcal{X}(y)} \log\, \frac{f_{\phi'}(x)}{g_{\phi'}(y)}\, f_\phi(x)\; dx \\ &\qquad = \log\, g_{\phi'}(y) \\ &\qquad = L(\phi')\,. \end{aligned}$$

As a final piece of notation, we'll write $L(\phi') = Q(\phi' \,|\, \phi) - H(\phi' \,|\, \phi)$, where

$$Q(\phi' \,|\, \phi) = E_\phi[\log\, f_{\phi'}(x) \,|\, y]\,.$$

Lemma 1. *For any pair of parameters ϕ, $\phi' \in \Omega$, we have that*

$$H(\phi' \,|\, \phi) \le H(\phi \,|\, \phi)\,.$$

Proof. Applying the definitions,

$$\begin{aligned} &H(\phi' \,|\, \phi) - H(\phi \,|\, \phi) = \\ &\qquad = \frac{1}{g_\phi(y)} \left(\int_{\mathcal{X}(y)} \log\, \frac{f_{\phi'}(x)}{g_{\phi'}(y)}\, f_\phi(x)\; dx - \int_{\mathcal{X}(y)} \log\, \frac{f_\phi(x)}{g_\phi(y)}\, f_\phi(x)\; dx \right) \\ &\qquad = \frac{1}{g_\phi(y)} \left(\int_{\mathcal{X}(y)} \log\, \frac{f_{\phi'}(x)}{f_\phi(x)}\, f_\phi(x)\; dx + \int_{\mathcal{X}(y)} \log\, \frac{g_\phi(y)}{g_{\phi'}(y)}\, f_\phi(x)\; dx \right) \\ &\qquad \le \log \left(\frac{g_{\phi'}(y)}{g_\phi(y)} \right) + \log \left(\frac{g_\phi(y)}{g_{\phi'}(y)} \right) \\ &\qquad = 0 \end{aligned}$$

with the inequality coming from an application of Jensen's inequality for conditional expectations. $\square$

Since

$$L(\phi') - L(\phi) = [Q(\phi' \mid \phi) - Q(\phi \mid \phi)] + [H(\phi \mid \phi) - H(\phi' \mid \phi)]$$

it follows from the lemma that so long as $Q(\phi' \mid \phi) - Q(\phi \mid \phi) \geq 0$, the likelihood of the observed data increases. This motivates the basic EM algorithm:

$$\begin{array}{rll} \text{Initialize:} & \phi_0 \in \Omega & \\ \text{Iterate:} & \text{E-step:} & \text{Compute } Q(\phi \mid \phi_m) \\ & \text{M-step:} & \text{Set } \phi_{m+1} = \operatorname{argmax}_{\phi \in \Omega} Q(\phi \mid \phi_m) \end{array}$$

This is a greedy version of the algorithm, maximizing the gain in likelihood from Q at each step. But it may be computationally expensive to compute the maximum in the M-step. It thus makes sense to consider a modified EM-algorithm:

$$\begin{array}{rll} \text{Initialize:} & \phi_0 \in \Omega & \\ \text{Iterate:} & \text{E-step:} & \text{Compute } Q(\phi \mid \phi_m) \\ & \text{M-step:} & \text{Set } \phi_{m+1} = M(\phi_m), \text{ where } M : \Omega \to \Omega \text{ is any map} \\ & & \text{satisfying } Q(M(\phi) \mid \phi) \geq Q(\phi \mid \phi). \end{array}$$

We find it convenient to modify the choice of Q in [3] in the following way. Define $\widehat{Q}$ by

$$\widehat{Q}(\phi' \mid \phi) = E_\phi \left[\log \frac{f_{\phi'}(x)}{f_\phi(x)} \,\middle|\, y \right] .$$

Then another application of Jensen's inequality shows that

$$\begin{aligned} \widehat{Q}(\phi' \mid \phi) &= E_\phi \left[\log \frac{f_{\phi'}(x)}{f_\phi(x)} \,\middle|\, y \right] \\ &\leq \log E_\phi \left[\frac{f_{\phi'}(x)}{f_\phi(x)} \,\middle|\, y \right] \\ &= \log \frac{g_{\phi'}(y)}{g_\phi(y)} \\ &= L(\phi') - L(\phi) . \end{aligned}$$

Thus, since $\widehat{Q}(\phi \mid \phi) = 0$, the function $\widehat{Q}$ can just as well be used in the EM algorithm.

Theorem 1. *Let $\phi_0, \phi_1, \phi_2, \ldots$ be a sequence for an EM-algorithm such that*

1) $|L(\phi_n)| < C < \infty$ for all n.

2) $Q(\phi_{n+1} \mid \phi_n) - Q(\phi_n \mid \phi_n) \geq \lambda \|\phi_{n+1} - \phi_n\|^2$ for some $\lambda > 0$.

Then $\phi_n \longrightarrow \phi_*$ in L^2, for some $\phi_* \in \bar{\Omega}$.

Proof. The proof is a simple dominated convergence argument. Since $L(\phi_n)$ is bounded and increasing, it converges to some number $L_* < \infty$. The problem is to show that ϕ_n converges to a vector ϕ_*. It will follow that $L(\phi_*) = L_*$.

Since the sequence $L(\phi_n)$ is Cauchy, for any $\varepsilon > 0$ there is an N such that

$$L(\phi_{n+r}) - L(\phi_n) = \sum_{j=1}^{r} L(\phi_{n+j}) - L(\phi_{n+j-1}) < \varepsilon$$

for all $r = 1, 2, \ldots$ and $n \geq N$. This implies that

$$0 \leq \sum_{j=1}^{r} Q(\phi_{n+j} \,|\, \phi_{n+j-1}) - Q(\phi_{n+j-1} \,|\, \phi_{n+j-1}) < \varepsilon$$

and by assumption 2), that

$$0 \leq \lambda \left(\sum_{j=1}^{r} \|\phi_{n+j} - \phi_{n+j-1}\|^2 \right) < \varepsilon\,.$$

Thus, ϕ_n converges. $\square$

Condition 2) in the theorem is a rather stringent assumption. It is not satisfied for many practical M's and Q's. It is thus possible for the sequence ϕ_n to wander around Ω without ever converging, while the likelihood steadily increases, and eventually converges.

2. The EM Algorithm and Language Modeling

The typical situation for the EM algorithm in language modeling is that we have a set of *histories* h and *futures* f and want to maximize the likelihood of predicting f from h, given a collection of *training events* $\mathcal{E} = \{(h, f)\}$:

$$L(\phi) = \sum_{\mathcal{E}} c(h, f) \log P_\phi(f \,|\, h)$$

where $c(h, f)$ is the multiplicity ("count") of the event (h, f) in $\mathcal{E}$. For example, f might be a word predicted from a history of the previous two words, or it might be an English sentence to be predicted from a French sentence. We add "structure" or "linguistics" to this setup by modeling some hidden quantity that we think is going to help predict f better.

2.1. Marginal Models. The incomplete data for most language models takes the form of a Cartesian product, and the fibers are the associated level sets. Thus, the models typically take a form where the "future" f and the hidden data x are specified by a joint distribution, and the observed data is given by the associated marginal distribution:

$$L(\phi) = \sum_{\mathcal{E}} c(h,f) \, \log \sum_{x} P_\phi(f, x \mid h) \, .$$

For example, the hidden data x could be a sequence of parts-of-speech, an alignment between French and English sentences, or a parse tree for a sentence. Typically the sum $\sum_x$ is exponential in the size of h and f. Part of the algorithmic art of language modeling is to make this sum manageable.

The difference in the conditional expectations of the complete data log-likelihoods given the observed data is written here as

$$\widehat{Q}(\phi' \mid \phi) = \sum_{\mathcal{E}} c(h,f) \sum_{x} P_\phi(x \mid h, f) \log \frac{P_{\phi'}(f, x \mid h)}{P_\phi(f, x \mid h)} \, .$$

For many language models, including probabilistic context-free grammar, this function is convex in ϕ', and the maximum can be calculated in closed form. This class of models is described next.

2.2. Algebraic Models. Most of the models that arise in language modeling are associated with *algebraic* expressions; that is, the probabilities are expressed as (typically homogeneous) polynomials in the parameters. Suppose, for example, that

$$P_\phi(f, x \mid h) = \prod_{\omega \in \Omega} \phi(\omega)^{c(\omega; x, f, h)} \, .$$

The $\phi(\omega)$'s are the *parameters*, and they are subject to certain linear constraints, such as

$$\phi(\omega) \geq 0 \quad \text{and} \quad \sum_{\omega \in \Omega} \phi(\omega) = 1 \, .$$

For such models, the M-step in the EM-algorithm can be carried out exactly, and the parameter updates take on a particularly simple form, which we now derive.

The EM algorithm tells us to compute the function $\widehat{Q}(\phi \mid \phi_n)$ and to solve the equation

$$\frac{\partial \widehat{Q}(\phi \mid \phi_n)}{\partial \phi} - \lambda = 0$$

where λ is a Lagrange multiplier, corresponding to the constraint $\sum_\omega \phi(\omega) = 1$. Using the algebraic form of the model we can calculate

$$\begin{aligned}\widehat{Q}(\phi' \,|\, \phi) &= \sum_{\mathcal{E}} c(h,f) \sum_x P_\phi(x \,|\, h,f) \log \frac{P_{\phi'}(f,x \,|\, h)}{P_\phi(f,x \,|\, h)} \\ &= \sum_{\mathcal{E}} c(h,f) \sum_x P_\phi(x \,|\, h,f) \sum_\omega c(\omega; x,h,f) \log \phi'(\omega) \\ &\qquad - \sum_{\mathcal{E}} c(h,f) \sum_x P_\phi(x \,|\, h,f) \log P_\phi(f,x \,|\, h)\,.\end{aligned}$$

In particular, $\widehat{Q}(\phi' \,|\, \phi)$ is a concave function of the parameters $\phi'(\omega)$. Note that

$$c(w; x,f,h) = \frac{\phi(\omega)}{P_\phi(x,f \,|\, h)} \frac{\partial P_\phi(x,f \,|\, h)}{\partial \phi(\omega)}\,.$$

Taking partial derivatives of $\widehat{Q}$ and including the Lagrange multiplier, we are led to the condition that must be obtained at the unique maximum in the M-step:

$$\sum_{\mathcal{E}} c(h,f) \sum_x P_\phi(x \,|\, h,f) \frac{c(\omega; x,h,f)}{\phi'(\omega)} = \lambda\,.$$

This, in turn, leads to the EM update formula

$$\phi_{n+1}(\omega) = \lambda^{-1} \sum_{\mathcal{E}} c(h,f) \sum_x P_{\phi_n}(x \,|\, h,f) c(\omega; x,h,f)\,.$$

Thus, the reestimated parameters are *normalized expected counts*. The expectation

$$E_{\phi_n}[c(\omega; f,h)] \equiv \sum_x P_{\phi_n}(x \,|\, f,h) c(\omega; x,f,h)$$

is interpreted as the expected number of times, under the model ϕ_n, that ω is used in generating f from h. We note that a similar analysis would hold for any model which is a *rational* function of its parameters.

Computing the expected counts usually involves some kind of "forward-backward" calculation or approximation to make the sum $\sum_x$ manageable. For general finite-state machines this calculation can be neatly characterized [2]. The calculation was demonstrated in [1] for the case where the hidden data is a parse tree derived from a context-free grammar. In the next section we will derive this calculation within the framework that we have set up.

3. The Inside-Outside Algorithm

Let G be a context-free grammar consisting of a collection of rules $\{A \to \alpha\}$, where each α is a string of terminals and nonterminals. For each string $w \in \mathcal{L}(G)$, the language of G, there is a corresponding set of parse trees t, each of which has $w = w_1\, w_2 \cdots w_N$ as leaves. If we observe only w, then for an ambiguous grammar, the actual tree used to derive w is hidden.

Suppose we have a joint distribution $P_\phi(w,t)$, giving the probability of deriving w using the tree t. Then the marginal distribution

$$P_\phi(w) = \sum_t P_\phi(w,t)$$

gives a language model. In the notation of Section 2, $P_\phi(w,t)$ is the *complete data density* $f_\phi(x)$ and $P_\phi(w)$ is the *incomplete data density* $g_\phi(y)$. The fiber $\mathcal{X}(w)$ over the sentence w is a finite collection of parse trees. The joint distribution takes the form

$$\begin{aligned} P_\phi(w,t) &= \prod_\omega \phi(\omega)^{c(\omega;\,t,w)} \\ &= \prod_{A\to\alpha} \phi(A\to\alpha)^{c(A\to\alpha;\,t,w)} \end{aligned}$$

where $c(A\to\alpha;\, t,w)$ is the number of times that the rule $A\to\alpha$ appears in the parse tree t for the sentence w. The parameters $\phi(A\to\alpha)$ are normalized so that

$$\sum_\alpha \phi(A\to\alpha) = 1\,.$$

Thus, there will be a Lagrange multiplier for each nonterminal A.

Such a model may be *deficient*, and not assign probability one to finite strings. A sufficient condition that this does not happen can be expressed by indexing the nonterminals as $A_1,\ldots,A_N$, and letting M be the $N\times N$ matrix given by

$$M_{ij} = \sum_\alpha \phi(A_i\to\alpha)\, n_j(\alpha)$$

where $n_j(\alpha)$ is the number of nonterminal symbols A_j appearing in α. If M has largest eigenvalue $\rho < 1$, then the language model $P_\phi(w)$ assigns probability one to finite sentences in the language of the grammar.

The model is parameterized by making the *Markov assumption* that the probability with which a nonterminal is rewritten as a string α depends only on the nonterminal, and not on any surrounding context. This assumption leads to an efficient training algorithm.

There are two distinct problems associated with this setup. The first, called the *language modeling problem*, is to find the set of parameters which maximize the probability $\prod_{w\in\mathcal{C}} P_\phi(w)$ of some training corpus $\mathcal{C}$. The second, called the *parsing problem*, is to maximize the "correctness" of the most probable parse

$$\hat{t}(w) = \mathrm{argmax}_t\; P_\phi(t\,|\,w)\,.$$

The EM algorithm is directly involved with only the language modeling problem. Experience has shown it to be difficult to couple the two problems.

To apply the EM algorithm, we consider the auxiliary function

$$\widehat{Q}(\phi' \mid \phi) = \sum_w c(w) \sum_t P_\phi(t \mid w) \log \frac{P_{\phi'}(t,w)}{P_\phi(t,w)}\,.$$

Taking the derivative $\partial/\partial\phi'(A \to \alpha)$ gives

$$\frac{\partial \widehat{Q}(\phi' \mid \phi)}{\partial \phi'(A \to \alpha)} = \sum_w c(w) \sum_t \frac{P_\phi(t \mid w) c(A \to \alpha;\, t, w)}{\phi'(A \to \alpha)}\,.$$

We thus need to compute the expected counts

$$\sum_t P_\phi(t \mid w) c(A \to \alpha;\, t, w)\,.$$

The sum $\sum_t$ is potentially exponential. But this is the same as evaluating

$$\frac{\phi(A \to \alpha)}{P_\phi(w)} \frac{\partial P_\phi(w)}{\partial \phi(A \to \alpha)} = \sum_t P_\phi(t \mid w) c(A \to \alpha;\, t, w)\,,$$

and it turns out that there is an efficient way of computing the partial derivative on the lefthand side.

We'll now assume, but only for convenience, that the grammar is in Chomsky normal form. Thus, each rule is either of the form $A \to BC$ or $A \to w$. The *position* of a rule $A \to BC$ within a tree t can be specified by a triple (i,j,k), $i \le j \le k$.

The partial derivative of the probability $P_\phi(S \Rightarrow w) = P_\phi(w)$ with respect to the parameter $\phi(A \to BC)$ only involves those parse trees which use the rule $A \to BC$. Consider the event "$S \Rightarrow w$ using $A \to BC$ in position (i,j,k)". Because of the Markov property, the probability of this event can be written as a product of four terms as follows:

$$\begin{aligned} &P_\phi(S \Rightarrow w;\ \text{using } A \to BC \text{ in position } (i,j,k)) = \\ &\qquad\qquad P_\phi(S \Rightarrow w_1 \cdots w_{i-1}\, A\, w_{k+1} \cdots w_N) \times \\ &\qquad\qquad\quad \times \phi(A \to BC)\, P(B \Rightarrow w_i \cdots w_j) P(c \Rightarrow w_{j+1} \cdots w_k)\,. \end{aligned}$$

From this it is not difficult to see that

$$\begin{aligned} &\frac{\partial P_\phi(S \Rightarrow w)}{\partial \phi(A \to BC)} = \\ &\sum_{i,j,k} P_\phi(S \Rightarrow w_1 \cdots w_{i-1}\, A\, w_{k+1} \cdots w_N)\, P(B \Rightarrow w_i \cdots w_j)\, P(C \Rightarrow w_{j+1} \cdots w_k)\,. \end{aligned}$$

Thus, the expected number of times that the rule $A \to BC$ is used in generating the sentence w using the model ϕ is given by

$$\begin{aligned} E_\phi[c(A \to BC; w)] &= \sum_t P_\phi(t \mid w) c(A \to BC; t, w) \\ &= \frac{\phi(A \to BC)}{P_\phi(w)} \sum_{i,j,k} \beta_{ik}(A) \alpha_{ij}(B) \alpha_{j+1k}(C) \end{aligned}$$

where

$$\alpha_{ij}(A) = P_\phi(A \Rightarrow w_i \cdots w_j)$$

and

$$\beta_{ij}(A) = P_\phi(S \Rightarrow w_1 \cdots w_{i-1}\, A\, w_{j+1} \cdots w_N)\,.$$

Similarly,

$$E_\phi[A \to a; w] = \frac{\phi(A \to a)}{P_\phi(w)} \sum_i \delta_a(w_i)\, \beta_{ii}(A)\,.$$

There is an efficient method for computing the α's and β's using the CKY chart-parsing algorithm. The method for doing this is implicit in the following formulas:

$$\begin{aligned} \alpha_{ij}(A) &= \sum_{B,C} \sum_{i \le k \le j} \phi(A \to BC) \alpha_{ik}(B)\, \alpha_{k+1j}(C) \\ \alpha_{ii}(A) &= \phi(A \to w_i) \\ \beta_{ij}(A) &= \sum_{B,C} \sum_{k<i} \phi(B \to CA) \alpha_{ki-1}(C)\, \beta_{kj}(B) + \\ &\quad + \sum_{B,C} \sum_{k>j} \phi(B \to AC) \alpha_{j+1k}(C)\, \beta_{ik}(B) \\ \beta_{1N}(A) &= \delta_S(A)\,. \end{aligned}$$

The reestimated parameters are then the normalized counts:

$$\phi'(A \to \alpha) = \lambda_A^{-1} \mathrm{count}(A \to \alpha)$$

where

$$\mathrm{count}(A \to \alpha) = \sum_{w \in \mathcal{E}} E_\phi[c(A \to \alpha); w]\,.$$

The sum is over all sentences in the training data $\mathcal{E}$. The Lagrange multiplier λ_A is given by

$$\lambda_A = \sum_\alpha \mathrm{count}(A \to \alpha)\,.$$

The results of Sections 2 and 3 prove that the resulting model does not assign a smaller likelihood to $\mathcal{E}$.

References

1. J.K. Baker, *Trainable grammars for speech recognition*, Proceedings of the Spring Conference of the Acoustical Society of America (Boston, MA), pp. 547–550, 1979.
2. L.E. Baum, *An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process*, Inequalities **3** (1972), 1–8.
3. A.P. Dempster, N.M. Laird, and D.B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society **39** (1977) B, 1–38.